MITSUBISHI

FATEC

Q172CPU(N)
Q173CPU(N)

MOTION CONTROLLER
SCHOOL TEXTBOOK

Microsoft® Windows® Personal Computer Operation Version
SW6RN-GSV22P

• SAFETY INSTRUCTIONS •

(Always read before starting practice)

When designing the system, always read the related manuals, and pay special attention to safety.
When practicing, pay attention to the following points and make sure to correctly handle the system.

[Precautions for Practice]

DANGER

- Do not touch the terminals while the power is ON. There is a risk of electric shock accidents.
- Always turn the power OFF or sufficiently confirm the surrounding safety before opening the safety covers.

CAUTION

- Always follow the instructor's instructions when practicing.
- Do not remove the practice unit or change the wiring without instructions from the instructor.
  Failure to observe this could lead to faults, incorrect operations, injuries or fires.
- Always turn the power OFF before installing or removing a unit.
  There is a risk of unit faults or electric shocks if this is carried out while the power is ON.
- If any abnormal odor or noise is sensed during practice, always press the "Power switch" or "Emergency Stop switch", and stop the machine.
- If any error occurs, notify your instructor immediately.

Revision History

* The textbook No. is indicated on the lower left of the back cover.

Date of print	*Textbook No.	Revision details
January 2001	SH-030010-A	First print
January 2001	SH-030010-B	Partial revision Table of contents, section 2.2, section 3.2, section 9.3, section 9.5, section 9.6, section 9.9, section 9.10, section 10.4, section 10.10
April 2001	SH-030010-C	Partial revision Table of contents, section 9.2, section 9.9, section 10.1, section 10.8
March 2002	SH-030010-D	Additions Section 3.2.5, section 2.3.6, section 3.2.7, section 3.2.8, Chapter 10, Appendix 5 Revision Changed teaching operation to monitor operation, section 9.10.2 Partial revision Table of contents, section 1.1, section 1.3, section 2.1, section 2.1.1, section 2.1.2, section 2.2, section 2.2.1, section 2.2.2, section 2.3, Chapter 3, section 3.1, section 3.1.2, section 3.2, section 4.1, section 4.2.1, section 4.2.4, section 4.3.1, section 4.3.2, section 4.3.3, section 4.3.4, section 4.3.5, section 4.4.1, section 4.4.2, section 7.2, section 7.4, section 8.1, section 8.2, section 8.3.1, section 8.4, section 9.1, section 9.2, section 9.3, section 9.5, section 9.6.1, section 9.6.2, section 9.6.3, section 9.6.4, section 9.6.5, section 9.7, section 9.9.1, section 9.9.2, section 9.9.3, section 9.9.4, section 9.9.5, section 9.10.1, section 9.10.3, section 11.3, section 11.4, section 11.4.1, section 11.8, Appendix 2, Appendix 3.2, Appendix 4

© 2002 MITSUBISHI ELECTRIC CORPORATION

CONTENTS

Chapter 1 Outline 1-1 to 1-6

1.1 Features of the motion controller 1-1
1.2 Outline of control 1-4
1.2.1 Real mode control for SV13 transfer assembly and SV22 automatic machine 1-4
1.2.2 Virtual mode control for SV22 automatic machine 1-5
1.3 Items required to start up system 1-6

Chapter 2 Explanation of Functions 2-1 to 2-8

2.1 List of specifications 2-1
2.1.1 List of motion controller specifications 2-1
2.1.2 List of SFC performance specifications 2-2
2.2 System configuration drawing 2-3
2.2.1 Q172CPU(N) system 2-3
2.2.2 Q173CPU(N) system 2-4
2.3 Names of each part 2-5

Chapter 3 Q-PLC Multi-CPU 3-1 to 3-14

3.1 Multi-CPU system 3-2
3.1.1 Setting the multi-CPU system 3-2
3.1.2 Mounting position of Q-PLC CPU/Q motion CPU 3-3
3.1.3 Input/output numbers. 3-4
3.1.4 Automatic refresh for shared memory 3-5
3.2 Multi-CPU motion dedicated commands 3-7
3.2.1 SFCS motion SFC program start command 3-7
3.2.2 GINT interrupt command to other machine’s CPU 3-9
3.2.3 Read from DDRD Q motion CPU device command 3-11
3.2.4 Write to DDWR Q motion CPU device command 3-13

Chapter 4 Q Motion CPU 4-1 to 4-30

4.1 System settings 4-1
4.2 Servo data 4-2
4.2.1 Basic system setting 4-3
4.2.2 Multi-CPU setting 4-4
4.2.3 Fixed parameters 4-5
4.2.4 Servo parameters 4-6
4.2.5 Zero point return data 4-10
4.2.6 JOG operation data 4-11
4.2.7 Parameter block 4-12
4.2.8 Limit switch output function 4-13
4.3 Positioning control device 4-14
4.3.1 Status/command signals M2400 to M5471 (For Q172) 4-15
4.3.2 Internal relays M2000 to M2319 (For Q172) 4-18

4.3.3 Data registers D0 to D1315 (For Q172) 4-20
4.3.4 Special relays M9073 to M9079, M9104, M9105 4-24
4.3.5 Special registers D752 to D799, D9017, D9019, D9104, D9180 to D9199 (For Q172) 4-24
4.4 Motion SFC dedicated devices 4-26
4.4.1 Motion registers (#0 to #8191) 4-26
4.4.2 Coast timer (FT) 4-29

Chapter 5 SFC Program 5-1 to 5-20

5.1 SFC program configuration 5-1
5.2 List of SFC symbols 5-2
5.3 List of branch/connection diagrams 5-4
5.4 SFC program name 5-5
5.5 Steps 5-6
5.5.1 Motion control step 5-6
5.5.2 Operation control step 5-7
5.5.3 Subroutine call/start step 5-8
5.5.4 Clear step 5-9
5.6 Transition 5-10
5.7 Jump/pointer 5-11
5.8 END 5-11
5.9 Branch/connection 5-12
5.9.1 Parallel shifting 5-12
5.9.2 Selective branch/connection 5-13
5.9.3 Parallel branch/parallel connection 5-14
5.10 Y/N transition 5-15
5.11 Task operation 5-17
5.12 SFC parameters 5-19
5.12.1 Task parameters 5-19
5.12.2 Program parameters 5-19
5.13 SFC program start method 5-20
5.14 SFC program end method 5-20

Chapter 6 SV22 Servo Programs 6-1 to 6-24

6.1 Servo program 6-1
6.1.1 Servo program configuration 6-1
6.1.2 List of servo commands 6-2
6.1.3 Linear control 6-8
6.1.4 Circular interpolation control using auxiliary point designation 6-9
6.1.5 Circular interpolation control using radius designation 6-10
6.1.6 Circular interpolation control using center point designation 6-11
6.1.7 Fixed-dimension feed control 6-12
6.1.8 Speed control 6-12
6.1.9 Speed/position changeover control 6-13
6.1.10 Speed changeover control 6-14
6.1.11 Constant-speed control 6-15
6.1.12 Repeated control (for speed changeover control and uniform speed control) 6-16
6.1.13 Simultaneous start 6-17

6.1.14 Zero point return 6-18
6.1.15 Position follow-up control 6-19
6.1.16 High-speed oscillation control 6-20
6.1.17 Helical interpolation control with auxiliary point designated 6-21
6.1.18 Helical interpolation control with radius designated 6-22
6.1.19 Helical interpolation control with center point designated 6-23
6.1.20 Current value change 6-24

Chapter 7 Operation Control Program 7-1 to 7-10

7.1 Order of operator and function priority 7-1
7.2 List of operation control and transition commands 7-2
7.3 Dedicated motion function (CHGV, CHGT) 7-4
7.4 Other commands 7-6

Chapter 8 Windows Personal Computer Operations 8-1 to 8-16

8.1 Flow of creating data for operating motion controller 8-1
8.2 Registering the main unit OS 8-2
8.3 Setting the Q-PLC CPU 8-7
8.3.1 Reading the sequence program 8-7
8.3.2 Setting the multi-CPU 8-8
8.3.3 Writing the sequence program 8-11
8.4 Starting up SW6RN-GSV22P 8-13

Chapter 9 Basic Practice Using the SV22 Real Mode 9-1 to 9-68

9.1 Details of practice 9-1
9.2 Q172CPU practice machine system configuration 9-2
9.3 System setting 9-5
9.4 Setting the servo data 9-12
9.5 Practice SFC programs 9-15
9.6 Creating SFC programs 9-19
9.6.1 Creating a new SFC program 9-19
9.6.2 Creating the SFC diagram 9-21
9.6.3 Inputting the transition and operation control step 9-27
9.6.4 Inputting the motion control step 9-31
9.6.5 SFC program parameter setting and batch conversion 9-36
9.7 Writing to the motion CPU 9-38
9.8 Test operation 9-40
9.8.1 JOG operation 9-40
9.8.2 Running the servo program 9-42
9.9 Program for operation 9-44
9.9.1 Initialization 9-44
9.9.2 JOG operation 9-46
9.9.3 Main routine SFC program (real mode operation) 9-49
9.9.4 Zero point return 9-50
9.9.5 Continuous positioning 9-51
9.10 Operating the practice machine 9-53
9.10.1 Operation 9-53

9.10.2 Monitor operation 9-56
9.10.3 Monitor trace graph 9-59
9.11 Ending the operations 9-68
9.11.1 Ending the SW6RN-GSV22P operations 9-68
9.11.2 GX Developer END operation 9-68

Chapter 10 Applied Practice with SV22 Real Mode 10-1 to 10-46

10.1 Details of practice 10-1
10.2 Q172CPU practice machine system configuration 10-2
10.3 Practice SFC programs 10-5
10.4 Writing to the motion CPU 10-11
10.5 Program for operation 10-16
10.5.1 JOG operation 10-16
10.5.2 Main routine SFC program (real mode operation) 10-19
10.5.3 Execution of servo program (motion control step) 10-20
10.5.4 Stopping 10-24
10.5.5 Error reset 10-25
10.5.6 Current value change 10-27
10.5.7 Speed change (CHGV) 10-28
10.5.8 Reading actual current value 10-30
10.5.9 Continuous positioning 10-32
10.5.10 M code function 10-34
10.5.11 Indirect setting of servo program address 10-36
10.6 Operating the practice machine 10-38
10.6.1 Operation 10-38

Chapter 11 Practicing with the SV22 Virtual Mode 11-1 to 11-44

11.1 Mechanism program 11-1
11.1.1 Mechanism module connection diagram 11-1
11.1.2 List of mechanism modules 11-2
11.1.3 Virtual servomotor 11-3
11.1.4 Synchronous encoder 11-3
11.1.5 Virtual axis 11-3
11.1.6 Gears 11-3
11.1.7 Clutch 11-4
11.1.8 Transmission 11-5
11.1.9 Differential gears 11-5
11.1.10 Rollers 11-5
11.1.11 Ball screw 11-6
11.1.12 Rotary table 11-6
11.1.13 Cam 11-7
11.2 Details of practice 11-9
11.3 Starting up SW3RN-CAMP and creating the cam 11-11
11.4 SFC program for virtual mode 11-16
11.4.1 New creation of SFC program for virtual mode 11-18
11.4.2 Inputting the motion control steps for the virtual mode 11-22
11.5 Editing the mechanism 11-26

11.6 Writing to the motion CPU 11-30
11.7 Reading of sequence program from Q-PLC CPU 11-31
11.8 SFC program for practice 11-33
11.9 Practice machine operations 11-39
11.10 Exercise (Roller setting) 11-44

Appendix A-1 to A-40

Appendix 1 Examples of programs for SV22 virtual mode A-1
Appendix 2 Sample motion SFC A-10
Appendix 3 Operating the Windows personal computer A-21
Appendix 3.1 Backing up an FD A-21
Appendix 3.2 Installing SW6RN-GSV22P A-23
Appendix 4 Comparison between A173UHCPU/A172SHCPUN A-28
Appendix 5 Sequence command dedicated to motion A-30
Appendix 5.1 SVST servo program start request command A-30
Appendix 5.2 CHGA current value change command A-32
Appendix 5.3 CHGV speed change command A-34
Appendix 5.4 CHGT torque limit value change request command A-36
Appendix 6 Explanation of terms A-38

Introduction

This is the school textbook prepared to provide an understanding of the motion controller to enable easy control of the multi-axis positioning operations.
In this textbook, the outline of the Q motion controller is explained, and the methods of setting the data to carry out positioning using a DOS/V personal computer and the SW6RN-GSV22P automatic machine software package are explained. In addition, the methods of creating servo programs, mechanical support language and sequence programs are explained.

(The software package and function specifications will differ according to the machine model.)

Chapter 1 Outline

1.1 Features of the motion controller

The motion controller has the following features.

(1) Q-PLC CPU and multi-CPU system

A flexible system configuration, which allows the processing load to be spread out, is realized by carrying out complicated servo control with the Q motion CPU unit, and other machine control and information with the Q-PLC CPU unit.

(2) Product line-up to match applications

The following motion controller models are available to match the system scale required for multi-axis positioning.

- Q172CPU(N) (1 to 8 axes multi-axis positioning function)
- Q173CPU(N) (1 to 32 axes multi-axis positioning function)

(3) Control using MR-H-B(N)/MR-J2(S)-B type servo amplifier possible

A 10W to 55kW servomotor can be controlled by connecting the MR-H-B(N)/MR-J2(S)-B/MR-J2M-B type servo amplifier externally to the motion network SSCNET.

(The Q172 can control up to eight servomotors and the Q173 up to 32 servomotors.)

(4) High-speed serial communication with servo amplifier possible

Using the motion network SSCNET high-speed serial communication, the servo data can be collected, the servo parameter can be changed, the servo can be tested and monitored, and the mechanism program can be monitored. The speed command can be output at up to 10Mpps, enabling high-speed high-accuracy positioning.

(5) Absolute position system possible

An absolute position system can be structured by using a servomotor with absolute position detector. (Zero point return is not required even if a power failure occurs.)

(6) Windows personal computer can be used as positioning programming tool

By using a Windows personal computer with the dedicated software package, the motion SFC can be programmed, the servo control can be programmed, monitored and tested.

Windows personal computer SW6RNC-GSVPRO

(7) Operating system (OS) can be changed

Software packages to match applications are available, and by directly writing the optical OS (refer to comparison table in section 2.1) into the CPU's built-in flash memory, a motion controller matching each machine can be created.
This system is also compatible with software package function upgrades.

1) SV13 for transfer assembly
   Using the dedicated servo commands, 1 to 4-axis linear interpolation, 2-axis circular interpolation, 3-axis helical interpolation, CP control (uniform speed control), speed control and position follow-up control can be carried out, making this system suitable for applications such as transfer machines and assembly machines.
   Order control is enabled with SFC.
2) SV22 for automatic machine
   Multiple servomotors can be simultaneously controlled with the mechanical support language, and cam control can be carried out with the software. This is suitable for applications such as automatic machines.

(8) Software cam ... Valid only with SV22

When the cam mechanism, often used in machine mechanisms is replaced with a virtual mode cam for servomotor control, the following features can be realized.

1) Cam curve data can be created easily with the cam curve creation software package, thereby eliminating the need to manufacture cam parts.
2) The cam can be replaced easily by changing the cam No. in the motion SFC program.
3) There is no need to consider wear and life unique to the cam.

(9) Mechanism support language (Mechanism program) ... Valid only with SV22

Conventionally, synchronous operation and coordinated operation were required for industrial machines and automatic machines, and as a means to achieve this, each operation was mechanically connected.
With this method, the output mechanisms such as the rotation operation, linear operation, reciprocating operation and feed operation, are operated from the main shaft, which is the drive source, using a conveyance mechanism such as gears, a clutch or crank. Although accurate synchronization operation and coordinate operations are possible, this method lacks flexibility.

The mechanical support language frees the system from the conventional mechanical connection, and allows the positioning control functions and performance to be improved because the servomotor is controlled by processing the machine mechanism movements with software. At the same time, this is an electrical method, so there are few limitations to the mechanism and rational designs can be made.

The system from the main shaft to the conveyance mechanism such as the gears, clutch, reduction gears or differential gears, to the output mechanism such as the roller output, ball screw output, rotary table output or cam output are described with figures on the peripheral device screen. Just by setting each module parameter, the synchronized operation and coordinate operation can be realized and a flexible control system can be easily structured.
Thus, machine parts such as the main shaft, gears, clutch, crank, reduction gears, differential gears and cam can be greatly reduced or omitted, allowing costs to be reduced and wear to be eliminated.

(10) Teaching function

A servo program to match the actual part can be created with the current value teaching function.

(11) Limit switch function

The ON/OFF signal corresponding to the watch data range, set for each output device (X, Y, M, L, B) is output.

Output devices for up to 32 points can be set.

1.2 Outline of control

1.2.1 Real mode control for SV13 transfer assembly and SV22 automatic machine

(a) A system containing a servomotor is directly controlled with the servo program.

(b) The positioning parameters must be set, and the servo program and positioning sequence program must be created.

(c) The procedures for positioning control are indicated below.

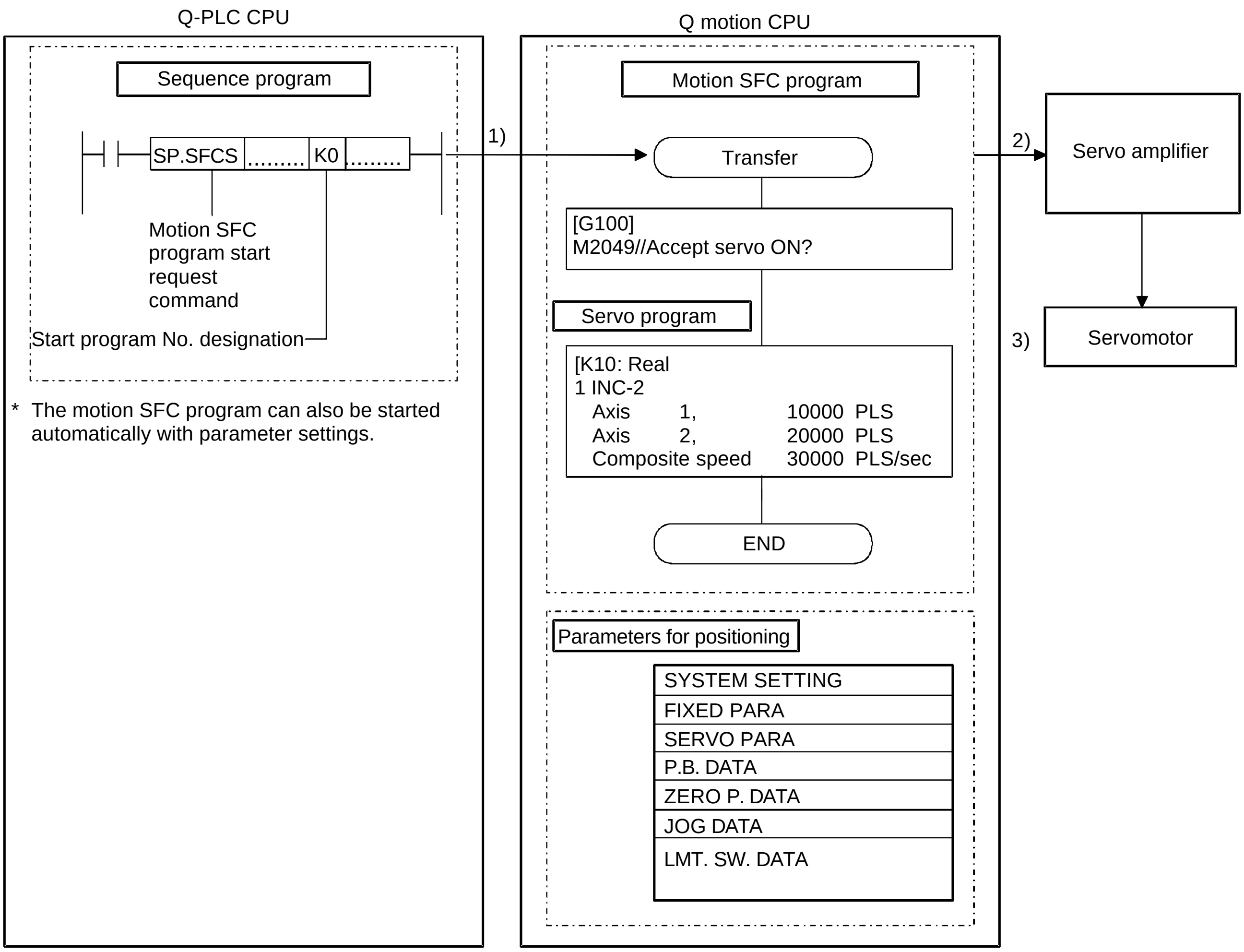
1) Start up of the motion SFC program is requested with the sequence program's SFCS command

↓

2) Positioning control is carried out with the designated motion SFC program

↓

3) The servomotor is controlled



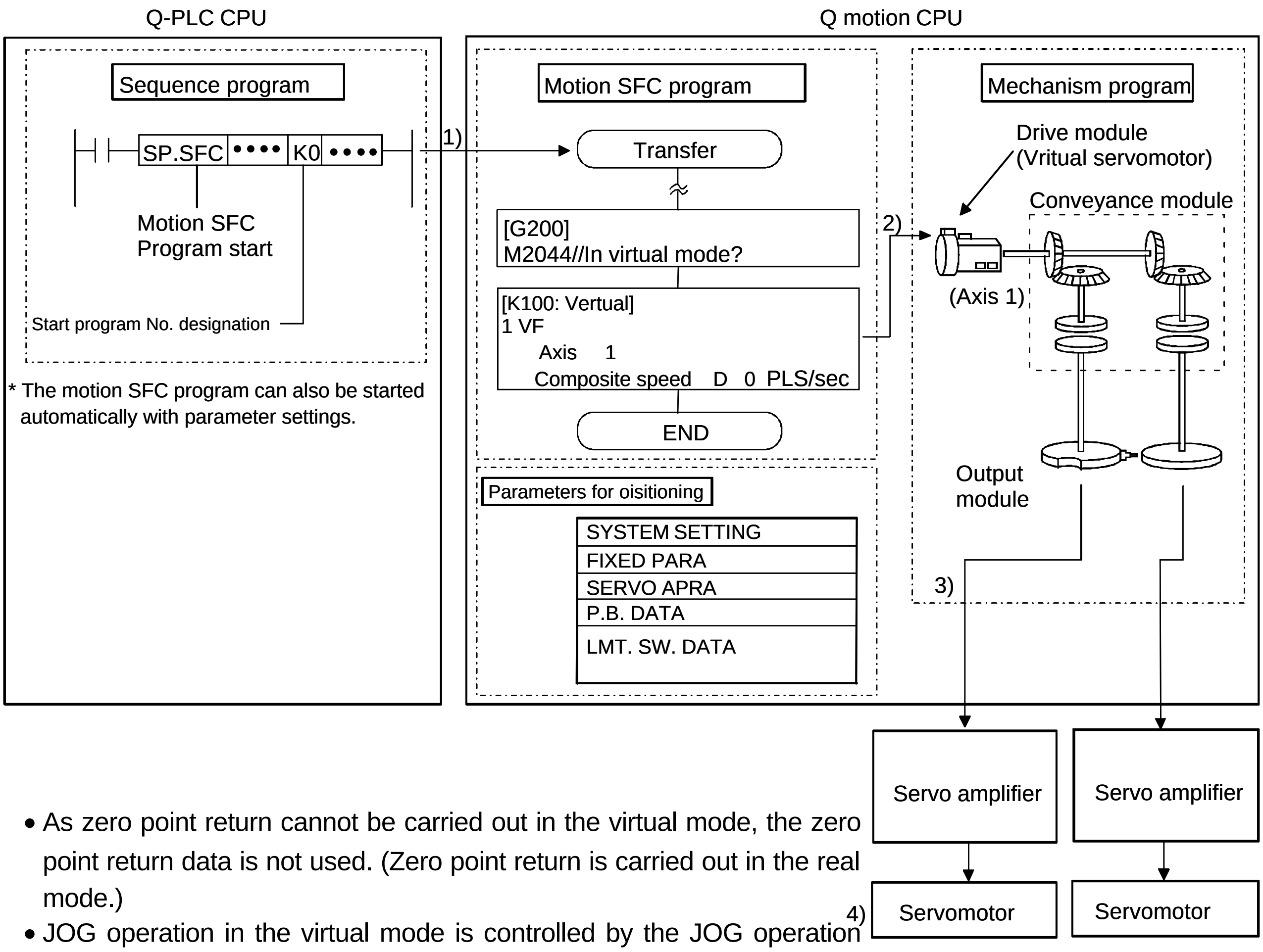
1.2.2 Virtual mode control for SV22 automatic machine

(a) The virtual mode processes synchronous control with the software using a mechanism program structured with a virtual main shaft and mechanism module. By using the virtual mode, the synchronous control conventionally carried out with a mechanism such as the main shaft, gears and cam, can be used for positioning control using a servomotor.

(b) With the virtual mode, in addition to the positioning parameters, servo program and motion SFC program used in the real mode, a mechanism program must be prepared.

(c) The procedures for carrying out positioning control in the virtual mode are indicated below.

1) Start of the SFC program for the virtual mode is requested with the SFCS command in the sequence program.
↓
2) Start mechanism program's virtual servomotor
↓
3) Via the conveyance model, the operation results are output to the servo amplifier set in the output module
↓
4) The servomotor is controlled



- As zero point return cannot be carried out in the virtual mode, the zero point return data is not used. (Zero point return is carried out in the real mode.)
- JOG operation in the virtual mode is controlled by the JOG operation data set in the drive module parameters.
- The external synchronous encoder pulses can be input to a [synchronous encoder input unit or manual pulse generator input unit] to operate the mechanism program's synchronous encoder.

1.3 Items required to start up system

Always carry out the steps enclosed in the solid-line box.
Carry out the steps enclosed in the dotted box as necessary.

1	Motion controller device selection, system assembly, wiring	Select the devices such as the Q-PLC base, power supply unit, Q motion CPU, Q-PLC CPU, motion unit, servo amplifier, servomotor and cables. Assemble and wire the system.
2	Installation of software package into Windows personal computer	Install the software package (SNETP, GSV13P, GSV22P, CAMP, GX Developer, etc.).
3	Setting of Q-PLC CPU multi-CPU	Create with GX Developer.
4	Sequence program creation	Create with GX Developer.
5	Writing of data to Q-PLC CPU	Using PC write operations, write the sequence programs and PC parameters.
6	CAMP startup and cam creation	Create the cam when using it for the output module with SV22.
7	SV13, SV22 start up (Project control)	Start up the software package to be used, and carry out project control.
8	Creation of system settings	Create the system basic settings, multi-CPU settings, Q-PLC base, motion unit, servo amplifier, servomotor and axis No., etc., as the motion controller system.
9	Servo data creation • Fixed parameters • Servo parameters • Zero point return data • JOG operation data • Parameter block	• Set the unit setting, movement amount per pulse, and stroke limit value, etc. • Set the rotation direction and automatic tuning, etc. • Set the zero point return direction, method, address and speed, etc. • Set the JOG speed limit value and parameter block No., etc. • Set the speed limit value, acceleration/deceleration time and torque limit value, etc.
10	Servo data creation • Limit switch data	Set this only when using the limit switch output function.
11	Creation of motion SFC program	
12	Mechanism program creation	Create this when using SV22.
13	Connection of cable to Q motion CPU	The Windows personal computer uses the SSCI/F card (A30CD-PCF) or SSCI/F board (A30BD-PCF/A10BD-PCF). (Cable Q170CDCBL3M/ Q170BDCBL3M)
14	SSCNET communication task start	Start up SW6RN-SNETP.
15	Installation of OS into Q motion CPU	Install using the installation operations on the Servo Menu screen. (Carry out only once when structuring the system.)
16	Writing of data to Q motion CPU	Write the motion SFC program, servo data, servo program, mechanism program and cam data.
17	Resetting of Q-PLC CPU	Press the RESET switch on the Q-PLC CPU.
18	Running of Q-PLC CPU and Q motion CPU	Press the RUN switch on the Q-PLC CPU and Q motion CPU.

Chapter 2 Explanation of Functions

The system functions are explained in this chapter.

2.1 List of specifications

2.1.1 List of motion controller specifications

<table>
<tr><th colspan="2">Model
Comparison item</th><th>Q172CPU(N)</th><th>Q173CPU(N)</th></tr>
<tr><td colspan="2">Outline dimensions [mm]</td><td colspan="2">122.4 (H) × 27.4 (W) × 89.3 (D): Q172/Q173CPU
104.4 (H) × 27.4 (W) × 114.3 (D): Q172/Q173CPUN</td></tr>
<tr><td colspan="2">Number of control axes</td><td>8 axes</td><td>32 axes</td></tr>
<tr><td colspan="2">Manual pulse generator</td><td colspan="2">3 units</td></tr>
<tr><td colspan="2">INC synchronous encoder/
ABS synchronous encoder</td><td>8 units</td><td>12 units</td></tr>
<tr><td colspan="2">Tracking enable input
(clutch ON/OFF)</td><td>8 points</td><td>12 points</td></tr>
<tr><td rowspan="2">Operation cycle
(at default)</td><td>SV13</td><td>0.88ms/1 to 8 axes</td><td>0.88ms/1 to 8 axes
1.77ms/9 to 16 axes
3.55ms/17 to 32 axes</td></tr>
<tr><td>SV22</td><td>0.88ms/1 to 4 axes
1.77ms/5 to 8 axes</td><td>0.88ms/1 to 4 axes
1.77ms/5 to 12 axes
3.55ms/13 to 24 axes
7.11ms/25 to 32 axes</td></tr>
<tr><td rowspan="2">Main OS</td><td>Motion SFC compatible for transfer assembly (SV13)</td><td>SW6RN-SV13QD</td><td>SW6RN-SV13QB</td></tr>
<tr><td>Motion SFC compatible for automatic machine (SV22)</td><td>SW6RN-SV22QC</td><td>SW6RN-SV22QA</td></tr>
<tr><td rowspan="3">Peri-pheral software</td><td>Motion SFC compatible for transfer assembly (SV13)</td><td colspan="2">SW6RN-GSV13P, SW6RN-SNETP, SW6RN-DOSCP, SW3RN-DOCPRNP, SW20RN-DOCPRNP</td></tr>
<tr><td>Motion SFC compatible for automatic machine (SV22)</td><td colspan="2">SW6RN-GSV22P, SW3RN-CAMP, SW6RN-SNETP, SW6RN-DOSCP,
SW3RN-DOCPRNP, SW20RN-DOCPRNP</td></tr>
<tr><td>Digital oscilloscope</td><td colspan="2">SW6RN-DOSCP</td></tr>
<tr><td rowspan="6">General startup support software package MT Develop-er</td><td rowspan="3">SW6RNC-GSVPRO</td><td colspan="2">SW6RNC-GSV (General startup support software (CD-ROM) × 1 disk)
• Transfer assembly software: SW6RN-GSV13P
• Automatic machine software: SW6RN-GSV22P
• Cam data creation software: SW3RN-CAMP
• Digital oscilloscope software: SW6RN-DOSCP
• Communication system software: SW6RN-SNETP
• Document printing software: SW3RN-DOCPRNP, SW20RN-DOCPRNP
Personal computer working environment
<table>
<tr><td>OS</td><td>Japanese Windows NT4.0 (Service Pack2 and above)/Windows98</td><td>Japanese Windows2000</td></tr>
<tr><td>CPU</td><td>Pentium 133MHz or higher recommended</td><td>Pentium II 233MHz or higher recommended</td></tr>
<tr><td>Memory</td><td>32MB or more recommended</td><td>64MB or more recommended</td></tr>
</table>
• Required hard disk capacity: SW6RNC-GSV ... 51MB + SW6RNC-GSVHELP ... 108MB
(custom installation possible)
• Display: SVGA (resolution: 800 × 600 dots, display colors: 256) or higher
• Application software: Word97, Excel97 or Word2000, Excel2000 (required for document printing)</td></tr>
<tr><td colspan="2">SW6RNC-GSVHELP (Operation Manual (CD-ROM) × 1 disk)</td></tr>
<tr><td colspan="2">Installation Manual</td></tr>
<tr><td rowspan="3">SW6RNC-GSVSET</td><td colspan="2">SW6RNC-GSVPRO</td></tr>
<tr><td colspan="2">A30CD-PCF (SSC I/F card (PCMIA TYPE II 1CH card))</td></tr>
<tr><td colspan="2">Q170CDCBL3M (A30CD-PCF cable 3m)</td></tr>
<tr><td colspan="2">PLC software package</td><td colspan="2">GX Developer: SW□D5C-GPPW *1</td></tr>
</table>

*1: Use version 6 or higher for □.

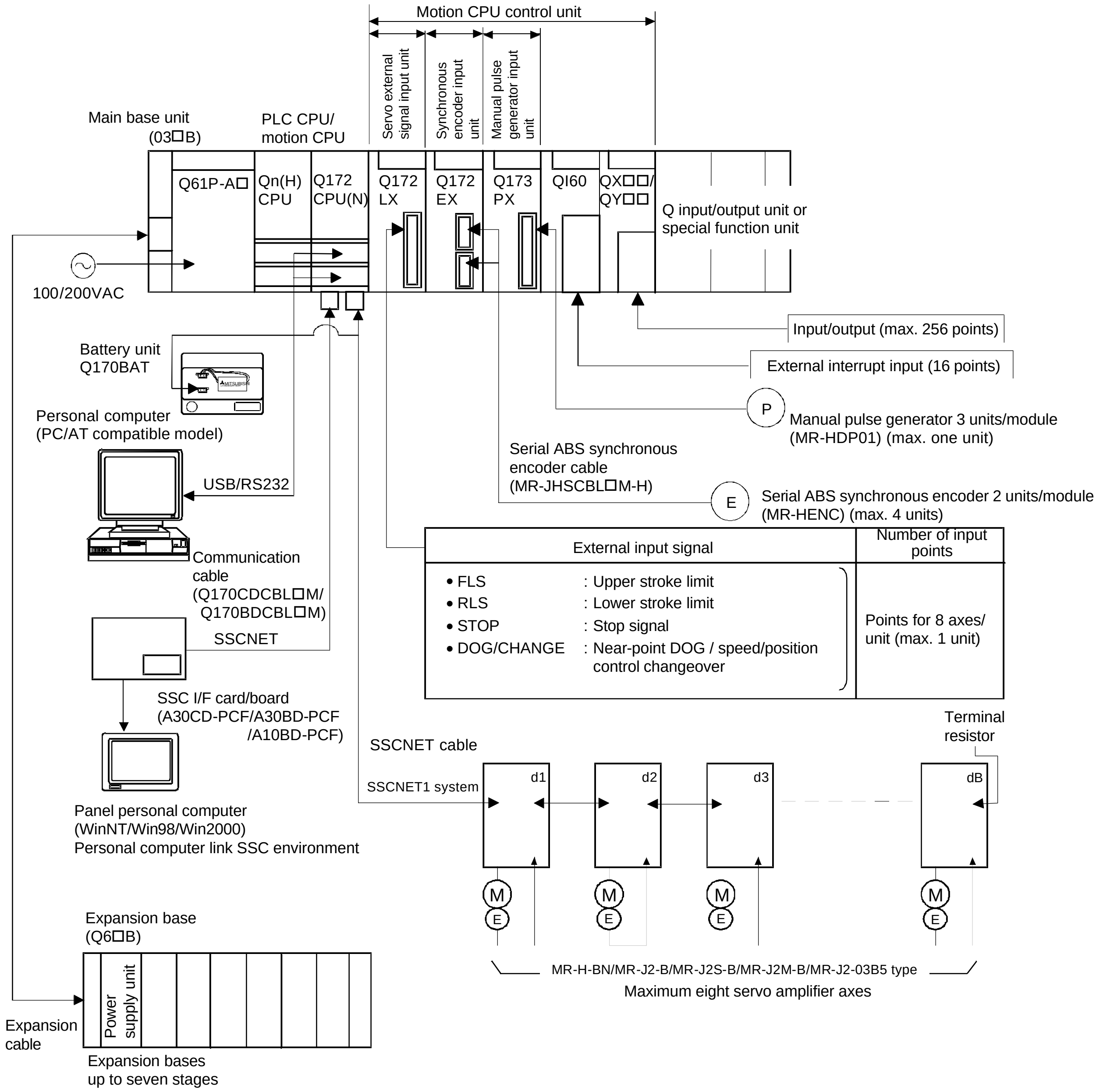
2.1.2 List of SFC performance specifications

Item				Q173CPU(N)/Q172CPU(N)
Program capacity	Code total (SFC diagram + operation control + transition)			287kB
	Text total (operation control + transition)			224kB
SFC program	Number of SFC programs			256 (No.0 to 255)
	SFC diagram size/program			Maximum 64kB (including SFC diagram comments)
	Number of SFC steps/program			Maximum 4094 steps
	Selective branches/branch			255
	Parallel branches/branch			255
	Parallel branching nest			Maximum 4 levels
Operation control program (F/FS), transition program (G)	Number of operation control programs			Total 4096 including F (one execution type)/ FS (scan execution type) (F/FS0 to F/FS4095)
	Number of transition programs			4096 (G0 to G4095)
	Code size/program			Maximum approx. 64kB (32766 steps)
	Number of blocks (lines)/program			Maximum 8192 blocks (for 4 steps (minimum)/block)
	Number of characters/block			Maximum 128 characters (including comments)
	Number of operators/block			Maximum 64 operators (operators: constant, word device, bit device)
	( ) nests/block			Maximum 32 levels
	Description method	Operation control program		Formula, bit condition expressions
		Transition program		Formula, bit condition expression, comparison condition expressions
Execution specifications	Number of simultaneous execution programs			Maximum 256 programs
	Number of simultaneous active programs			Maximum 256 steps/all programs
	Execution tasks	Normal task		Execute at motion main cycle
		Event task (masking possible)	Set period	Execute at each set period (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)
			External interrupt	Of interrupt unit QI60's 16 input points, execute at set input ON
			PLC interrupt	Execute with interrupt command from PLC
		NMI task		Of interrupt unit QI60's 16 input points, execute at set input ON
Input/output (X/Y) specifications	8192 points			
Number of actual input/ output (PX/PY) points	256 points			
Devices only in motion CPU	Number of internal relay (M) points			8192 points
	Number of latch relay (L) points			
	Number of link relay (B) points			8192 points
	Number of annunciator (F) points			2048 points
	Number of special relay (M) points			256 points
	Number of data register (D) points			8192 points
	Number of link register (W) points			8192 points
	Number of special register (D) points			256 points
	Number of motion device (#) points			8192 points
	Number of free run timer (FT) points			1 point (888µs)

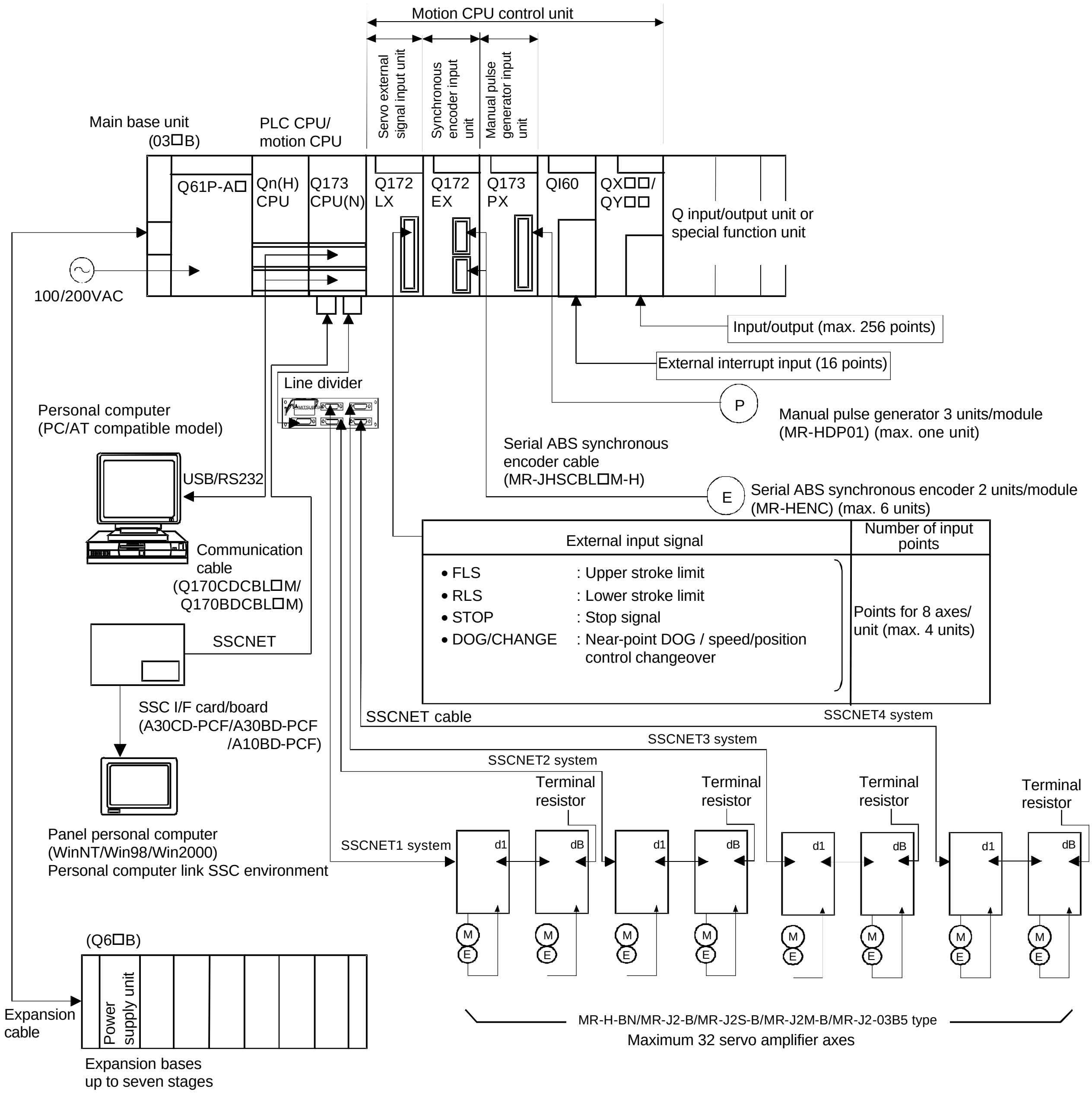
2.2 System configuration drawing

Refer to the User's Manual for details on the wiring.

2.2.1 Q172CPU(N) system



2.2.2 Q173CPU(N) system



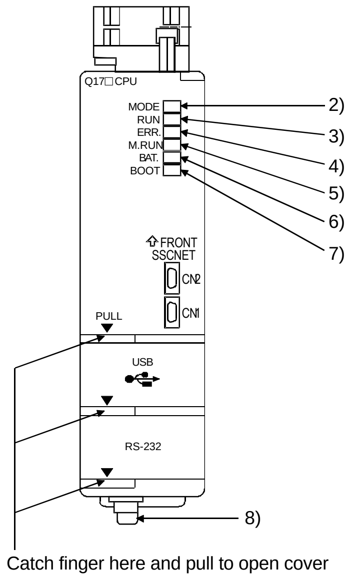
Remarks
To connect eight or more servo amplifier axes: (1) Use a line divider (Q173DV), or (2) Use a branch cable (Q173J2BΔCBL□M/Q173HBΔCBL□M)

2.3 Names of each part

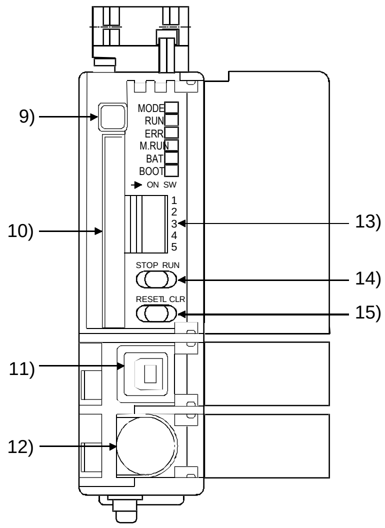
The names and applications of each Q172CPU(N)/Q173CPU(N) part are shown below.

• Q172CPU/Q173CPU

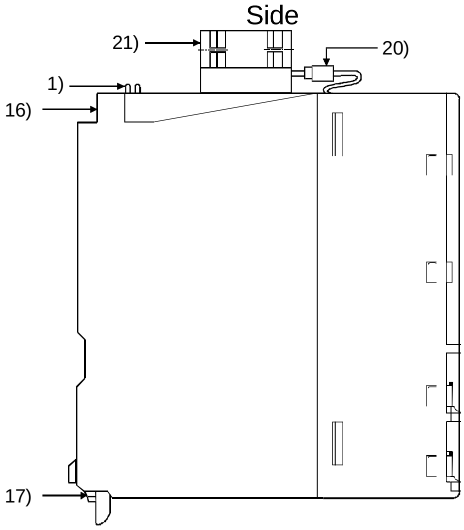
Front



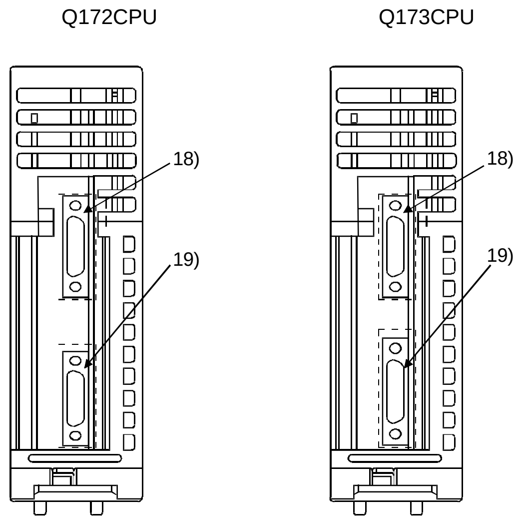
State with front cover opened



Side

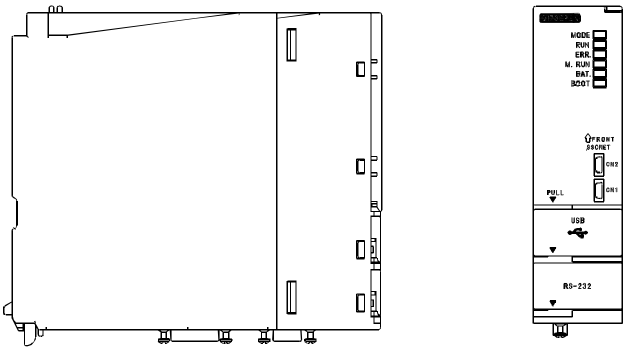


Bottom



• Q172CPUN/Q173CPUN

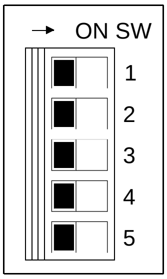
The names are the same as Q172CPU/Q173CPU.



Functions of each part

No.	Item	Function
1)	Module fixing hook	• Hook for fixing module onto base unit. (One-touch attachment)
2)	Mode judgment LED	• Green : Normal mode • Orange : Installation mode, ROM write mode
3)	RUN LED	• ON : Motion CPU normally started • OFF : Motion CPU error. Turns OFF when an error is found in the check before starting the motion CPU, or when a WDT error occurs.
4)	ERROR LED	• ON : This LED turns ON when the following errors occur (1) WDT error (2) System setting error (3) Servo error (4) Motion SFC error (5) Turns ON when a self-diagnosis error (excluding battery error) that does not stop the operation is detected. • Flicker : Flickers when a self-diagnosis error that stops operation is detected. • OFF : Normal
5)	MOTION RUN LED	• ON : Turns ON during motion control execution. • Flicker : Flickers at the start of latch clear. • OFF : Turns OFF when motion control is not being executed, and when a self-diagnosis error that stops operation is detected.
6)	BAT.ALARM LED	• ON : Turns ON when a battery error occurs. (When using external battery.)
7)	BOOT LED	• ON : During normal mode ROM operation • OFF : During normal mode RAM operation, installation or ROM write mode
8)	Module mounting lever	• Lever for mounting modules onto base unit.
9)	Memory card EJECT button	• Used to eject memory card.
10)	Memory card connection connector	• Connector for connecting memory card to CPU. (Use of memory card depends on software package.)
11)	USB connector	• Connector for connecting USB-compatible peripheral devices. (Connector type B) • Connect with USB-dedicated cable.
12)	RS-232 connector	• Connector for connecting peripheral devices. • Connect with RS-232 connection cable (QC30R2).

Q173CPU(N)/Q172CPU(N) switch and connector functions

No.	Item	Function	
13)	DIP switch 	DIP switches 1	Use prohibited (OFF at shipment from maker)
		DIP switches 2	ROM operation setting (OFF at shipment from maker) SW3 SW2 OFF OFF → RAM operation mode OFF ON → Setting prohibited ON OFF → Setting prohibited ON ON → ROM operation mode
		DIP switches 3	
		DIP switches 4	Use prohibited (OFF at shipment from maker)
		DIP switches 5	Install/ROM write switches ON : Install/ROM write mode OFF : Normal mode (RAM operation mode/ROM operation mode) This switch is turned ON in the following cases: • When installing the CPU module operating system (OS) from a peripheral device. • When writing the programs and parameters stored in the RAM into the ROM for ROM operation. To change the mode, change the switch setting and then restart the system.
14)	RUN/STOP (momentary switch)	• Use this switch by setting it to RUN or STOP. RUN : The motion program is executed. STOP : The motion program is stopped.	
15)	RESET/L.CLR switch*1 (momentary switch)	• RESET : The hardware is reset when the switch is set to the RESET side once. When an operation error occurs, the error is reset, and the operation is initialized. • L.CLR : All data in the latch area, set with the parameters, is cleared (set to OFF or 0). (Data other than that in the latch area is cleared simultaneously.) Latch clear operation methods (1) Set the RUN/STOP switch to STOP. (2) Set the RESET/L.CLR switch to the L.CLR side several times until the MOTION RUN LED flickers. (MOTION RUN LED flicker: Preparation for latch clear completed.) (3) Set the RESET/L.CLR switch to the L.CLR side again. (MOTION RUN LED turns OFF.)	
16)	Module fixing screw hole	• Screw hole used to fix module onto base unit. (M3×12 screw: prepared by user)	
17)	Module fixing projection	• Projection used to fix module to base unit.	
18)	CN2 connector	• Connector for establishing SSCNET connection with personal computer.	
19)	CN1 connector*2	• Connector for connecting with MR-H-BN/MR-J2S-B/MR-J2-B/MR-J2M-B/MR-J2-03B5.	
20)	Cooling fan connector*3	• Connector for connecting Q172CPU/Q173CPU dedicated cooling fan unit (Q170FAN).	
21)	Cooling fan unit*4	• Q172CPU/Q173CPU dedicated cooling fan unit (Q170FAN)	

*1: With the multi-CPU system, the QCPU/motion CPU for units No. 2 to No. 4 cannot be reset independently. When reset, MULTI CPU DOWN (error code: 7000) will occur in the other machines, and the entire multi-CPU system will stop. To reset the entire system, reset the No. 1 unit's QCPU.
*2: When using the Q173, the signals for the SSCNET1 to 4 systems are input in the CN1 connector. These must be branched to each system using a line divider Q173DV or a branch cable (A173J2BΔCBL□M/Q173HBΔCBL□M).
*3: Do not remove the caution plate until the cooling fan unit (Q170FAN) is used.
*4: Conditions for using cooling fan unit (Q170FAN):

Controller peripheral temperature / Number of Q motion CPUs in use	40°C or less	40°C or more, 55°C or less
One Q172CPU/Q173CPU module	Cooling fan not required	Cooling fan required
Two or more Q172CPU/Q173CPU modules	Cooling fan required	
Q172CPUN/Q173CPUN	Cooling fan not required	

Point

1) Turn the power OFF before setting the install switch.
2) After setting the switch, turn the power ON and check the switch state.
3) In the default state, the switch is set as shown above. (■) indicates the setting.
4) After setting the switch. Turn the power ON and check the switch state.

MEMO

Chapter 3 Q-PLC Multi-CPU

Using the sequence program, the input/output unit and special function unit sequence control is executed, and operations are executed with the applicable commands and dedicated commands.
The SFCS (motion SFC start request) command to start the motion SFC program, GINT change command to execute the interrupt against motion CPU, DDRD and DDWR commands to read/write the device directly from/to the motion CPU, and the SVST commands, CHGA current value change, CHGV speed change and CHGT torque change commands to request the servo program for start are executed.

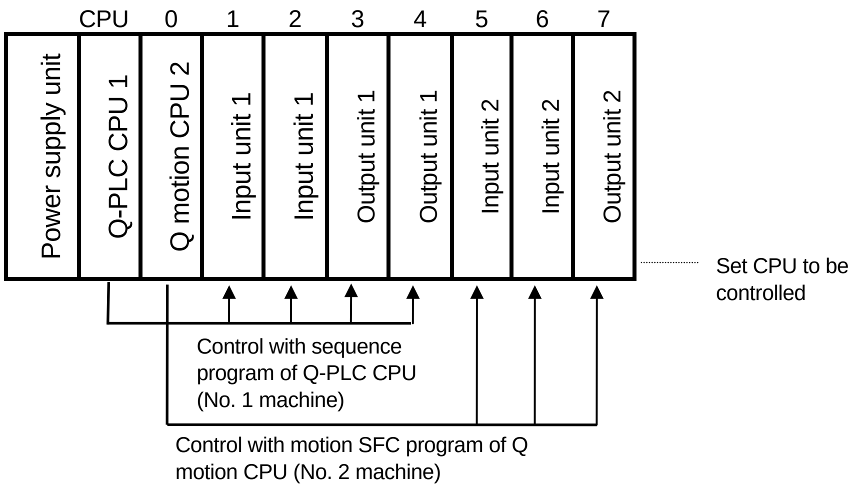
The Q172 specifications are explained in this chapter.
(For details on the SVST, CHGA, CHGV and CHGT commands, refer to Appendix 5.)

3.1 Multi-CPU system

The multi-CPU system is configured by mounting multiple Q-PLC CPUs/Q motion CPUs (maximum, 4 units) on the main base unit, and is used to control the input/output unit and intelligent function unit using each Q-PLC CPU/Q motion CPU.
Since the complicated servo control is executed by the Q motion CPU, and the other mechanical control and information control are executed by the Q-PLC CPU, it is possible to distribute the processing load.

3.1.1 Setting the multi-CPU system

For the multi-CPU system, it is necessary to set (control CPU setting) which input/output unit and intelligent function unit are controlled by which Q-PLC CPU/Q motion CPU as well as the number of Q-PLC CPUs/Q motion CPUs to be mounted. This must be set for each Q-PLC CPU/Q motion CPU.
(The operation procedure for multi-CPU setting are explained in the section 8.4.2.)



When initialization is executed, the Q motion CPU compares the parameters shown in the following table with the parameters in the No. 1 machine's Q-PLC. If the parameters do not match, they must be changed as shown below to prevent errors.

<table>
<tr><th rowspan="2">No.</th><th rowspan="2" colspan="2">Comparison item</th><th colspan="4">Parameter</th><th rowspan="2">Remarks</th></tr>
<tr><th colspan="2">Name for Q motion CPU</th><th colspan="2">Name of Q-PLC CPU</th></tr>
<tr><td>1</td><td colspan="2">Unit control CPU machine</td><td colspan="2">Motion slot setting</td><td rowspan="4">I/O assignment setting</td><td>Control CPU</td><td>Only the unit No. set on the Q motion CPU side is compared.</td></tr>
<tr><td>2</td><td colspan="2">Total number of bases</td><td colspan="2" rowspan="3">Base setting</td><td rowspan="3">Basic setting</td><td rowspan="3">Not compared when the base setting is not executed on the Q-PLC CPU side.</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Base unit</td><td>Base No.</td></tr>
<tr><td>Number of base slots</td></tr>
<tr><td>4</td><td colspan="2">Number of CPUs</td><td rowspan="3">Multi-CPU setting</td><td>Number of multi-CPUs</td><td rowspan="3">Multi-CPU setting</td><td>Number of CPUs</td><td></td></tr>
<tr><td>5</td><td colspan="2">Operation mode at CPU stop error</td><td>Operation mode</td><td>Operation mode</td><td></td></tr>
<tr><td>6</td><td colspan="2">Number of automatic refresh points</td><td>Automatic refresh setting</td><td>Automatic refresh setting</td><td></td></tr>
</table>

3.1.2 Mounting position of Q-PLC CPU/Q motion CPU

It is possible to mount up to four Q-PLC CPU/Q motion CPUs in the CPU slots (located at right side of power supply unit) up to the slot No. of main base unit sequentially.
It is not possible to leave an open slot between the Q-PLC CPU and Q-PLC CPU, between Q-PLC CPU and Q motion CPU, or between Q motion CPU and Q motion CPU.
Group and mount the Q motion CPU in the right-hand slot of Q-PLC CPU.
(The Q-PLC CPU cannot be mounted at the right of the Q motion CPU.)
(The personal computer CPU can be mounted at the right of the Q motion CPU.)

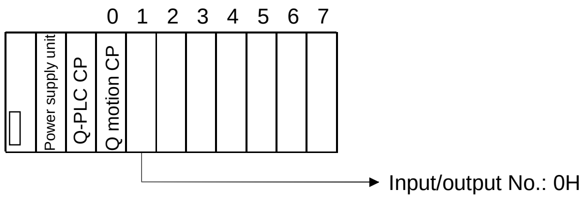
Mounting position of Q-PLC CPU/Q motion CPU

Number of CPUs	Mounting position of Q-PLC CPU/Q motion CPU	
2	CPU 0 1 2 Powersupply unit / Q-PLC CPU / Q motion CPU / /	
3	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q-PLC CPU / Q motion CPU /	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q motion CPU / Q motion CPU /
	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q-PLC CPU / Personal computer CPU	
4	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q-PLC CPU / Q-PLC CPU / Q motion CPU	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q-PLC CPU / Q motion CPU / Q motion CPU
	CPU 0 1 2 Power supply unit / Q-PLC CPU / Q motion CPU / Q motion CPU / Q motion CPU	CPU 0 1 2 3 Power supply unit / Q-PLC CPU / Q-PLC CPU / Q motion CPU / Personal computer CPU
	CPU 0 1 2 3 Power supply unit / Q-PLC CPU / Q motion CPU / Q motion CPU / Personal computer CPU	

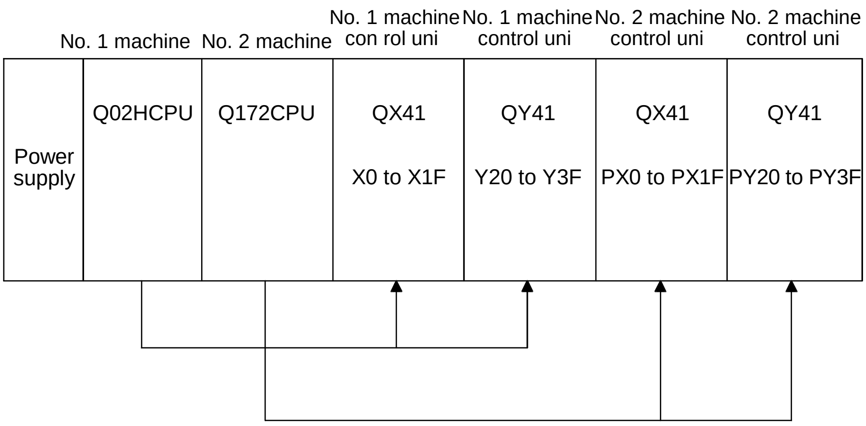
3.1.3 Input/output numbers

With the multi-CPU system, the slots equivalent to the number of CPUs set in the PC parameter multi-CPU setting are occupied by the Q-PLC CPU/Q motion CPU.
The input/output numbers are assigned sequentially to the right with the input/output unit and intelligent function unit (mounted at the right slot occupied by the Q-PLC CPU/ Q motion CPU) assigned as "0H".

Q-PLC CPU: When the number of PLC CPUs is set to "2 units"



The Q motion CPU I/O No. is independent of that of Q-PLC CPU. It is the I/O No. set by Q motion CPU system setting. (The I/O No. of a unit controlled by the Q motion CPU is indicated as "PX/PY".)
Note that the I/O No. of a Q motion CPU control unit assigned in "I/O assignment of Q-PLC" is invalid even if it is assigned.
It is basically recommended to execute such setting that it is serial in all CPUs.

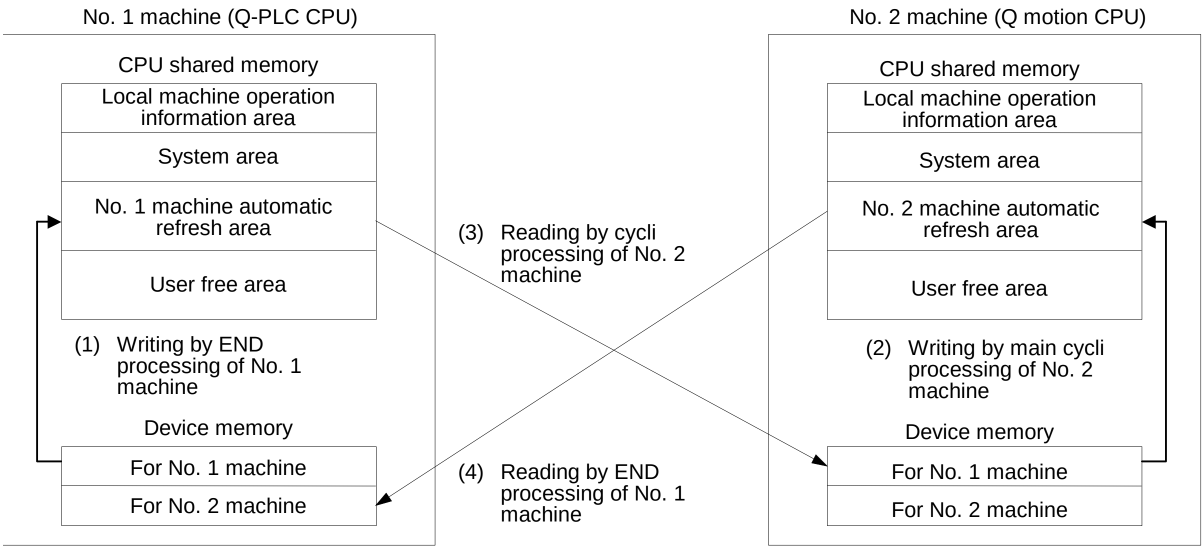


When assigning the Q-PLC CPU I/O, set the Q motion CPU control unit as shown in the following table. (The Q172LX and Q172EX are treated to occupy the 32 intelligent function unit points in the Q-PLC CPU.)

Unit	Type	Number of points	Remarks
Input unit	Input	Set depending on unit used	• Set the control CPU to the machine corresponding to Q motion CPU. • The type and number of points do not need to be set.
Output unit	Output	Set depending on unit used	
Input/output mixed unit	Input/output mixed	Set depending on unit used	
Interrupt unit (Q160)	Interrupt	16 points	
Q172LX	Intelligent	32 points	
Q172EX	Intelligent	32 points	

3.1.4 Automatic refresh for shared memory

(1) With automatic refresh of the CPU shared memory, the transmission/reception of data between each CPU of multi-CPU system is executed automatically by the Q-PLC CPU during END processing, and by the Q motion CPU during main cyclic processing (dead time other than motion control) respectively.
Since the data is read automatically from the device memory of other machines when the automatic refresh is used, it is possible to use the device data of other machines as device data of local machine.



.
The following shows the outline operation in case when the No. 1 machine executes the automatic refresh for 32 points of B0 to B1F and the No. 2 machine executes the 32 points of B20 to B3F.
Contents of processing (END processing of No. 1 machine)

(1) Shift of transmission device (B0 to B1F) data for No. 1 machine to automatic refresh area of shared memory of local machine
(4) Shift of automatic refresh area data within shared memory of No. 2 machine to B20 to B3F of local machine

Contents of processing (main cyclic processing of No. 2 machine)

(2) Shift of transmission device (B20 to B3F) data for No. 2 machine to automatic refresh area of shared memory of local machine
(3) Shift of automatic refresh area data within shared memory of No. 1 machine, to B0 to B1F of local machine

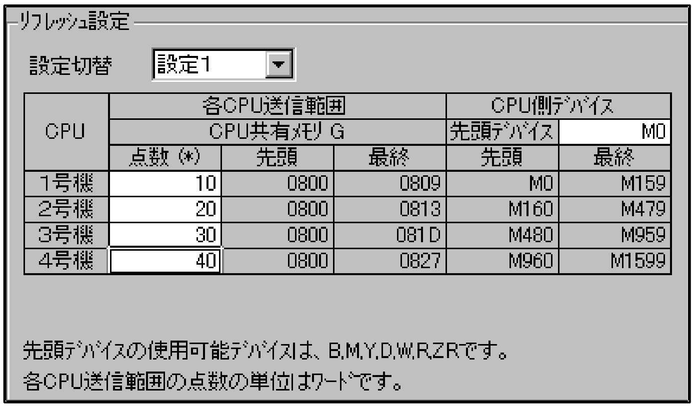
(2) To execute automatic refresh, it is necessary for Q-PLC CPU with the multi-CPU setting of PC parameter, and for the Q motion CPU with the multi-CPU setting of basic setting to set the number of points transmitted by each CPU, and the device (device used to execute the automatic refresh) to store the data.
The head device can be set in the following two ways.

1) Auto setting
   - When the applicable device is set at the head device setting column (★1) of each automatic refresh setting, the head device of each CPU is set automatically with the device as the head.

2) Manual setting
   - When "*" is set at the head device setting column (★1) of each automatic refresh setting, the head device of each CPU can be set at the (★2) column optionally.
   - It is possible to set a 'DUMMY' at the head device (★2) of a machine other than the local machine.
     To set the 'DUMMY', set the "*". Note that the automatic refresh is not executed for machine to which the 'DUMMY' is set.

CPU	Transmission range of each CPU			CPU side device	
	Shared memory G of CPU			Head device	★1
	No. of points (*)	Head	Final	Head	Final
No. 1 machine				★2	
No. 2 machine				★2	
No. 3 machine				★2	
No. 4 machine				★2	

The following shows a setting example of automatic refresh for outline operation.

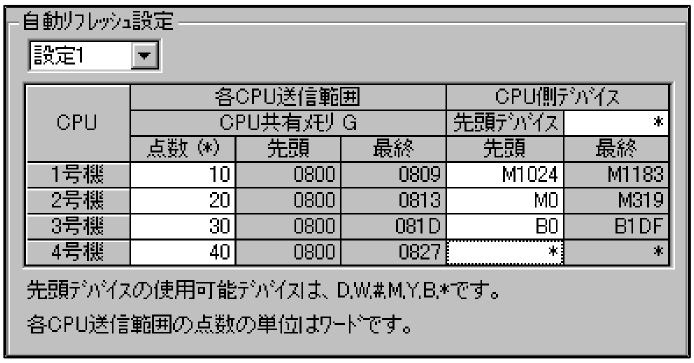

ﾘﾌﾚｯｼｭ設定

設定切替 設定1

CPU	各CPU送信範囲			CPU側ﾃﾞﾊﾞｲｽ	
	CPU共有ﾒﾓﾘ G			先頭ﾃﾞﾊﾞｲｽ	M0
	点数 (*)	先頭	最終	先頭	最終
1号機	10	0800	0809	M0	M159
2号機	20	0800	0813	M160	M479
3号機	30	0800	081D	M480	M959
4号機	40	0800	0827	M960	M1599

先頭ﾃﾞﾊﾞｲｽの使用可能ﾃﾞﾊﾞｲｽは、B,M,Y,D,W,R,ZRです。
各CPU送信範囲の点数の単位はﾜｰﾄﾞです。

No. 1 machine (Q-PLC CPU)


自動ﾘﾌﾚｯｼｭ設定

設定1

CPU	各CPU送信範囲			CPU側ﾃﾞﾊﾞｲｽ	
	CPU共有ﾒﾓﾘ G			先頭ﾃﾞﾊﾞｲｽ	*
	点数 (*)	先頭	最終	先頭	最終
1号機	10	0800	0809	M1024	M1183
2号機	20	0800	0813	M0	M319
3号機	30	0800	081D	B0	B1DF
4号機	40	0800	0827	*	*

先頭ﾃﾞﾊﾞｲｽの使用可能ﾃﾞﾊﾞｲｽは、D,W,#,M,Y,B,*です。
各CPU送信範囲の点数の単位はﾜｰﾄﾞです。

No. 2 machine (Q motion CPU)

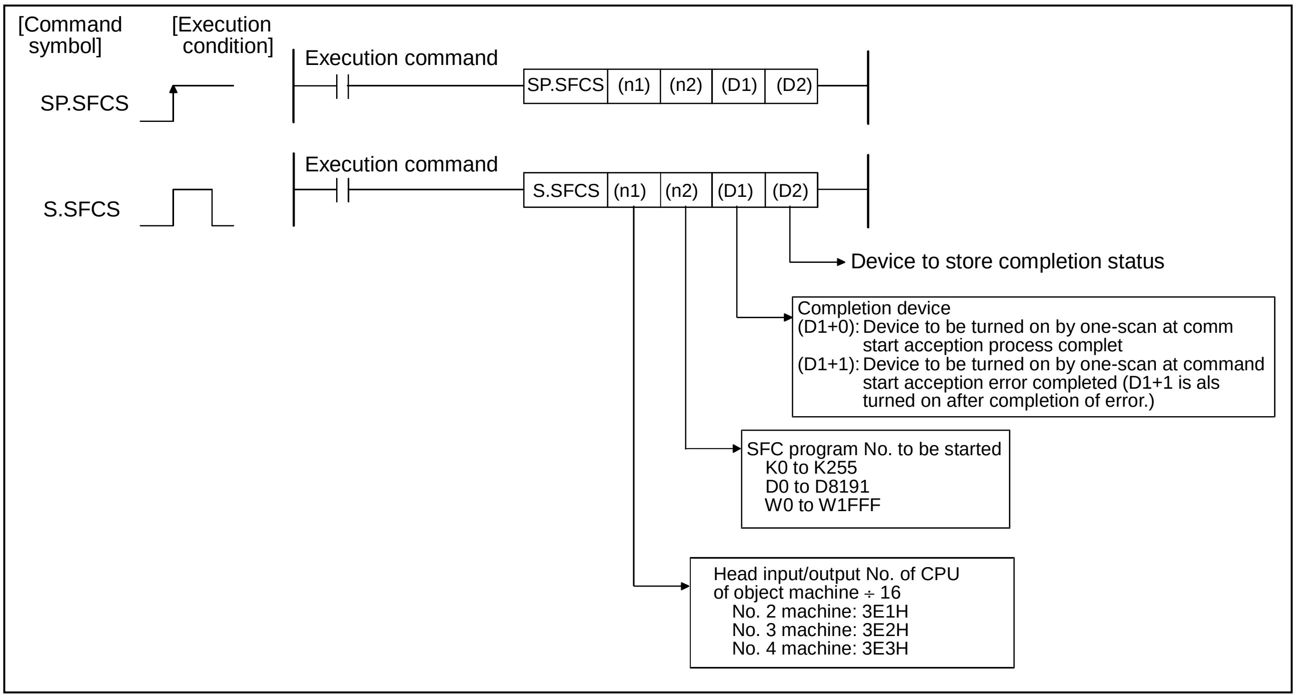
(The operation procedure for automatic refresh setting is explained in the sections 8.4.2 and 9.3.)

3.2 Multi-CPU motion dedicated commands

The multi-CPU's dedicated commands (SFCS, GINT, DDRD, DDWR) are explained in this section.

3.2.1 SFCS motion SFC program start command

The SFCS (SFC start) command is used to start the designated SFC program.



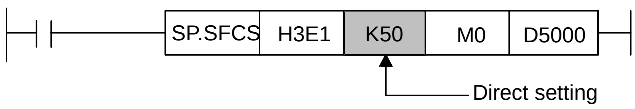
(1) Setting the SFC program No.

The SFC program No. can be set directly or indirectly.

(a) When setting directly, the SFC program No. is set as a direct numeric value (k0 to k255).

Example

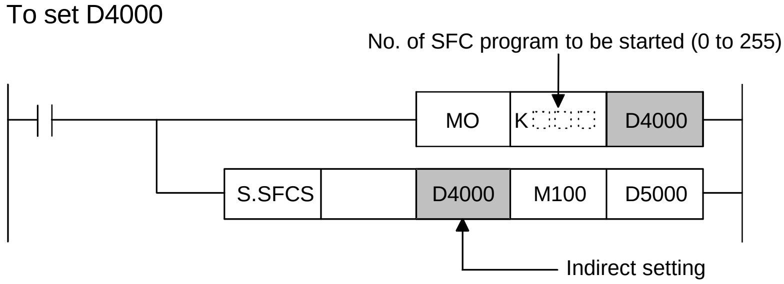
SFC program No. 50 is set as shown below.



(b) When setting indirectly, the SFC program No. is set with the word device (D0 to D8191, W0 to W1FF) details.

Example

To set D4000

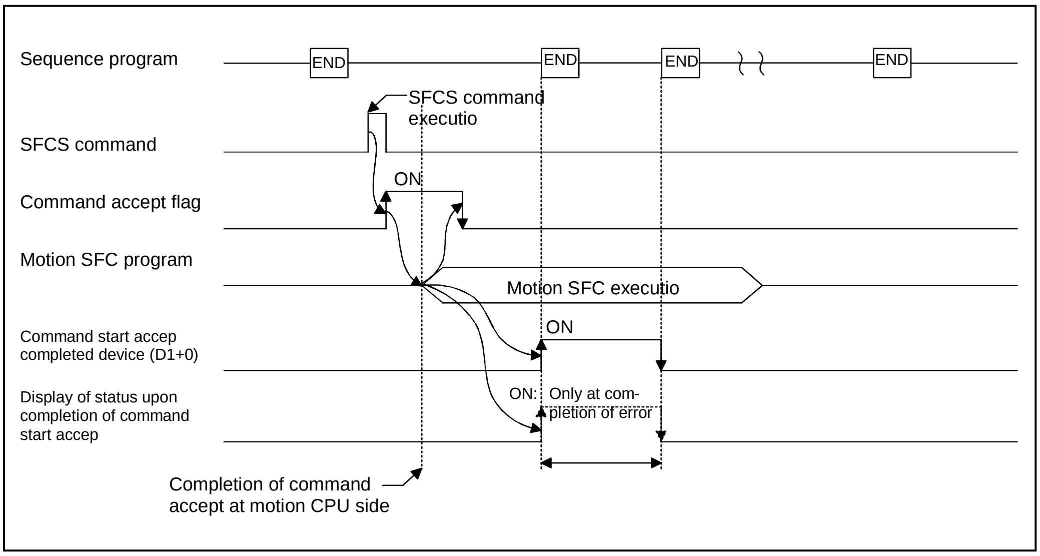


(2) Execution timing

Starting of the designated SFC program is requested at the rising edge (OFF → ON) of the SFCS command.

The SFC program to be started can be any task setting in the Normal task execution or NMI task execution.

It is effective at all times including the real mode, virtual mode and during mode switching.



(3) Operation error conditions

The operation error will occur in the following cases, and the SFCS command will not be executed.

(a) When the machine reserved by the head input/output No. of target machine CPU ÷ 16 (nl) is designated

(b) When the local machine is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(c) When a CPU other than the Q motion CPU is designated by the head input/ output No. of target machine CPU ÷ 16 (nl)

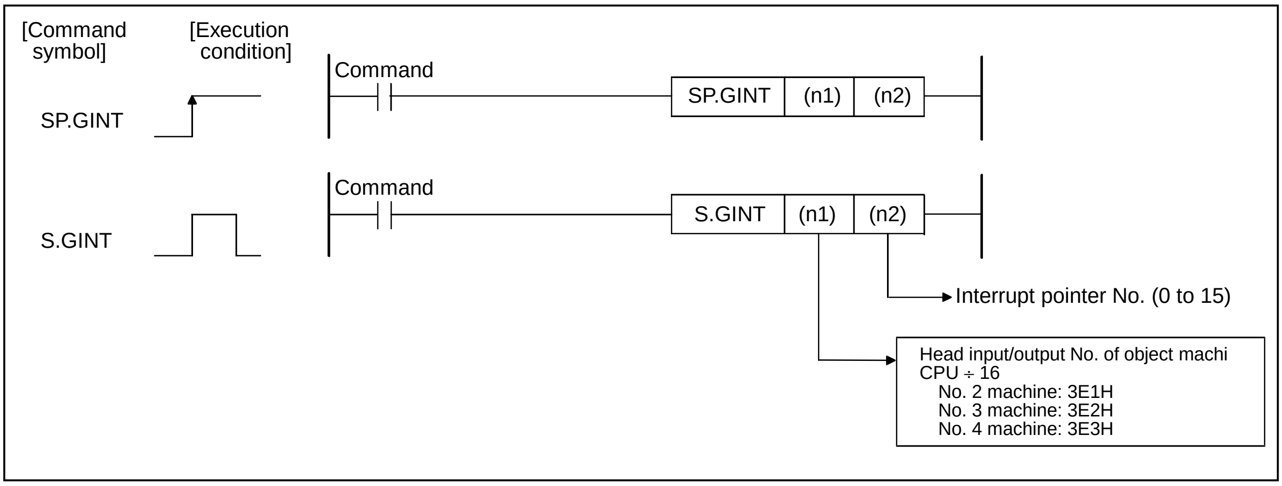
(d) When the designated command name is incorrect

(e) When the command is made up of a device other than the applicable device

(f) When 0 to 3DFH, 3E4H and following are designated by the head input/ output No. of target machine CPU ÷ 16 (nl)

3.2.2 GINT interrupt command to other machine’s CPU

The command is used to generate an interrupt to the Q motion CPU.

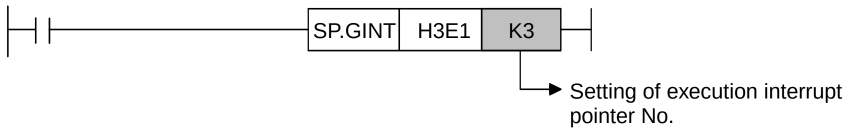


(1) Setting the GINT command interrupt pointer No.

Set the interrupt pointer No. directly by numerical value (K0 to K15).

Example

The interrupt pointer No.3 is set as follows.

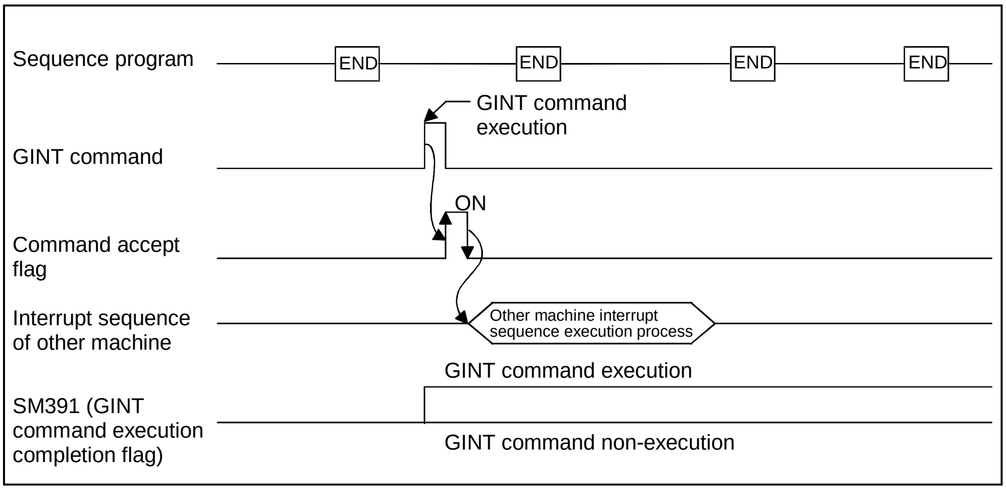


(2) Execution timing

An interrupt is generated to the Q motion CPU at the rising edge (OFF → ON) of the GINT command.

When an interrupt is generated from the Q-PLC CPU, the Q motion CPU starts a process with th "PLC interrupt" in respect to the active step of the SFC program.

It is effective at all times including the real mode, virtual mode and during mode switching. When the Q motion is in DI (interrupt disable), the event processing is not executed until the EI (interrupt enable) command is executed.



(3) Operation error conditions

The operation error will occur in the following cases, and the SFCS command will not be executed.

(a) When 0 to 3DFH, 3E4H and following are designated by the head input/output No. of target machine CPU ÷ 16 (nl)

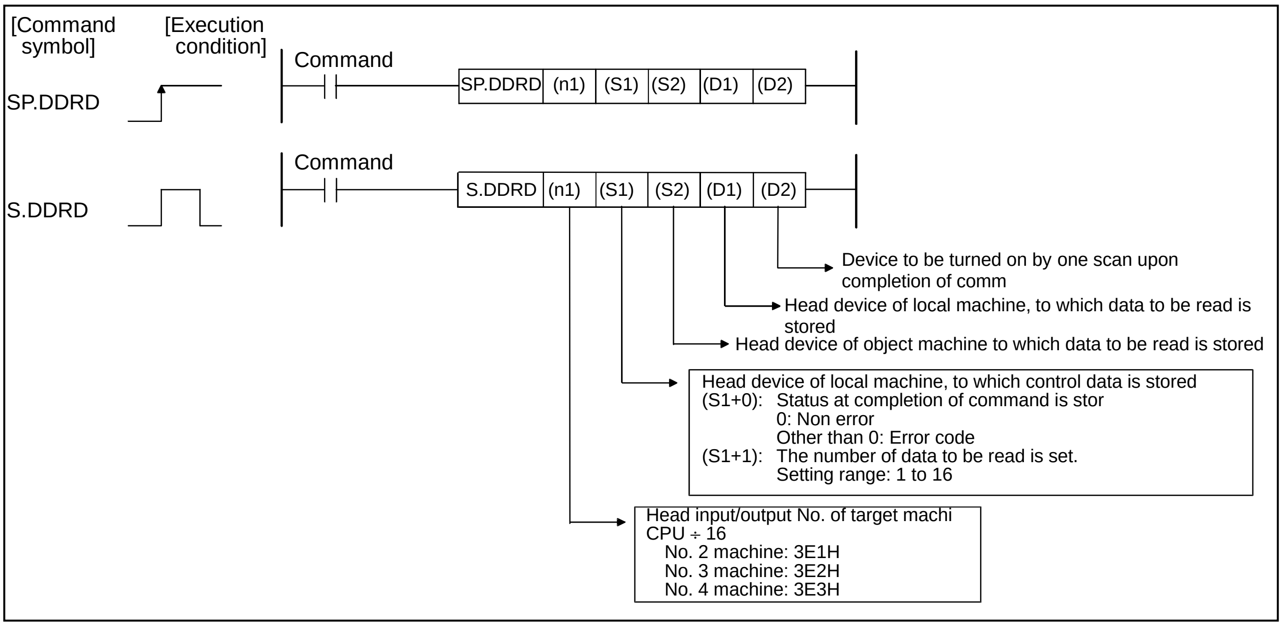
(b) When the local machine is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(c) When a CPU that does not support GINT command is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(d) When the machine reserved by the head input/output No. of target machine CPU ÷ 16 (nl) is designated

3.2.3 Read from DDRD Q motion CPU device command

The command is used to directly read the device data in the Q motion CPU with Q-PLC CPU.

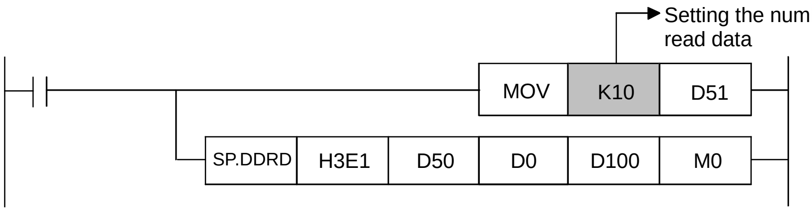


(1) Setting the number of data points to be read

Set the number of data points to be read indirectly.

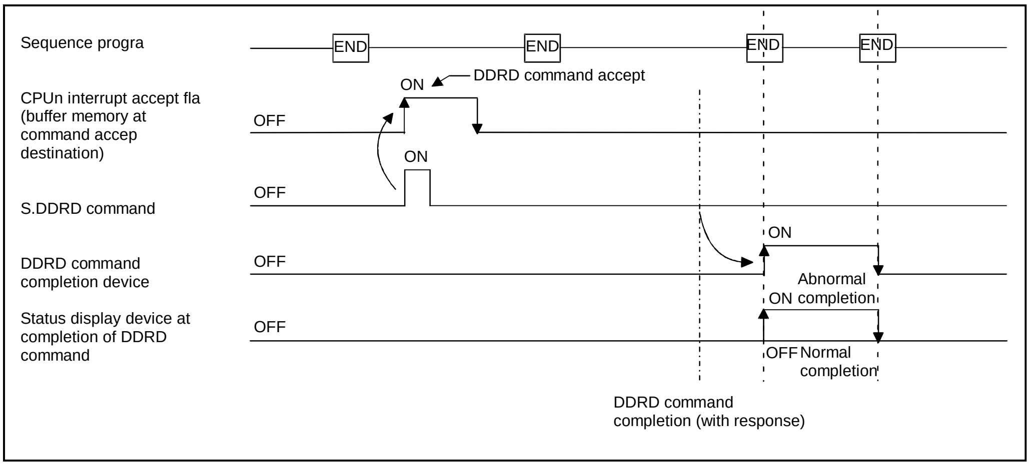
Example

To read the data for 10 points from D0 of No. 2 machine to D100 and following of local machine (No. 1 machine), it is executed as follows.



(2) Execution timing

The data of designated device is read to the Q motion CPU at the rising edge (OFF → ON) of the DDRD command execution command.



(3) Operation error conditions

The operation error will occur in the following cases, in which the DDRD command is not executed.

(a) When the local machine reserved by the head input/output No. of target machine CPU ÷ 16 (nl) is designated

(b) When the local machine is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(c) When a CPU other than the Q motion CPU is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(d) When the designated command name is incorrect

(e) When the command is made up of a device other than the applicable device

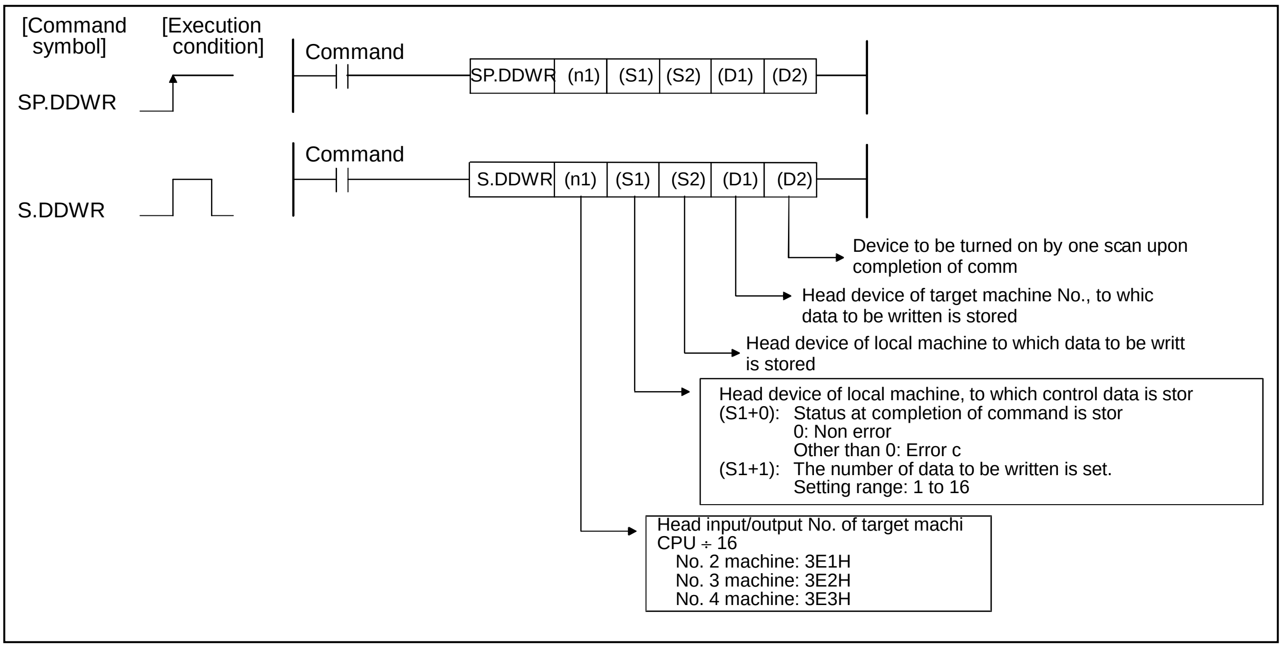
(f) When 0 to 3DFH, 3E4H and following are designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(g) When the number of read data items is other than 1 to 16

(h) When the number of read data items exceeds the range of the read data storage device

3.2.4 Write to DDWR Q motion CPU device command

This command is used to directly write the device data in the Q motion CPU to the Q-PLC CPU.

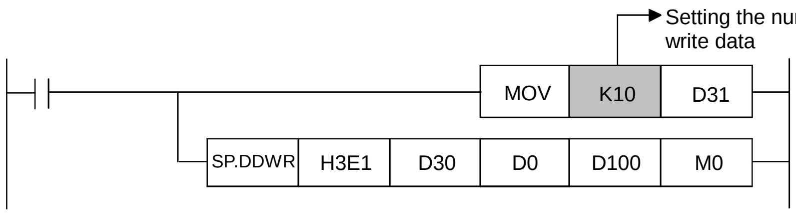


(1) Setting the number of data points to be written

Set the number of data points to be written, indirectly.

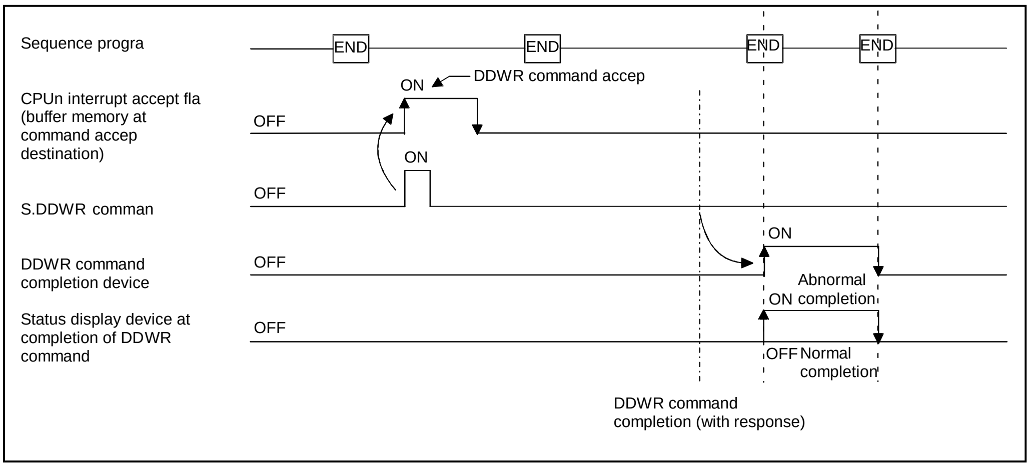
Example

To write the data for 10 points from D0 of local machine (No. 1 machine) to D100 and following of No. 2 machine, it is executed as follows.



(2) Execution timing

The data of designated device is written to the Q motion CPU at the rising edge (OFF → ON) of the DDWR command execution command.



(3) Operation error conditions

The operation error will occur in the following cases, and the DDWR command will not be executed.

(a) When the local machine set for reservation by the head input/output No. of target machine CPU ÷ 16 (nl) is designated

(b) When the local machine is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(c) When a CPU other than the Q motion CPU is designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(d) When the designated command name is incorrect

(e) When the command is made up of a device other than the applicable device

(f) When 0 to 3DFH, 3E4H and following are designated by the head input/output No. of target machine CPU ÷ 16 (nl)

(g) When the number of write data items is other than 1 to 16

(h) When the number of write data items exceeds the range of write data storage device

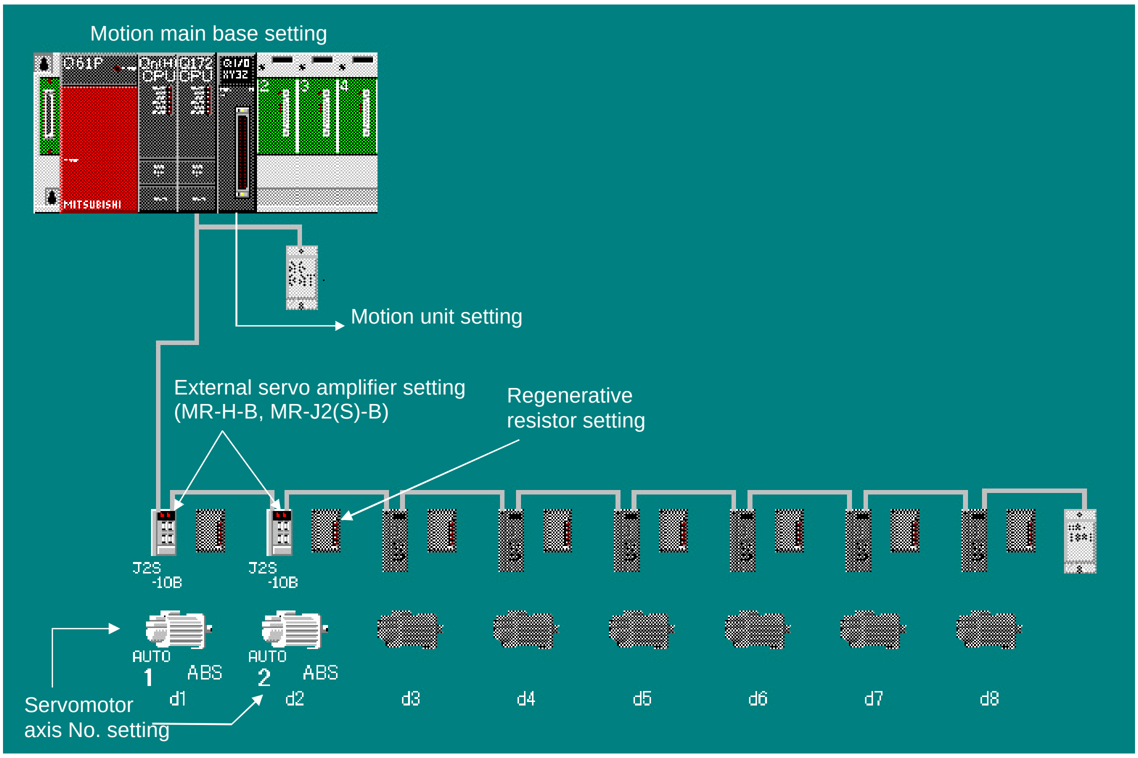
Chapter 4 Q Motion CPU

This CPU holds the system setting data and servo data, and executes the servo program and mechanism support language for multi-axis positioning.

4.1 System settings

This setting selects the base and units to be used, and determines the axis No. and the servo amplifier and servomotor types.

(1) An example of the Q172CPU(N) system settings is shown below. Refer to section 9.3 System settings in Chapter 9 for details on creating the screen.



4.2 Servo data

The following types of data are provided. Default values are set and must be changed to data that matches the system.
The data is stored in the motion CPU's memory area (SRAM battery backup).
One axis data and parameter block must be set.
The limit switch output data is created as required.

- System setting data
  - Basic system setting .... This is the basic operation data for Q motion CPU
  - Multi-CPU setting .... This is the data related to the multi-CPU
- Data
  - Axis data
    - Fixed parameter (Axis 1 to 8) .... This is fixed data determined according to the machine system for each axis.
    - Servo parameter (Axis 1 to 8) .... This is data determined by the specifications of the servomotor controlling each axis.
    - Zero point return data (Axis 1 to 8) .... This is data for the zero point return direction, method and speed, etc., for each axis.
    - JOG operation data (Axis 1 to 8) .... This is the data for the JOG speed limit and applicable parameter block No. for each axis.
  - Parameter block (Block No. 1 to 64) .... This is the zero point return data, JOG operation data and acceleration/deceleration time data used in the servo program.
  - Limit switch output data (maximum number of output points: 32) .... The ON/OFF pattern is output to the output devices (X, Y, M, L, B).

4.2.1 Basic system setting

The basic system setting contents are shown below.

No.	Item	Setting range	Default: Initial value	Default: Units	Remarks
1	Operation cycle setting	0: 0.8ms 1: 1.7ms 2: 3.5ms 3: 7.1ms 4: 14.2ms 5: Automatic setting	5	ms	• Set the motion operation cycle.
2	Operation setting for STOP → RUN	0: Turns on M2000 when the switch is changed from STOP to RUN. 1: Turns on M2000 when the switch is changed from STOP to RUN and the setting register (D704) is set to "1".	0	–	• Set such condition as to turn on the PLC READY flag (M2000).
3	Emergency stop input setting	0: No setting 1: X (PX) (0 to 1FFF) 2: M (0 to 1FFF)	0	–	• Set the bit device for emergency stop. • Bit device OFF → While emergency stop input is ON • Bit device ON → While emergency stop input is OFF
4	Latch range setting	Sets the range (latch (1)) that can be cleared using latch clear key and the range (latch (2)) that cannot be cleared using latch clear key.	–	–	• Set the latch range for devices (M, B, F, D, W) respectively.

4.2.2 Multi-CPU setting

The multi-CPU setting contents are shown below.

No.	Item	Setting range	Default		Remarks
			Initial value	Units	
1	Number of multi-CPUs	0: 2 1: 3 2: 4	0	–	• Set the total number of multi-CPUs including Q-PLC.
2	Automatic refresh setting for each multi-CPU	It is possible to set the devices (D, W, M, Y, B) by up to 2K words per each CPU against the setting (1 to 4).	–	–	• Automatic refresh setting for each CPU using shared memory of multi-CPU
3	Operation setting for CPU stop error	Note that the selective items depend on the number of CPUs set with the number of multi-CPUs.	–	–	• Select whether or not to stop the entire system at CPU stop error. Set for each CPU. • Always set for the No. 1 machine; selection cannot be canceled.

4.2.3 Fixed parameters

The fixed parameters to be set are shown below.

No.	Item		Setting range: mm – Setting range	mm – Units	inch – Setting range	inch – Units	degree – Setting range	degree – Units	PULSE – Setting range	PULSE – Units	Default: Initial value	Default: Units	Remarks
1	UNIT SETTING		0	–	1	–	2	–	3	–	3	–	• Set the command unit in positioning control for each axis.
2	Movement amount per pulse (A)	PULSE/ TURN (AP)	1 to 65535PLS								20000	PLS	• Set the number of feedback pulses per motor rotation.
3	Movement amount per pulse (A)	MOVE-MENT/ TURN (AL)	0.1 to 6553.5	µm	0.00001 to 0.65535	inch	0.00001 to 0.65535	degree	1 to 65535	PLS	20000	PLS	• Set the movement amount per motor revolution, which is determined by the mechanical system.
4	BACKLASH		0.0 to 6553.5	µm	0.0 to 0.65535	inch	0.0 to 0.65535	degree	0 to 65535	PLS	0	PLS	• Set the amount of backlash in the machine. • Every time the positioning direction changes during positioning, compensation by the backlash compensation amount is executed.
5	STROKE LIMIT MAX.		–214748364.8 to 214748364.7	µm	–21474.83648 to 21474.83647	inch	0 to 359.99999	degree	–2147483648 to 2147483647	PLS	2147483647	PLS	• Set the upper limit for the machine movement amount.
6	STROKE LIMIT MIN.		–214748364.8 to 214748364.7	µm	–21474.83648 to 21474.83647	inch	0 to 359.99999	degree	–2147483648 to 2147483647	PLS	0	PLS	• Set the lower limit for the machine movement amount.
7	CMD. IN-POS. RANGE		0.1 to 214748364.7	µm	0.00001 to 21474.83647	inch	0.00001 to 359.99999	degree	1 to 2147483647	PLS	100	PLS	• Set the position at which the command in-position signal M2403+20n is turned ON [(Positioning address) – (Feed present value)].

Setting example for "Movement amount per pulse"

<Conditions>

- The ball screw lead is 8mm, and the gear ratio is 1/2.
- Servomotor: HC-MFS13 (number of encoder output pulses: 131072 pls)

<Relative expression>

$$A = \frac{A_L}{A_P} = \frac{\text{Lead} \times \text{Gear ratio}}{\text{Number of encoder output pulses}} = \frac{8000 \times \frac{1}{2}}{131072} = \frac{4000}{131072}$$

Consequently, set the number of encoder pulses (A_P) per rotation to '131072', and the movement amount (A_L) per rotation to '4000.0' respectively.

4.2.4 Servo parameters

The parameters to be set are shown below.

[Basic servo parameters

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
1*	AMP. SETTING	Set automatically in accordance with the system settings										
2*	RESISTANCE											
3*	DYNAMIC BRAKE											
4*	MOTOR TYPE											
5*	MOTOR CAPACITY											
6	MOTOR REVOLUTION (R)											
7	FEEDBACK PULSE (N)											
8*	SPIN DIRECTION	0: Forward rotation (CCW) when the positioning address increases. 1: Reverse rotation (CW) when the positioning address decreases.								0	–	• Set the direction of rotation as seen from the load side. Forward rotation:  Reverse rotation: 
9	AUTO TUNING*[1]	0: VELOCITY 1: POS. & VEL. 2: NONE								1*[2]	–	• Set the gain (speed/position, speed) for executing automatic setting.
10	GAIN ADJUSTMENT MODE SELECTION*[3]	0: Interpolation mode 1: Auto tuning mode 1 2: Auto turning mode 2 3: Manual mode 1 4: Manual mode 2								1	–	• Sets the gain at which the automatic setting is executed. For details of each mode, refer to the following.
11	SERVO RESPONSE	1 to 12 (MR-H-B(N))/15(MR-J2S-B, MR-J2M-B)/5(MR-J2-B) (invalid during automatic tuning)								1	–	• Set in order to increase servo responsiveness.

*1: Cannot be set with MR-J2S-B and MR-J2M-B.
*2: For MR-J2(S)-B and WR-J2M-B, the default will be "2".
*3: Cannot be set with MR-H-B(N) and MR-J2-B.

[Gain adjustment mode selection]

Set the gain for executing automatic setting.

- Interpolation mode : Fixes the position control gain 1/speed control gain 1.
- Auto tuning mode 1 : Normal auto tuning
- Auto tuning mode 2 : Fixes the load inertia ratio.
- Manual mode 1 : Executes the simplified manual adjustment.
- Manual mode 2 : Executes the adjustment of all gains manually.

	Inter-polation mode	Auto tuning		Manual	
		Mode 1	Mode 2	Mode 1	Mode 2
LOAD INERTIA RATIO	○	○			
POSITION CONTROL GAIN 1		○	○		
VELOCITY LOOP GAIN 1		○	○	○	
POSITION LOOP GAIN 2	○	○	○	○	
VELOCITY LOOP GAIN 2	○	○	○		
VEL. INTEGRAL COMPS.	○	○	○		

○: Items to be set automatically

POINT

When the items marked with * in the above table are changed, reset the Q-PLC CPU or turn off the PLC READY (M2000) and turn it on again before turning the servo power supply on.

[Adjustment parameters]

When real-time auto tuning is enabled, the values (No.1 to No.6) are tuned and changed during test operation.

Read the values from the servo amplifier to the personal computer, and then write them in the Q172CPU(N) before turning OFF the system power supply.

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
1	LOAD INERTIA RATIO	(MR-H-B (N)/MR-J2-B) 0.0 to 100.0 (MR-J2S-B, MR-J2M-B) 0.0 to 300.0								3.0*[1]	–	• Set the ratio of moment of load inertia for the motor.
2	POSITION CONTROL GAIN 1	(MR-H-B (N)/MR-J2-B) Valid range 4 to 1000 rad/sec (MR-J2S-B, MR-J2M-B) Valid range 4 to 2000 rad/sec Setting range 1 to 9999 rad/sec								70*[2]	rad/sec	• Set to increase the follow-up with respect to the position command.
3	VELOCITY LOOP GAIN 1	(MR-H-B (N)/MR-J2-B) Valid range 20 to 5000 rad/sec (MR-J2S-B, MR-J2M-B) Valid range 20 to 8000 rad/sec Setting range 1 to 9999 rad/sec								1200*[3]	rad/sec	• Set to increase the follow-up with respect to the speed command.
4	POSITION LOOP GAIN 2	(MR-H-B (N)/MR-J2-B) Valid range 1 to 500 rad/sec (MR-J2S-B, MR-J2M-B) Valid range 1 to 1000 rad/sec Setting range 1 to 9999 rad/sec								25*[4]	rad/sec	• Set to increase the position response with respect to load disturbance.
5	VELOCITY LOOP GAIN 2	(MR-H-B (N)/MR-J2-B) Valid range 20 to 8000 rad/sec Setting range 1 to 9999 rad/sec (MR-J2S-B, MR-J2M-B) Valid range 20 to 20000 rad/sec Setting range 1 to 99999 rad/sec								600*[5]	rad/sec	• Set when vibration is generated, for example in machines with a large backlash.
6	VEL. INTEGRAL COMPS.	Valid range 1 to 1000 ms Setting range 1 to 9999 ms								20*[6]	ms	• Set the time constant for integral compensation.
7	NOTCH FILTER*[7]	0: Not used 1: 1125 2: 750 3: 562 4: 450 5: 375 6: 321 7: 281								0	Hz	• Set the frequency for the notch filter.
8	FEED FORWARD GAIN	0 to 100% 0: Feed forward control is not executed.								0	%	• Set the feed forward coefficient used in positioning control.
9	IN-POSITION RANGE*[8]	0.1 to 3276.7	μm	0.00001 to 0.32767	inch	0.00001 to 0.32767	degree	1 to 32767	PLS	100	PLS	• Sets the quantity of droop pulses in the deviation counter. • The in-position signal is ON when the number of droop pulses is within the set range. 1 ≤ (in-position range) × AP/AL·AM ≤ 32767
10	MAGNETIC BRAKE OUT	0 to 1000ms								100	ms	• Set the time delay between actuation of the electromagnetic brake and base disconnection.
11	MON. OUT. MODE (MON. 1)	(MR-H-B (N)/MR-J-B) 0: VEL. (±) 1: TRQ. (±) 2: VEL. (+) 3: TRQ. (+) 4: CUR CMD. OUT 5: CMD FΔT 6: DEV. PULSE 1/1 7: DEV. PULSE 1/4 8: DEV. PULSE 1/16 9: DEV. PULSE 1/32		(MR-J2-B) 0: VEL. (±) 1: TRQ. (±) 2: VEL.(+) 3: TRQ. (+) 4: CUR CMD. OUT 5: CMD FΔT 6: DEV. PULSE 1/1 7: DEV. PULSE 1/16 8: DEV. PULSE 1/64 9: DEV. PULSE 1/256 10: DEV. PULSE 1/1024		(MR-J2S-B, MR-J2M-B) 0: VEL. (±) 1: TRQ. (±) 2: VEL.(+) 3: TRQ. (+) 4: CUR CMD. OUT 5: CMD FΔT 6: DEV. PULSE 1/1 7: DEV. PULSE 1/16 8: DEV. PULSE 1/64 9: DEV. PULSE 1/256 10: DEV. PULSE 1/1024 11: Bus voltage				0	–	• Set the monitor items output as analogue outputs in real time.
12	MON. OUT. MODE (MON. 2)									1	–	
13	OPTIONAL FUNCTION 1 (CARRIER)*[9]	0: 2.25kHz (non low-noise operation) 3: 9kHz (low-noise operation)								0	kHz	• Set the "low noise" to improve the sound of the frequencies generated from the motor.
14	OPTIONAL FUNCTION 1 (ENCODER TYPE)	0: 2 LINES 1: 4 LINES								0	–	• Set the type of encoder cable.
15	OPTIONAL FUNCTION 1 (EXT. EMRG STOP SIG)*[10]	0: USE 1: NO-USE								0	–	• To invalidate the external emergency stop signal (EMG) set "not used".

*[1]: For MR-J2 (S)-B and MR-J2M-B, the default is "7.0".
*[2]: For MR-J2S-B and MR-J2M-B, the default is "35".
*[3]: For MR-J2S-B and MR-J2M-B, the default is "117".
*[4]: For MR-J2S-B and MR-J2M-B, the default is "35".
*[5]: For MR-J2S-B and MR-J2M-B, the default is "817".
*[6]: For MR-J2S-B and MR-J2M-B, the default is "48".
*[7]: Setting not possible for MR-J2S-B and MR-J2M-B.
*[8]: The display of the possible setting range differs according to the electronic gear value.
*[9]: Cannot be set with MR-J2 (S)-B and MR-J2M-B.
*[10]: Cannot be set with MR-H-B (N).

[Adjustment parameters] Continued

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
16	OPTIONAL FUNCTION 2 (NON MOTOR SELECT)*9	0: NO 1: YES								0	–	• To check the status without connecting a motor, set "YES".
17	OPTIONAL FUNCTION 2 (MAGNETIC BRAKE INTERLOCK TIMING)*9	0: TIMING ARE AS FOLLOWS (NOT RELATED WITH MOTOR SPEED) • SERVO OFF • ALARM • EMG. STOP OFF (YES) 1: ABOVE MENTIONED STATUS & ZERO SPEED								0	–	• Set the interlock timing for the electromagnetic brake interlock signal.
18	OPTIONAL FUNCTION 2 (VIBRATION CONTROL)*10	0: NO 1: YES								0	–	• Set "YES" to suppress vibration on stopping.
19	OPTIONAL FUNCTION 2 (MOTOR LOCK)	0: NO 1: YES								0	–	• To carry out test operation without rotating the motor, set "YES".
20	NOTCH DEPTH*10, *11	0: –40dB 1: –14dB 2: –8dB 3: –4dB								0	dB	• Set the depth to decrease the gain.
21	NOTCH FREQUENCY *10, *11	0: NO 1: 4500 2: 2250 3: 1500 4: 1125 5: 900 6: 750 7: 642.9		8: 562.5 9: 500 10: 450 11: 409.1 12: 375 13: 346.2 14: 321.4 15: 300		16: 281.3 17: 264.7 18: 250 19: 236.8 20: 225 21: 214.3 22: 204.5 23: 195.7		24: 187.5 25: 180 26: 173.1 27: 166.7 28: 160.1 29: 155.2 30: 150 31: 145.2		0	Hz	• Set the frequency of notch filter.
22	LOW-PASS FILTER SELECTION *10, *11	0: YES 1: NO								0	–	• When set to "YES," a filter in the band of speed control gain 2 setting value × 10 2π × (1+ Non-applicable inertia setting value is automatically set.)
23	ADAPTIVE VIBRATION CONTROL *10, *11	0: NO 1: YES 2: HOLD								0	–	• Set "YES" for section that always detects machine resonance frequency and generates filter corresponding to the resonance.
24	ADAPTIVE VIBRATION CONTROL SENSITIVITY *10, *11	0: NORMAL 1: HIGH SENSITIVITY								0	–	• Set the sensitivity at which the machine resonance is detected.

*9: Cannot be set with MR-J2 (S)-B and MR-J2M-B.
*10: Cannot be set with MR-H-B (N).
*11: Cannot be set with MR-J2-B.

[Option]

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
1	CLAMP SPEED 0	1 to 65535 r/min								1000	–	• Measure the clamp speed.

[Extended servo parameters]

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
1	MON. OUT. 1 OFFSET	(MR-H-B(N)) –9999 to 9999 mv				MR-J2(S)-B, MR-J2M-B –999 to 999 mv				0	mv	• Set the offset value for monitor output 1.
2	MON. OUT. 2 OFFSET	(MR-H-B(N)) –9999 to 9999 mv				MR-J2(S)-B, MR-J2M-B –999 to 999 mv				0	mv	• Set the offset value for monitor output 2.
3	BEF. ALRM. DATA SELECT (SAMPLING TIM SEL.)[*1]	0: 1.77 1: 3.55 2: 7.11 3: 14.22 4: 28.44								0	ms	• Set the analogue data output when an alarm occurs.
4	BEF. ALRM. DATA SELECT (DATA SEL. 1)[*1]	0: VEL. (±) 1: TRQ. (±) 2: VEL. (+) 3: TRQ. (+) 4: CUR CMD. OUT 5: CMD FΔT 6: DEV. PULSE 1/1 7: DEV. PULSE 1/4 8: DEV. PULSE 1/16 9: DEV. PULSE 1/32 10: DEV. PULSE 1/64								0	–	
5	BEF. ALRM DATA SELECT (DATA SEL. 2)[*1]									1	–	
6	ZERO SPEED	0 to 10000 r/min								10000	r/min	• Set the speed at which the motor speed is judged to be "0".
7	EX. ERROR ALARM LEVEL	1 to 1000 kPLS								8.0	kPLS	• Set the value at which an excessive droop pulses alarm is output.
8	CLOSED PLG. TURN	Refer to the Full Closed Control Reference Manual (IB-67316).										
9	ZERO STD. PLG											
10	OPTIONAL FUNCTION 5 (PI-PID CTRL. CHG.)	0: INVALID 1: CAN_CHG._BY_DROOP_WHEN_POS. CTRL 2: VEL, AMP, CTRL								0	–	• Set the conditions for PI-PID control changing.
11	OPTIONAL FUNCTION 5 (SERVO READ LANG)[*1]	0: JAPANESE 1: ENGLISH								0	–	• Set the display format for the parameter unit.
12	PI-PID CHANGE POSITION DROOP	0 to 50000 PLS								0	PLS	• Set the amount of position droop at the change to PI-PID control when position control is executed.
13	TORQUE COMPS. FACTOR[*1]	–19 to 9979								0	–	• Set to expand the torque control range up to the speed limit value in torque control.
14	VELOCITY DIFF. COMPS	0 to 1000								980	–	• Set the differential compensation value for the actual speed loop.
15	OPTIONAL FUNCTION-6 (COMMUNI-CATION BAUD RATE)[*2][*3]	0: 9600 bps 1: 19200 bps 2: 38400 bps 3: 57600 bps								0	bps	• Set the serial communication baud rate in case when the set-up software is used.
16	OPTIONAL FUNCTION-6 (COMMUNI-CATION RESPONSE DELAY TIME)[*2][*3]	0: NO 1: YES								0	–	• Set 'YES' for selection of communication response delay time when the set-up software is used.
17	OPTIONAL FUNCTION-6 (DETECTOR PULSE OUTPUT SETTING)[*2][*3]	0: OUTPUT PULSE 1: DIVIDING RATIO								0	–	• Set the type of detector output pulse output from communication connector (CN3).
18	DETECTOR OUTPUT PULSE[*2][*3]	0 to 65535 PLS								4000	PLS	• Set the detector pulse output from servo amplifier.
19	MOTOR GEAR	Refer to the Full Closed Control Reference Manual (IB-67316).										
20	MACHINE GEAR											
21	CLOSED PLG. PULSE[*1][*2]											
22	RUN-AWAY DETECTION FUNCTION-1											
23	RUN-AWAY DETECTION FUNCTION-2											

[*1]: Cannot be set with MR-J2 (S)-B and MR-J2M-B.
[*2]: Cannot be set with MR-H-B (N).
[*3]: Cannot be set with MR-J2-B.

4.2.5 Zero point return data

The data to be set is shown below.

No.	Item	Setting range								Default	Remarks
		mm		inch		degree		PULSE		Initial value	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units		
A	DIRECTION	0: REVERSE (CW) 1: FORWARD (CCW)								0	• Sets the direction for zero point return. • Moves in the designated direction when zero point return is started.
B	METHOD	0: Near-zero point dog method 1: Count method 2: Data set method 1 3: Data set method 2								0	• Sets the zero point return method. • The near-zero point dog method or count method is recommended for a servo amplifier which does not support absolute data, and the data set method is recommended for a servo amplifier which supports absolute data.
C	ADDRESS	–2147483648 to 2147483647	$\times10^{-1}$ μm	–2147483648 to 2147483647	$\times10^{-5}$ inch	0 to 35999999	$\times10^{-5}$ degree	–2147483648 to 2147483647	PLS	0	• Sets the present value of the zero point after zero point return.
D	SPEED	0.01 to 6000000.00	mm/ min	0.01 to 600000.000	inch/ min	0.001 to 2147483.647	degree/ min	1 to 1000000	PLS/ s	1	• Sets the speed for zero point return.
E	CREEP SPEED	0.01 to 6000000.00	mm/ min	0.001 to 600000.000	inch/ min	0.001 to 2147483.647	degree/ min	1 to 1000000	PLS/ s	1	• Sets the creep speed (low speed immediately before stopping after deceleration from zero point return speed) after the near-zero point dog.
F	MOVEMENT AFTER DOG	0 to 214748364.7	μm	0 to 21474.83647	inch	0 to 21474.83647	degree	0 to 2147483647	PLS	0	• Sets the movement amount after the near-zero point dog for the count method. • Set greater than the deceleration distance at the zero point return speed.
G	P.B. NO.	1 to 64								1	• Sets the parameter block to use for zero point return (see Section 4.2.7).

* Zero point return is executed with the following servo program example.

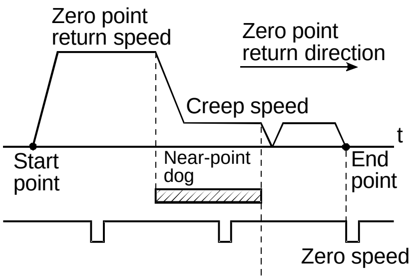
Examples

Servo program No.1 <K 1>

The zero point return axis is axis 1

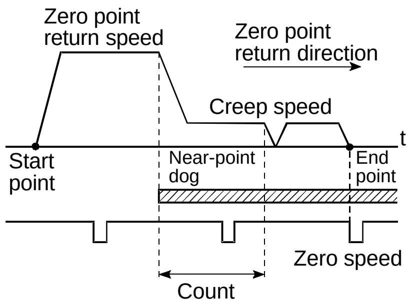
```
ZERO
     Axis 1
```

Near-point dog method



Moves to the next zero point after the near-point dog turns OFF.

Count method



Moves to the next zero point after the count.

Data set method 1

The stopped position is the zero point.

Near-point dog is not required.

Data set method 2

The actual motor position is the zero point.

Near-point dog is not required.
(Suitable for stopper type)

4.2.6 JOG operation data

The data to be set is shown below.

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
A	JOG SPEED LIMIT	0.01 to 6000000.00	mm/min	0.001 to 600000.000	inch/min	0.001 to 2147483.647	degree/min	1 to 1000000	PLS/s	20000	PLS/s	• Sets the max. speed during JOG operation. • The JOG speed limit value becomes the JOG operation speed if the JOG operation speed is set greater than JOG speed limit value.
B	P.B. NO.	1 to 64								1	–	• Sets the parameter block number used for JOG operation. (See section 4.2.7.)

4.2.7 Parameter block

The parameter block is used to determine the acceleration time, deceleration time and torque limit value, etc., used for zero point return operation, JOG operation and positioning with the servo program.
The parameter blocks are No. 1 to No. 16.
The data to be set is shown below.

No.	Item	Setting range								Default		Remarks
		mm		inch		degree		PULSE		Initial value	Units	
		Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units			
1	CONTROL UNIT	0	–	1	–	2	–	3	–	3	–	• Set the units for compensation control. • Can also be used as the units for the command speed and allowable error range for circular interpolation set in the servo program.
2	SPEED RESTRICTION *1	0.01 to 6000000.00	mm/min	0.001 to 600000.000	inch/min	0.001 to 2147483.647	degree/min	1 to 1000000	PLS/s	200000	PLS/s	• Sets the maximum speed for positioning/zero point return. • If the positioning speed or zero point return speed setting exceeds the speed limit value, control is executed at the speed limit value.
3	ACCELERA-TION TIME	1 to 65535ms								1000	ms	• Set the time taken to reach the speed limit value from the start of motion.
4	DECELERA-TION TIME	1 to 65535ms								1000	ms	• Set the time taken to stop from the speed limit value.
5	SHORT STOP TIME	1 to 65535ms								1000	ms	• Set the time taken to stop from the speed limit value when a rapid stop is executed.
6	S RATIO	0 to 100%								0	%	• Set the S curve ration for S pattern processing. • When the S curve ratio is 0%, trapezoidal acceleration/ deceleration processing is executed.
7	TORQUE LIMIT*1	1 to 500%								300	%	• Set the torque limit value in the servo program.
8	STOP METHOD	0: Deceleration stop executed based on the deceleration time 1: Deceleration stop executed based on the rapid stop deceleration time								0	–	• Set the deceleration processing when external signals (STOP, FLS, RLS) are input.
9	CIRCULAR ERROR RANGE	0 to 10000.0	μm	0 to 1.00000	inch	0 to 1.00000	degree	0 to 100000	PLS	100	PLS	• Set the permissible range for the locus of the arc and the set end point coordinates.

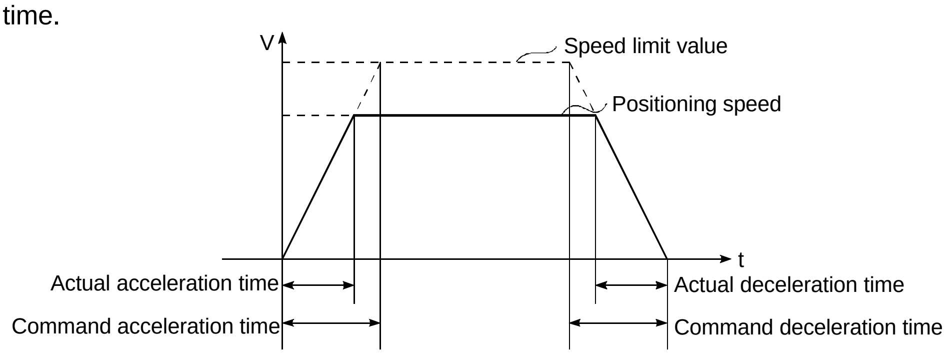
*1 Not used for virtual mode. (Invalid)

POINT

(1) Parameter blocks are designated in the zero point return data, JOG operation data, or servo program.
(2) The various parameter block data can be changed in the servo program

Acceleration time, deceleration time, sudden stop time

As shown below the speed limit value is used as the base. If the positioning command speed is lower than the speed limit value, the actual acceleration time or actual deceleration time will be shorter than the commanded time.



When the positioning command speed is faster than the speed limit value, the axis will move at the speed limit value speed.

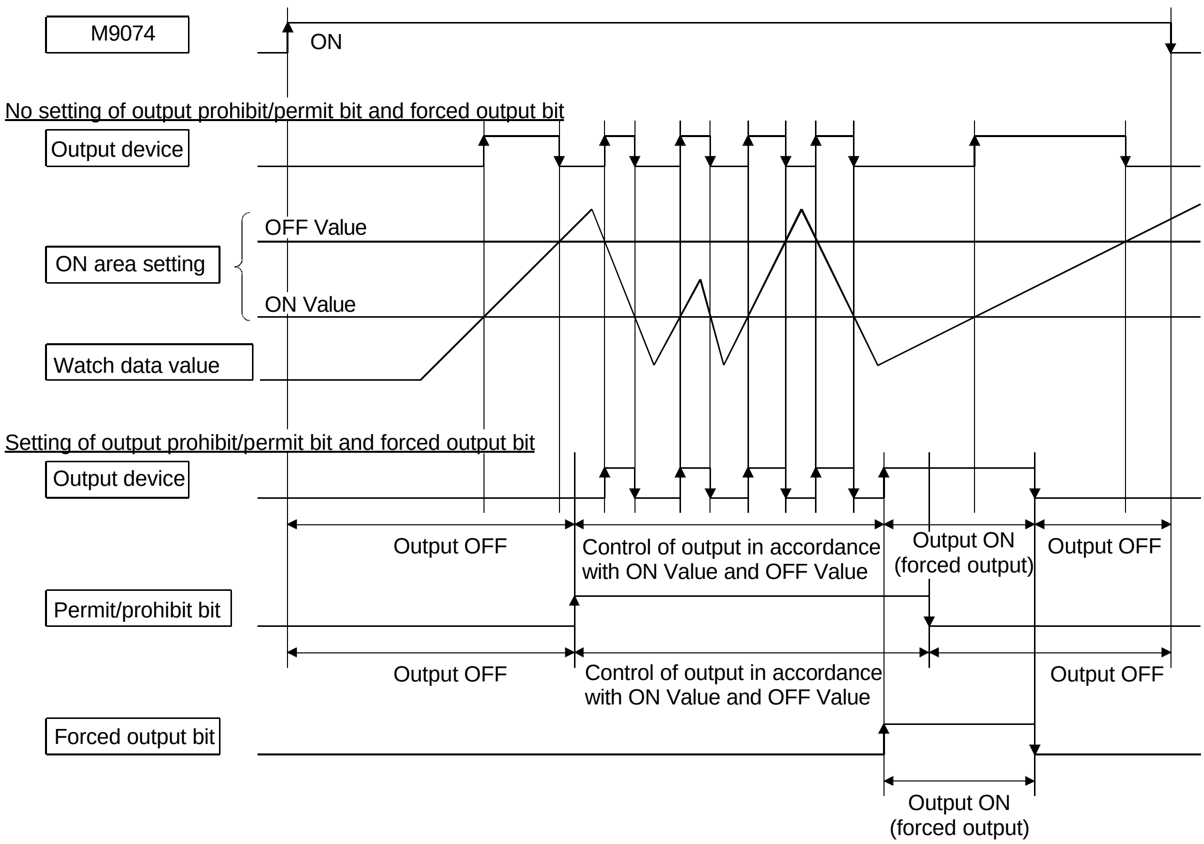
4.2.8 Limit switch output function

This function is used to output the ON/OFF signal corresponding to the range of watch data set for each output device.
It is possible to set the output device for a maximum of 32 points.

(1) The limit switch output function serves to output the ON to the output device while the value of watch data is located within the ON output area set by (ON Value) and (OFF Value).
To execute the limit switch output function, it is necessary to set each data using the peripheral device.

No.	Item		Setting range	Received cycle	Refresh cycle	Remarks
1	Output device		Bit devices (X, Y, M, L, B)	–	Operation cycle	
2	Watch data		Motion control data /Word devices (D, W, #, absolute address) (16-bit integer type/32-bit integer type/64-bit floating decimal point type)	Operation cycle	–	
3	ON area setting	ON Value	Word devices (D, W, #)/Constants (K, H)	Operation cycle	–	
		OFF Value	Word devices (D, W, #)/Constants (K, H)	Operation cycle	–	
4	Output permit/ prohibit bit		Bit devices (X, Y, M, L, B, F, special-M) /None (Default)	Operation cycle	–	ON : Permit OFF : Prohibit None : Permit all the time
5	Forced output bit		Bit devices (X, Y, M, L, B, F, special-M) /None (Default)	Operation cycle	–	None : Not output forcibly all the time (OFF status)

(2) When the multiple watch data, ON area, output permit/prohibit bit and forced output bit are set to the same output device, the logical sum of output result is output in accordance with each setting.



4.3 Positioning control device

The Q motion CPU is provided with positioning control devices for positioning information.
The explanations of the devices in sections 4.3.1 to 4.3.5 are only for the Q172 specifications.

(Device)	(For Q172)	(Details for Q172)
Each axis status	M2400 to M2559	Refer to section 4.3.1.
Each axis command signal	M3200 to M3359	
Virtual servomotor axis status	M4000 to M4159	
Synchronous encoder axis status	M4640 to M4671	
Virtual servomotor axis command signal	M4800 to M4939	
Synchronous encoder axis command signal	M5440 to M5471	
Internal relay	M2000 to M2319	Refer to section 4.3.2.
Data register	D0 to D1315	Refer to section 4.3.3.
Special relay	M9073 to M9079	Refer to section 4.3.4.
Special registers	D752 to D799 D9180 to D9199	Refer to section 4.3.5.

4.3.1 Status/command signals M2400 to M5471 (For Q172)

The Q172CPU(N) has 8192 internal relay and latch relay points M/L0 to M/L8191. Of these points, M2400 to M3359 are used to exchange data for each axis. The signal name and input/output No. for each axis is determined as shown below.

(1) List of M2400 to M3359

(In the virtual mode, the output module is the target instead of the drive module.)

	Device								Signal name	Applicable mode	
	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8		Real	Virtual
Axis status	M2400	M2420	M2440	M2460	M2480	M2500	M2520	M2540	Positioning start completed	○	–
	M2401	M2421	M2441	M2461	M2481	M2501	M2521	M2541	Positioning completed	○	–
	M2402	M2422	M2442	M2462	M2482	M2502	M2522	M2542	In-position	○	○
	M2403	M2423	M2443	M2463	M2483	M2503	M2523	M2543	Command in-position	○	–
	M2404	M2424	M2444	M2464	M2484	M2504	M2524	M2544	Speed control in progress	○	–
	M2405	M2425	M2445	M2465	M2485	M2505	M2525	M2545	Speed/position switching latch	○	–
	M2406	M2426	M2446	M2466	M2486	M2506	M2526	M2546	Zero pass	○	○
	M2407	M2427	M2447	M2467	M2487	M2507	M2527	M2547	Error detection	○	○
	M2408	M2428	M2448	M2468	M2488	M2508	M2528	M2548	Servo error detection	○	○
	M2409	M2429	M2449	M2469	M2489	M2509	M2529	M2549	Zero point return request	○	○
	M2410	M2430	M2450	M2470	M2490	M2510	M2530	M2550	Zero point return completed	○	○
	M2411	M2431	M2451	M2471	M2491	M2511	M2531	M2551	External signal FLS (Upper limit switch)	○	○
	M2412	M2432	M2452	M2472	M2492	M2512	M2532	M2552	External signal RLS (Lower limit switch)	○	○
	M2413	M2433	M2453	M2473	M2493	M2513	M2533	M2553	External signal (STOP)	○	○
	M2414	M2434	M2454	M2474	M2494	M2514	M2534	M2554	External signal DOG/CHANGE	○	○
	M2415	M2435	M2455	M2475	M2495	M2515	M2535	M2555	Servo ON/OFF	○	○
	M2416	M2436	M2456	M2476	M2496	M2516	M2536	M2556	Torque control in progress	○	○
	M2417	M2437	M2457	M2477	M2497	M2517	M2537	M2557	Unusable	–	–
	M2418	M2438	M2458	M2478	M2498	M2518	M2538	M2558	Virtual mode continuous operation disabled warning signal	○	○
	M2419	M2439	M2459	M2479	M2499	M2519	M2539	M2559	M code output in progress	○	–
Axis command signal	M3200	M3220	M3240	M3260	M3280	M3300	M3320	M3340	Stop command	○	–
	M3201	M3221	M3241	M3261	M3281	M3301	M3321	M3341	Sudden stop command	○	–
	M3202	M3222	M3242	M3262	M3282	M3302	M3322	M3342	Forward JOG start	○	–
	M3203	M3223	M3243	M3263	M3283	M3303	M3323	M3343	Reverse JOG start	○	–
	M3204	M3224	M3244	M3264	M3284	M3304	M3324	M3344	End signal OFF command	○	–
	M3205	M3225	M3245	M3265	M3285	M3305	M3325	M3345	Speed/position changeover enabled	○	–
	M3206	M3226	M3246	M3266	M3286	M3306	M3326	M3346	Unusable	–	–
	M3207	M3227	M3247	M3267	M3287	M3307	M3327	M3347	Error reset	○	○
	M3208	M3228	M3248	M3268	M3288	M3308	M3328	M3348	Servo error reset	○	○
	M3209	M3229	M3249	M3269	M3289	M3309	M3329	M3349	External STOP input valid/invalid when starting	○	–
	M3210	M3230	M3250	M3270	M3290	M3310	M3330	M3350	Unusable	–	–
	M3211	M3231	M3251	M3271	M3291	M3311	M3331	M3351		–	–
	M3212	M3232	M3252	M3272	M3292	M3312	M3332	M3352	Current value update request command	○	–
	M3213	M3233	M3253	M3273	M3293	M3313	M3333	M3353	Address clutch reference setting	–	○
	M3214	M3234	M3254	M3274	M3294	M3314	M3334	M3354	Cam reference position setting	–	○
	M3215	M3235	M3255	M3275	M3295	M3315	M3335	M3355	Servo OFF	○	○
	M3216	M3236	M3256	M3276	M3296	M3316	M3336	M3356	Unusable	–	–
	M3217	M3237	M3257	M3277	M3297	M3317	M3337	M3357		–	–
	M3218	M3238	M3258	M3278	M3298	M3318	M3338	M3358		–	–
	M3219	M3239	M3259	M3279	M3299	M3319	M3339	M3359	FIN signal	○	–

Note) If servo OFF is turned ON to release the servo lock for each axis, the servo ON/OFF state will turn OFF and a confirmation can be made.

(2) List of M4000 to M5471

(In the virtual mode, the output module is the target instead of the drive module.)

	Device															
	Synchronous encoder								Virtual servomotor							
	P1/ E1	P2/ E2	P3/ E3	P4/ E4	P5/ E5	P6/ E6	P7/ E7	P8/ E8	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Axis status	–	–	–	–	–	–	–	–	M4000	M4020	M4040	M4060	M4080	M4100	M4120	M4140
									M4001	M4021	M4041	M4061	M4081	M4101	M4121	M4141
									M4002	M4022	M4042	M4062	M4082	M4102	M4122	M4142
									M4003	M4023	M4043	M4063	M4083	M4103	M4123	M4143
									M4004	M4024	M4044	M4064	M4084	M4104	M4124	M4144
									M4005	M4025	M4045	M4065	M4085	M4105	M4125	M4145
									M4006	M4026	M4046	M4066	M4086	M4106	M4126	M4146
									M4007	M4027	M4047	M4067	M4087	M4107	M4127	M4147
									M4008 to M4018	M4028 to M4038	M4048 to M4058	M4068 to M4078	M4088 to M4098	M4108 to M4118	M4128 to M4138	M4148 to M4158
									M4019	M4039	M4059	M4079	M4099	M4119	M4139	M4159
	M4640	M4644	M4648	M4652	M4656	M4660	M4664	M4668	–	–	–	–	–	–	–	–
	M4641	M4645	M4649	M4653	M4657	M4661	M4665	M4669	–	–	–	–	–	–	–	–
	M4642	M4646	M4650	M4654	M4658	M4662	M4666	M4670	–	–	–	–	–	–	–	–
	M4643	M4647	M4651	M4655	M4659	M4663	M4667	M4671	–	–	–	–	–	–	–	–
Axis command signal	–	–	–	–	–	–	–	–	M4800	M4820	M4840	M4860	M4880	M4900	M4920	M4940
									M4801	M4821	M4841	M4861	M4881	M4901	M4921	M4941
									M4802	M4822	M4842	M4862	M4882	M4902	M4922	M4942
									M4803	M4823	M4843	M4863	M4883	M4903	M4923	M4943
									M4804	M4824	M4844	M4864	M4884	M4904	M4924	M4944
									M4805	M4825	M4845	M4865	M4885	M4905	M4925	M4945
									M4806	M4826	M4846	M4866	M4886	M4906	M4926	M4946
									M4807	M4827	M4847	M4867	M4887	M4907	M4927	M4947
									M4808	M4828	M4848	M4868	M4888	M4908	M4928	M4948
									M4809	M4829	M4849	M4869	M4889	M4909	M4929	M4949
									M4810 to M4818	M4830 to M4838	M4850 to M4858	M4870 to M4878	M4890 to M4898	M4910 to M4918	M4930 to M4938	M4950 to M4958
									M4819	M4839	M4859	M4879	M4899	M4919	M4939	M4959
	M5440	M5444	M5448	M5452	M5456	M5460	M5464	M5468	–	–	–	–	–	–	–	–
	M5441	M5445	M5449	M5453	M5457	M5461	M5465	M5469	–	–	–	–	–	–	–	–
	M5442	M5446	M5450	M5454	M5458	M5462	M5466	M5470	–	–	–	–	–	–	–	–
	M5443	M5447	M5451	M5455	M5459	M5463	M5467	M5471	–	–	–	–	–	–	–	–

	Signal name	Applicable mode	
		Real	Virtual
	Positioning start completed	–	○
	Positioning completed		
	Unusable	–	–
	Command in-position	–	○
	Speed control in progress	–	○
	Unusable	–	–
	Error detection	–	○
	Unusable	–	–
	M code output in progress	–	○
	Error detection	○	○
	External signal TREN detection		
	Virtual mode continuous operation disabled warning signal		
	Unusable	–	–
	Stop command	–	○
	Sudden stop command		
	Forward JOG start		
	Reverse JOG start		
	End signal OFF command		
	Unusable	–	–
	Error reset	–	○
	Unusable		
	External STOP input valid/invalid when starting		
	Unusable	–	–
	FIN signal	–	○
	Error reset	–	○
	Unusable	–	–

P1 to P8 : Increment synchronization encoder connected to the Q173PX manual pulse generator input I/F.

E1 to E8 : Absolute synchronous encoder connected to the Q172EX serial synchronization encoder.

External signal TERN : Detection signal that turns ON when signal is input in Q173PX tracking I/F.

4.3.2 Internal relays M2000 to M2319 (For Q172)

The Q172CPU(N) has 8192 internal relay and latch relay points M/L0 to M/L8191.
Of these points, M2000 to M2319 are used for positioning control with applications determined as shown below.

Device No.	Signal name	Applicable mode	
		Real	Virtual
M2000	PLC READY	○	○
M2001	Axis 1 start accept flag	○	○
M2002	Axis 2 start accept flag		
M2003	Axis 3 start accept flag		
M2004	Axis 4 start accept flag		
M2005	Axis 5 start accept flag		
M2006	Axis 6 start accept flag		
M2007	Axis 7 start accept flag		
M2008	Axis 8 start accept flag		
M2009 to M2033	Unusable	–	–
M2034	PC link communication error flag	○	○
M2035 to M2038	Unusable	–	–
M2039	SFC error detection flag	○	○
M2040	Speed changeover point designation flag	○	○
M2041	System setting error flag	○	○
M2042	All-axes servo ON command	○	○
M2043	Real/virtual mode changeover request	–	○
M2044	Real/virtual mode changeover status	–	○
M2045	Real/virtual mode changeover error	–	○
M2046	Synchronisation deviation error	–	○
M2047	Motion slot module error	○	○
M2048	JOG simultaneous start command	○	○
M2049	All-axes servo ON accept flat	○	○
M2050	Start buffer full	○	○
M2051	Manual pulse generator 1 enable	○	○
M2052	Manual pulse generator 2 enable	○	○
M2053	Manual pulse generator 3 enable	○	○
M2054	Operation cycle over flag	○	○
M2055 to M2060	Unusable	–	–
M2061	Axis 1 speed change flag	○	○
M2062	Axis 2 speed change flag		
M2063	Axis 3 speed change flag		
M2064	Axis 4 speed change flag		
M2065	Axis 5 speed change flag		
M2066	Axis 6 speed change flag		
M2067	Axis 7 speed change flag		
M2068	Axis 8 speed change flag		
M2069 to M2100	Unusable	–	–

Device No.	Signal name		Applicable mode	
			Real	Virtual
M2101	1-axis synchronous encoder current value change flag		–	○
M2102	2-axis synchronous encoder current value change flag		–	○
M2103	3-axis synchronous encoder current value change flag		–	○
M2104	4-axis synchronous encoder current value change flag		–	○
M2105	5-axis synchronous encoder current value change flag		–	○
M2106	6-axis synchronous encoder current value change flag		–	○
M2107	7-axis synchronous encoder current value change flag		–	○
M2108	8-axis synchronous encoder current value change flag		–	○
M2109 to M2127	Unusable		–	–
M2128	1-axis automatic deceleration flag		○	○
M2129	2-axis automatic deceleration flag		○	○
M2130	3-axis automatic deceleration flag		○	○
M2131	4-axis automatic deceleration flag		○	○
M2132	5-axis automatic deceleration flag		○	○
M2133	6-axis automatic deceleration flag		○	○
M2134	7-axis automatic deceleration flag		○	○
M2135	8-axis automatic deceleration flag		○	○
M2136 to M2159	Unusable		–	–
M2160	Output module axis 1	Main shaft side clutch status	–	○
M2161		Auxiliary input side clutch status	–	○
M2162	Output module axis 2	Main shaft side clutch status	–	○
M2163		Auxiliary input side clutch status	–	○
M2164	Output module axis 3	Main shaft side clutch status	–	○
M2165		Auxiliary input side clutch status	–	○
M2166	Output module axis 4	Main shaft side clutch status	–	○
M2167		Auxiliary input side clutch status	–	○
M2168	Output module axis 5	Main shaft side clutch status	–	○
M2169		Auxiliary input side clutch status	–	○
M2170	Output module axis 6	Main shaft side clutch status	–	○
M2171		Auxiliary input side clutch status	–	○
M2172	Output module axis 7	Main shaft side clutch status	–	○
M2173		Auxiliary input side clutch status	–	○
M2174	Output module axis 8	Main shaft side clutch status	–	○
M2175		Auxiliary input side clutch status	–	○
M2176 to M2239	Unusable		–	–
M2240	1-axis speed change [0] accept flag		○	○
M2241	2-axis speed change [0] accept flag		○	○
M2242	3-axis speed change [0] accept flag		○	○
M2243	4-axis speed change [0] accept flag		○	○
M2244	5-axis speed change [0] accept flag		○	○
M2245	6-axis speed change [0] accept flag		○	○
M2246	7-axis speed change [0] accept flag		○	○
M2247	8-axis speed change [0] accept flag		○	○
M2248 to M2319	Unusable		–	–

4.3.3 Data registers D0 to D1315 (For Q172)

The Q172CPU(N) has 8192 data register points D0 to D8191.
Of these points, the 1316 points D0 to D1315 are used for positioning control with applications determined as shown below.

Device																
Synchronous encoder								Virtual servomotor								
P1/E1	P2/E2	P3/E3	P4/E4	P5/E5	P6/E6	P7/E7	P8/E8	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8	
–	–	–	–	–	–	–	–	D0 D1	D20 D21	D40 D41	D60 D61	D80 D81	D100 D101	D120 D121	D140 D141	
								D2 D3	D22 D23	D42 D43	D62 D63	D82 D883	D102 D103	D122 D123	D142 D143	
								D4 D5	D24 D25	D44 D45	D64 D65	D84 D85	D104 D105	D124 D125	D144 D145	
								D6	D26	D46	D66	D86	D106	D126	D146	
								D7	D27	D47	D67	D87	D107	D127	D147	
								D8	D28	D48	D68	D88	D108	D128	D148	
								D9	D29	D49	D69	D89	D109	D129	D149	
								D10	D30	D50	D70	D90	D110	D130	D150	
								D11	D31	D51	D71	D91	D111	D131	D151	
								D12	D32	D52	D72	D92	D112	D132	D152	
								D13	D33	D53	D73	D93	D113	D133	D153	
								D14	D34	D54	D74	D94	D114	D134	D154	
								D15	D35	D55	D75	D95	D115	D135	D155	
								D16	D36	D56	D76	D96	D116	D136	D156	
								D17	D37	D57	D77	D97	D117	D137	D157	
								D18	D38	D58	D78	D98	D118	D138	D158	
								D19	D39	D59	D79	D99	D119	D139	D159	
								D640 D641	D642 D643	D644 D645	D646 D647	D648 D649	D650 D651	D652 D653	D654 D655	
								D704								
								D705								
								D706								
								D707								
								D708								
								D709								
								D710 to D713								
								D714, D715								
								D716, D717								
								D718, D719								
								D720	D721	D722	D723	D724	D725	D726	D727	

(Continued on page 4-22)

	Signal name	Applicable mode	
		Real	Virtual
	Current feed value/Roller peripheral speed	○	○
	Actual current value		
	Deviation counter		
	Minor error code		
	Major error code		
	Servo error code		
	Movement amount for repeat zero point return	○	–
	Movement amount after proximity dog ON		
	Execution program number	○	–
	M code		
	Torque limit value	○	○
	Data set pointer for uniform speed control	○	○
	Movement amount change register	○	–
	Actual current value when STOP is input	○	–
	JOG speed setting	○	○
	PLC READY flag request	○	○
	Speed changeover point designation flag request	○	○
	All-axis servo ON command request	○	○
	Real/virtual mode changeover request (SV22)	○	○
	Simultaneous JOG start command request	○	○
	Unusable	–	–
	Simultaneous JOG start axis setting	○	○
	Register to set axis No. to be controlled by manual pulse generator 1	○	○
	Register to set axis No. to be controlled by manual pulse generator 2	○	○
	Register to set axis No. to be controlled by manual pulse generator 3	○	○
	1 pulse input magnification setting for manual pulse generator	○	○

P1 to P8 : Increment synchronisation encoder connected to the Q173PX manual pulse generator input I/F.

E1 to E8 : Absolute synchronous encoder connected to the Q172EX serial synchronisation encoder.

(Continued from page 4-20)

Device																
Synchronous encoder								**Virtual servomotor**								
P1/E1	**P2/E2**	**P3/E3**	**P4/E4**	**P5/E5**	**P6/E6**	**P7/E7**	**P8/E8**	**Axis 1**	**Axis 2**	**Axis 3**	**Axis 4**	**Axis 5**	**Axis 6**	**Axis 7**	**Axis 8**	
								D800 D801	D810 D811	D820 D821	D830 D831	D840 D841	D850 D851	D860 D861	D870 D871	
D1120 D1121	D1130 D1131	D1140 D1141	D1150 D1151	D1160 D1161	D1170 D1171	D1180 D1181	D1190 D1191	–	–	–	–	–	–	–	–	
D1122	D1132	D1142	D1152	D1162	D1172	D1182	D1192	D802	D812	D822	D832	D842	D852	D862	D872	
D1123	D1133	D1143	D1153	D1163	D1173	D1183	D1193	D803	D813	D823	D833	D843	D853	D863	D873	
								D804	D814	D824	D834	D844	D854	D864	D874	
								D805	D815	D825	D835	D845	D855	D865	D875	
D1126 D1127	D1136 D1137	D1146 D1147	D1156 D1157	D1166 D1167	D1176 D1177	D1186 D1187	D1196 D1197	D806 D807	D816 D817	D826 D827	D836 D837	D846 D847	D856 D857	D866 D867	D876 D877	
								D808	D818	D828	D838	D848	D858	D868	D878	
								D809	D819	D829	D839	D849	D859	D869	D879	
								D1241	D1251	D1261	D1271	D1281	D1291	D1301	D1311	
–	–	–	–	–	–	–	–	D1242 D1243	D1252 D1253	D1262 D1263	D1272 D1273	D1282 D1283	D1292 D1293	D1302 D1303	D1312 D1313	
								D1244 D1245	D1254 D1255	D1264 D1265	D1274 D1275	D1284 D1285	D1294 D1295	D1304 D1305	D1314 D1315	

	Signal name	Applicable mode	
		Real	Virtual
	Current feed value	–	○
	Current value	–	○
	Minor error code	–	○
	Major error code	–	○
	Unusable	–	–
	Unusable	–	–
	Current value after main shaft differential gears	–	○
	Error search output axis check No.	–	○
	Unusable	–	–
	Execution cam No.	–	○
	Execution stroke amount		
	Current value within one cam shaft rotation		

P1 to P8 : Increment synchronization encoder connected to the Q173PX manual pulse generator input I/F.

E1 to E8 : Absolute synchronous encoder connected to the Q172EX serial synchronisation encoder.

4.3.4 Special relays M9073 to M9079, M9104, M9105

The Q172CPU(N) has 256 special relay points M9000 to M9255.
Of these points, the nine points M9073 to M9079, M9104 and M9105 are used for positioning control with applications determined as shown below.

Device No.	Signal name	Applicable mode	
		Real	Virtual
M9073	PCPU WDT error flag	○	○
M9074	PCPU READY completion flag		
M9075	Test mode flag		
M9076	Emergency stop input flag		
M9077	Manual pulse generator axis setting error flag		
M9078	Test mode request error flag		
M9079	Servo program setting error flag		
M9104	Servo parameter read request flag		
M9105	Servo parameter read flag		

4.3.5 Special registers D752 to D799, D9017, D9019, D9104, D9180 to D9199 (For Q172)

The Q172CPU(N) has 256 special register points D9000 to D9255.
Of these points, the 71 points D752 to D799, D9017, D9019, D9104 and D9180 to D9199 are used for positioning control with applications determined as shown below.

Device No.	Signal name	Applicable mode	
		Real	Virtual
D752	Manual pulse generator (P1) smoothing magnification setting area	○	○
D753	Manual pulse generator (P2) smoothing magnification setting area		
D754	Manual pulse generator (P3) smoothing magnification setting area		
D755	Manual pulse generator 1 permit flag set request		
D756	Manual pulse generator 2 permit flag set request		
D757	Manual pulse generator 3 permit flag set request		
D758	Unusable		
D759	PCPU READY flag status (0: OFF/1: ON)		
D760 to D791	Unusable	–	–
D792 to D799	Servo amplifier type	○	○
D9017	Current main cycle	○	○
D9019	Maximum main cycle	○	○
D9104	Servo parameter read request axis No.	○	○
D9180 to D9181	Unusable	–	–
D9182 to D9183	Test mode request error	○	○
D9184	Motion CPU WDT error factor	○	○
D9185 to D9187	Manual pulse generator axis setting error	○	○
D9188	Motion operation cycle	○	○
D9189	Error program number	○	○
D9190	Servo program setting error output	○	○
D9191 to D9192	Servo amplifier mounting information	○	○
D9193 to D9195	Unusable	–	–
D9196	PC link communication error	○	○
D9197	Motion setting operation cycle	○	○
D9198 to D9199	Unusable	–	–

POINT

Handling of registers (D704 to D708, D755 to D757)

Since the bit devices shown below cannot be turned ON/OFF for each bit from the Q-PLC CPU, each bit device is turned ON when the least significant bit is changed from "0" to "1", and is turned off when it is changed from "1" to "0".
The DDRD and DDWR commands are used to send the request from the Q-PLC CPU.
It is possible to turn ON/OFF the bit devices directly with the motion SFC program

No.	Functions	Bit devices	Request registers
1	PLC READY flag	M2000	D704
2	Speed changeover point designation flag	M2040	D705
3	All-axis servo ON command	M2042	D706
4	Real/virtual mode changeover request (SV22 only)	M2043	D707
5	Simultaneous JOG start command	M2048	D708
6	Manual pulse generator 1 enable flag	M2051	D755
7	Manual pulse generator 2 enable flag	M2052	D756
8	Manual pulse generator 3 enable flag	M2053	D757

4.4 Motion SFC dedicated devices

The motion CPU (PCPU) dedicated devices include the motion registers (#0 to #8191) and coast timer (FT).
These devices can be used in the operation control (F/FS) program or transition (G) program.
It is not possible to directly access these devices from the PLC. When the devices are used at the PLC side, therefore, substitute them for PLC devices before accessing.

4.4.1 Motion registers (#0 to #8191)

Motion devices	Item	Q173CPU(N)/Q172CPU(N)
Motion register (#)	Number of points	8192 points (#0 to #8191)
	Data size	16 bits/point
	Latch	Latched except for SFC dedicated devices. (Cleared at all points due to latch clear operation.)
	Applicable task	Normal , event , NMI
	Access	READ/WRITE is executable at all ranges.

(1) List of motion registers
These motion registers are common to all OSs.

Device No.	Division of application	Remarks
#0 to	User devices (8,000 points)	Cleared by latch clear.
#8000 to	SFC dedicated devices (66 points)	Cleared only when the power supply is turned ON and when the operation is reset using the key.
#8066 to #8191	Servo monitor devices (126 points)	Not cleared.

POINT
Note that the motion register (#) cannot be set to be used as an indirect designation device for the mechanism program.

(2) **SFC dedicated devices (#8000 to #8191)**

The SFC dedicated devices are shown below.

It indicates the refresh cycle for device of which signal direction is "Status", and the retrieval cycle for device of which signal direction is "Command".

Device No.	Signal name		Signal direction		Refresh cycle	Received cycle
			Status	Command		
#8000	7th oldest error information (oldest error information)	SFC error history (8 times) (64 points)	○	–	When an error occurs	–
#8008	6th oldest error information					
#8016	5th oldest error information					
#8024	4th oldest error information					
#8032	3rd oldest error information					
#8040	2nd oldest error information					
#8048	Last error information					
#8056	Newest error information					
#8064	Axis 1 amplifier type		○	–	When amplifier power supply is turned ON	–
#8065	Axis 1 motor current				3.55ms	
#8066 #8067	Axis 1 motor speed					
#8068	Axis 2 amplifier type				When amplifier power supply is turned ON	
#8069	Axis 2 motor current				3.55ms	
#8070 #8071	Axis 2 motor speed					
#8072	Axis 3 amplifier type				When amplifier power supply is turned ON	
#8073	Axis 3 motor current				3.55ms	
#8074 #8075	Axis 3 motor speed					
#8076	Axis 4 amplifier type				When amplifier power supply is turned ON	
#8077	Axis 4 motor current				3.55ms	
#8078 #8079	Axis 4 motor speed					
#8080	Axis 5 amplifier type				When amplifier power supply is turned ON	
#8081	Axis 5 motor current				3.55ms	
#8082 #8083	Axis 5 motor speed					
#8084	Axis 6 amplifier type				When amplifier power supply is turned ON	
#8085	Axis 6 motor current				3.55ms	
#8086 #8087	Axis 6 motor speed					
#8088	Axis 7 amplifier type				When amplifier power supply is turned ON	
#8089	Axis 7 motor current				3.55ms	
#8090 #8091	Axis 7 motor speed					
#8092	Axis 8 amplifier type				When amplifier power supply is turned ON	
#8093	Axis 8 motor current				3.55ms	
#8094 #8095	Axis 8 motor speed					

(3) SFC error history device

The error information for up to eight past errors after turning ON the CPU power supply is stored as a history. The error information of #8056 to #8063 contains newest error.

The error during SFC control, and conventional errors such as minor error, major error, servo error, servo program error, mode select error, etc., are all integrated in the history.

When an error occurs, the "SFC error detection signal M2039" is set at the same time.

The error information is as shown below.

<table>
<tr><th rowspan="2">No.</th><th rowspan="2" colspan="2">Signal name</th><th colspan="2">Contents</th></tr>
<tr><th>Error in SFC control</th><th>Conventional error</th></tr>
<tr><td>+0</td><td colspan="2">Erroneous SFC program No.</td><td>0 to 255 : Erroneous SFC program No.
–1 : Not related to SFC program</td><td>–1</td></tr>
<tr><td>1</td><td colspan="2">Error type</td><td>1 : In case of F/FS
2 : In case of G
–1 : In case of SFC drawing
–2 : In case of K or others (in case of other than F/FS, G, SFC drawing)</td><td>3 : Minor/major error (output module in real mode/virtual mode (SV22 only))
4 : Minor/major error (virtual servomotor shaft) (SV22 only))
5 : Minor/major error (synchronous encoder shaft) (SV22 only))
6 : Servomotor
7 : Servo program error
8 : Mode select error (SV22 only)
9 : Manual pulse generator shaft setting error
10 : Test mode request error
11 : PCPWDT error
12 : Personal computer link communication error</td></tr>
<tr><td>2</td><td colspan="2">Error program No.</td><td>0 to 4095 : F/FS, G, K program No.
0 to 255 : GSUB program No.
–1 : Not concerned with F/FS, G, K, GSUB</td><td>0 to 4095 : Servo program No. when error type is "3 (In real mode)", "4" or "7"
–1 : Other cases (including zero point return in un-start, JOG, manual pulse generator, test mode servo start and start at servo diagnosis)</td></tr>
<tr><td>3</td><td colspan="2">Error block No.
SFC list line No./
axis No.</td><td>0 to 8191 : F/FS or block No. (line No.) of program when the error type is "1" or "2"
0 to 8188 : SFC list line No. when the error type is "–2"
–1 : When the error type is "–1", or when the error type is "1" or "2" and is not concerned with block</td><td>1 to 32 : Axis No. concerned when the error type is "3" to "6"
–1 : Other cases</td></tr>
<tr><td>4</td><td colspan="2">Error code</td><td>16000 and following</td><td>• Conventional error code (below 16000) when the error type is "3" to "6"
• Error code stored in D9190 when the error type is "7"
• Error code stored in D9193 (A273UH-S3), D9195 (A172SH) when the error type is "8"
• -1 when the error type is "9" or "10"
• Error code stored in D9184 when the error type is "11"
• Error code stored in D9196 when the error type is "12"</td></tr>
<tr><td>5</td><td rowspan="3">Error occur-rence time</td><td>Year/ month</td><td colspan="2" rowspan="3">Set the time data (D9025, D9025, D9027) of PLC.
(BCD code, Year: Lower 2 digits of Christian era)</td></tr>
<tr><td>6</td><td>Day/ hour</td></tr>
<tr><td>7</td><td>Minute/ second</td></tr>
</table>

(4) **SFC error detection signal (M2039)**

(Refresh cycle: operation cycle)

The SFC error detection signal (M2039) is turned ON when any errors detected by the motion CPU are generated.

When an error occurs, it is set to the error device in accordance with the following procedure.

(a) The error code is set to each axis or error device.

(b) The error detection signal of each axis or error device is turned ON.

(c) The error is set to the "SFC error history device (#8000 to #8063)" shown above.

(d) The SFC error detection signal (M2039) is turned ON.

For user program, read the error history when the "SFC error detection signal (M2039)" is turned on to reset the "SFC error detection signal (M2039)".

The "SFC error detection signal (M2039)" is turned ON when an error occurs newly thereafter.

POINT

- Even if the "SFC (Error detection signal (M2039)" is reset, the "SFC error history device (#8000 to #8063)" will not be reset (0 clear).
  The error history is controlled continuously all the time after turning ON the power supply.
- Set the clock data and clock data read request (M9028) using the user program.

4.4.2 Coast timer (FT)

Motion device	Item	Q172CPU(N)/Q173CPU(N)
Coast timer (FT)	Number of points	1 point (FT)
	Data size	32 bits/point (–2147483648 to 2147483647)
	Latch	No latch Cleared to "0" when the power supply is turned ON or when the reset key is pressed.
	Applicable task	Normal , event , NMI
	Access	Only reading is executable.
	Timer specifications	888µs timer (The current value (FT) is incremented by 1 every 888µs.)

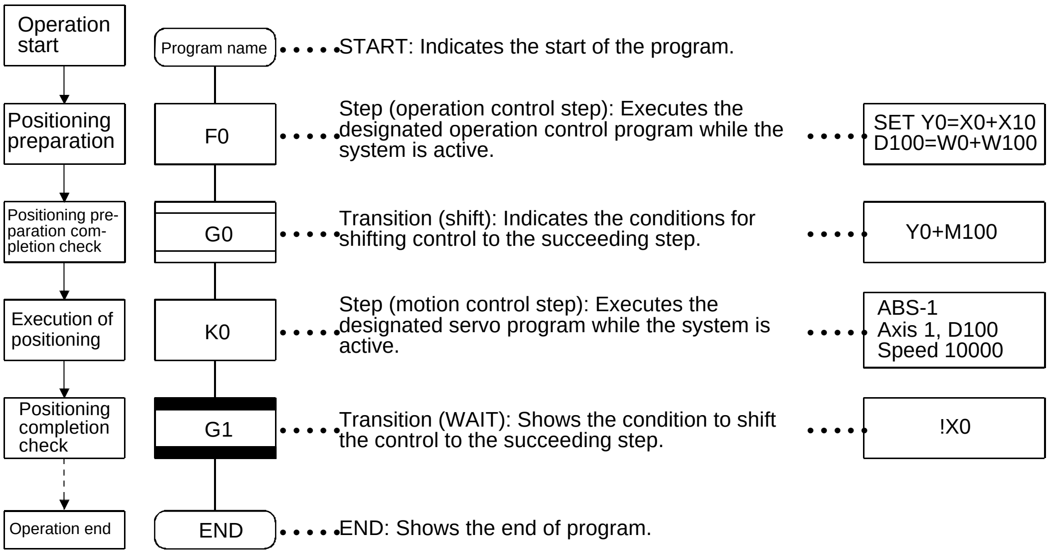
Memo

Chapter 5 SFC Program

This chapter describes the configuration and each element of the SFC program.

5.1 SFC program configuration

The SFC program consists of a combination of START, step, transition, and END, etc., as shown below.



The running SFC program operates as follows.

(1) The step (F0) is activated, and the designated operation is executed at step (F0) (positioning preparation). The step in active state is called "Active step".

(2) Whether the conditions designated by transition (G0) are established (or whether it is ready to start the positioning program) is checked.
When the conditions are established, the active step (F0) is inactivated, and the succeeding step (K0) is activated (start of servo program K0).

(3) At transition (G1), whether the operation at step (K0) is completed (completion of servo program K0 positioning) is checked. When the positioning is completed (conditions established), the control is shifted to the succeeding step.

(4) The control is executed by shifting the active step as shown in the above items (1) to (3), and ended when END is input.

POINT
The number of steps that can be an active step simultaneously is 256 or less (total in all SFC programs).

5.2 List of SFC symbols

The parts that can be a constituent element of SFC program are shown below. The SFC program expresses the operation sequence and shift control by connecting such parts with oriented line.

Division	Designation	Symbol (code size: byte)	List expression	Function
Program start/end	START	Program name (0)	Program name	• Indicates the start of program with the program name. • Designates the program name when the subroutine is called. • START is limited to one per program.
	END	END (8)	END	• Indicates the end of program. • Returns to the calling source program when a subroutine is called. • The multiple ENDS can be set in one program, but it is not absolutely necessary.
Step	Motion control step	Kn (8)	CALL Kn	• Starts the servo program Kn (K0 to K4095).
	1-time execution type operation control step	Fn (8)	CALL Fn	• Executes the operation control program Fn (F0 to F4095) once.
	Scan execution type operation control step	FSn (8)	CALL FSn	• Executes the operation control program FSn (FS0 to FS4095) repeatedly until the succeeding shift conditions are established.
	Subroutine call/start step	Program name (8)	GSUB program name	• When WAIT follows GSUB, the "subroutine" is called and the control is shifted to the designated program. When the END is executed, the control is returned to the calling source program. • When a command other than WAIT follows GSUB, the "subroutine" is started. The designated program is started causing the control to be shifted to the succeeding (lower) program. The start source program and start destination program are executed at the same time, and the start destination program is stopped when the END is executed.
	Clear step	CLR Program name (8)	CLR program name	• Interrupts and ends the designated program that is running. When a stopped program is restarted, it restarts from the initial point (start step). • When the designated program is in "Subroutine call", the execution of the subroutine program is interrupted. • When the designated program is located after "Subroutine start", the execution of the subroutine program is not interrupted. • When the "subroutine called" is cleared, the execution of the designated subroutine is interrupted, and the calling source program is resumed to shift the operation to the succeeding step.

POINT

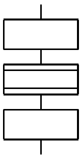
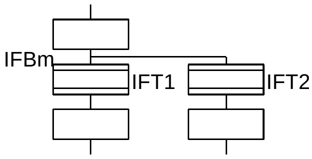
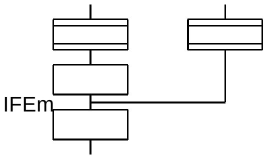
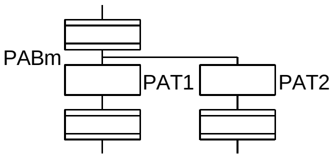
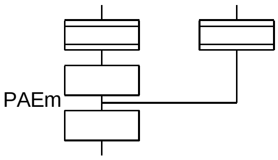
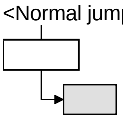
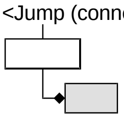
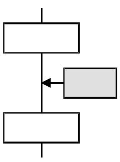
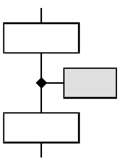
It is possible to set the comment to each symbol such as step, transition, etc. within SFC diagram.

- Program start/end : It is unable to set the comment.
- Step/transition : Display of 80 single-byte (40 double-byte) characters, 20 characters × 4 lines
- Jump/pointer : Display of 64 single-byte (32 double-byte) characters, 16 characters × 4 lines

Division	Designation	Symbol (code size: byte)	List expression	Function
Transition	SHIFT (shifting to advance reading)	Gn (8)	SFT Gn	• If the last step is a motion control step, the operation is shifted to the succeeding step when the shifting conditions Gn (G0 to G4095) are established without waiting for the motion to end. • If the last step is an operation control step, the operation is shifted to the succeeding step when the shifting conditions are established after execution of operation. • If the last step is a subroutine call/start step, the operation is shifted to the succeeding step when the shifting conditions are established without waiting for the subroutine to end.
	WAIT	Gn (8)	WAIT Gn	• If the last step is a motion control step, the operation is shifted to the succeeding step when the shifting conditions Gn (G0 to G4095) are established without waiting for the motion to end. • If the last step is an operation control step, the operation is shifted to the succeeding step when the shifting conditions are established after execution of operation. (Same operation as SHIFT) • If the last step is a subroutine call/start step, the operation is shifted to the succeeding step when the shifting conditions are established without waiting for the subroutine to end.
	WAITON	ON bit device Kn (14)	WAITON bit device	• Prepares to start the succeeding motion control step, and outputs the control command as soon as the designated bit device is turned ON. • Be sure to set it together with motion control step in pairs (1:1).
	WAITOFF	OFF bit device Kn (14)	WAITOFF bit device	• Prepares to start the succeeding motion control step, and outputs the control command as soon as the designated bit device is turned OFF. • Be sure to set it together with motion control step in pairs (1:1).
	SHIFT Y/N	(When the conditions fail to be established) Gn N Y (When the conditions are established)	IFBm IFT1 SFT Gn : JMP IFEm IFT2 SFT Gn+? : JMP IFEm IFEm	• If the last step is a motion control step, the operation is shifted to the succeeding step when the shifting conditions Gn (G0 to G4095) are established without waiting for the motion to end. If the conditions are not established, shifts to the step connected to the right. • If the last step is an operation control step, the operation is shifted to the lower step when the shifting conditions are established after execution of operation. If the conditions are not established, shifts to the step connected to the right. • If the last step is a subroutine call/start step, the operation is shifted to the lower step when the shifting conditions are established before the operation of subroutine is completed. If the conditions are not established, shifts to the step connected to the right.
	WAIT Y/N	(When the conditions fail to be established) Gn N Y (When the conditions are established)	IFBm IFT1 WAIT Gn : JMP IFEm IFT2 WAIT Gn+? : JMP IFEm IFEm	• If the last step is a motion control step, the operation is shifted to the lower step when the shifting conditions Gn (G0 to G4095) are established after the operation of motion is completed. If the conditions are not established, shifts to the step connected to the right. • If the last step is an operation control step, the operation is shifted to the lower step when the shifting conditions are established after execution of operation. If the conditions are not established, shifts to the step connected to the right. (Same operation as SHIFT) • If the last step is a subroutine call/start step, the operation is shifted to the lower step when the shifting conditions are established after the operation of subroutine is completed. If the conditions are not established, shifts to the step connected to the right.
Jump	JUMP	Pn (14)	JMP Pn	• Jumps to the designated pointer Pn (P0 to P16383) within the local program.
Pointer	POINTER	Pn (8)	Pn	• Indicates the jump destination pointer (label). • The step, transition, branch point and connection point can be set. • P0 to P16383 can be set in one program. The program No. may be overlapped with another program No.

5.3 List of branch/connection diagrams

The following shows the branch/connection patterns used to designate the steps/transitions within SFC diagram.

	Designation (code size: byte)	SFC symbol		List expression	Function
Basic type	Series shifting (Each symbol size)			Refer to the list corresponding to symbol shown in SFC diagram of section 5.2.	• Sequentially executes steps and transitions connected in a series from the top. • The steps and transitions may be arranged alternately. • When the transition is omitted, the shifting processing is executed unconditionally.
	Selective branch (Number of branches + 2 × 10)			CALL Kn IFBm IFT1 SFT Gn CALL Fn : JMP IFEm IFT2 SFT Gn' CALL Fn' : (JMP IFEm) IFEm CALL Fn"	• Executes the route for which shifting conditions are established first after execution of step or transition just before the branch. • The head of branch destination for selective branch is limited to transitions, and must be all SHIFT or WAIT. (When the SHIFT and WAIT co-exist, it becomes the parallel branch.)
	Selective connection (8)				• Shifts to the connection point after execution of processing for route branched by selective branch. • Either step or transition may be used just before and after connection.
	Parallel branch (Number of branches × 22 + Number of connections × 2 + 12)			SFT Gn PABm PAT1 CALL Fn SFT Gn' : JMP PAEm PAT2 CALL Fn' SFT Gn" : (JMP PAEm) PAEm CALL Fn" :	• Executes the multiple routes (steps) connected in series simultaneously. • Either step or transition may be used at the head of parallel branch destination.
	Parallel connection (8)				• Waits for completion of execution at each route branched by parallel branch, and shifts to the succeeding step after completion of execution for all routes. Either step or transition may be used just before and after connection. • When the step just before connection is an FS step, the scan is executed even if the system is waiting. The scan is not executed after completion of waiting.
	Jump (shift) (Each symbol size)	<Normal jump> 	<Jump (connection)> 	CALL Fn JMP Pn	(1) Normal jump • Shifts the execution to the pointer Pn designated within local program after execution of last step or transition. • Either step or transition may be used at jump destination. • Even when the jump is executed from FS step to transition, the scan is executed while it is under waiting for establishment of shifting conditions for jump destination. (2) Jump (connection) • "Jump (connection)" occurs when the jump is executed to the other route within parallel branch after completion of parallel branch, in which the waiting is executed at jump destination.
				CALL Fn' Pn CALL Kn	

5.4 SFC program name

Set the "SFC program name" for each SFC program No. 0 to No. 255.
The SFC program name is set within 16 single-byte characters (double-byte: 8 characters). Designate this SFC program name for the "subroutine call/start step (GSUB)" and the "clear step (CLR)".

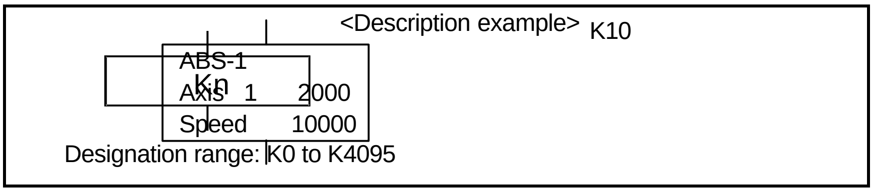
POINT

(1) The SFC program can be set to a random number between 0 and 255.
(2) "$ (single byte)" cannot be set as the first character of SFC program name.
(3) "\", "/", ":", ";", ",", ".", "*", "?", " " ", "<", ">", "|" (single byte) cannot be set in the SFC program name.

5.5 Steps

5.5.1 Motion control step

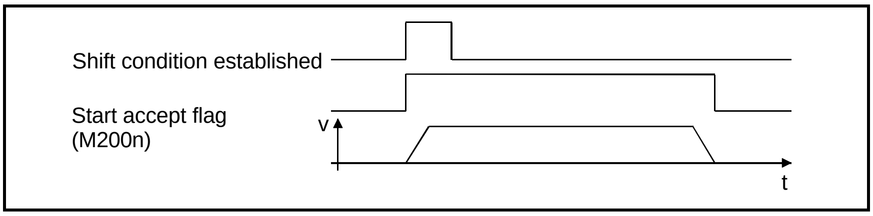
Start the servo program Kn.



(1) Description of operation

(a) The start accept flag for the axis designated in the designated servo program Kn (n = 0 to 4095) is turned ON.

(b) The designated servo program Kn (n = 0 to 4095) is started.



(2) Error

In the following case, an error will occur and execution of the SFC program will stop.

(a) When the designated servo program Kn does not exist.

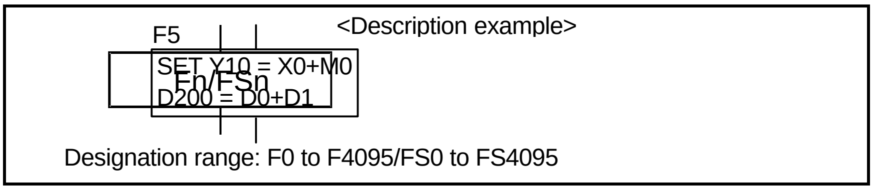
(3) Precautions

(a) To change the current value within the SFC program, designate the CHGA command using the servo program and call it at the motion control step.

(b) Even if the major or minor error occurs at or during the start of the servo program, causing the servo program to be stopped due to error, the SFC program will executed continuously. If the SFC program must be stopped when an error occurs, input the error detection conditions at the transition (shifting condition).

5.5.2 Operation control step

Execute the operation control program Fn/FSn.



(1) Description of operation

(a) 1-time execution type operation control step Fn
The operation control program Fn (n = 0 to 4095) is executed once.

(b) Scan execution type operation control step FSn
The operation control program FSn (n = 0 to 4095) is executed repeatedly until the succeeding shifting condition is established.

(2) Error

In the following case, an error will occur and execution of the SFC program will stop.

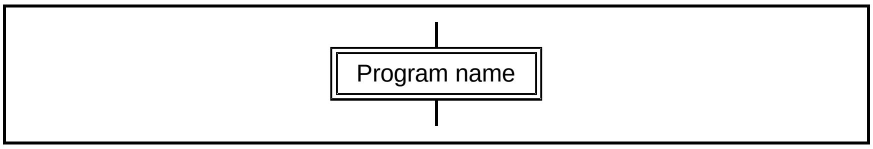
(a) When the designated operation control program Fn/FSn does not exist.

(3) Precautions

(a) If the SFC program must be stopped when an error occurs, input the error detection conditions at the transition (shifting condition).

5.5.3 Subroutine call/start step

Call/start the SFC program of designated program name.

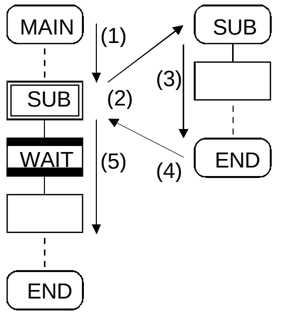


(1) Description of operation

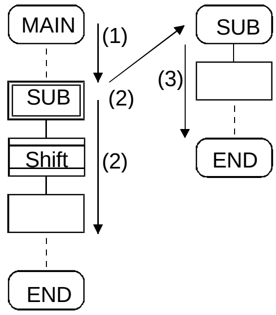
(a) The designated SFC program is called and started.

(b) The control may differ depending on the type of transition connected after the subroutine call/start step.

- WAIT .... Subroutine call
- Other than WAIT .... Subroutine start



In case of WAIT (subroutine call)



In case of other than WAIT (subroutine start)

(2) Error

In the following case, an error will occur and execution of the SFC program will stop.

(a) When the designated SFC program does not exist in the called or started subroutine.

(b) When the called/started SFC program has already been started in a calling/starting subroutine.

(c) When the local program is called or started in a called or started subroutine.

(3) Precautions

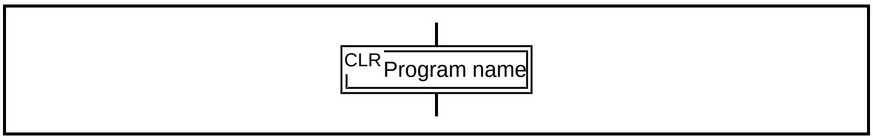
(a) The depth of nesting for subroutine call/start is not limited.

(b) For subroutine start, the start source SFC program is executed continuously even if the start destination SFC program is stopped due to error.

(c) When the call destination SFC program is stopped due to error in case of subroutine call, the execution of calling source SFC program is interrupted at the error.

5.5.4 Clear step

Interrupt the execution of SFC program of designated program name.

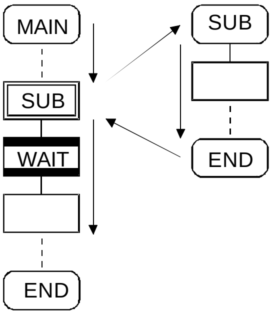


(1) Description of operation

(a) The designated SFC program being executed is interrupted.

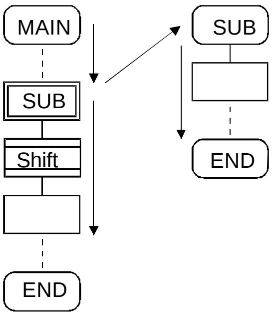
(b) Even if the SFC program (designated for "Clear") is set to automatic start, it will not start automatically after interruption.

(c) When the designated program is calling the subroutine, the execution of subroutine program called is also interrupted.



When "subroutine call" is set as shown on the left:
- When MAIN is cleared
  Execution of both MAIN and SUB is interrupted.
- When the subroutine (SUB) is cleared
  Execution of the subroutine (SUB) is interrupted to return to MAIN if the subroutine (SUB) is being executed.

(d) When the program is designated after a subroutine is started, execution of the started subroutine program continues.



When "subroutine start" is set as shown on the left:
- When MAIN is cleared
  Even if the subroutine (SUB) is being executed, execution of MAIN is interrupted. However, execution of the subroutine (SUB) continues.
- When the SUB is cleared
  Execution of only SUB is interrupted if the subroutine (SUB) is being executed.

(2) Error

An error occurs in the following case, in which the execution of SFC program is interrupted.

(a) When the SFC program designated at clear step does not exist.

(3) Precautions

(a) When the SFC program designated at clear step has not been started, it will not result in an error in particular, but is ignored.

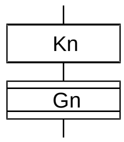
(b) Even if the execution of SFC program is stopped at the clear step, the output is held.

5.6 Transition

Either a conditional expression or operation expression can be described for transition. The operation expression described here is executed repeatedly until the shifting condition is established.

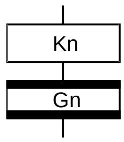
(1) Description of operation

(a) Motion control step + Shift



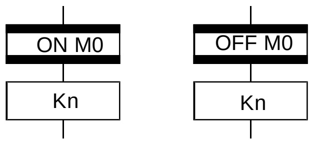
- The operation is shifted to the succeeding step when the shifting condition Gn is established without waiting for the servo program Kn (started at the motion control step) to end.

(b) Motion control step + WAIT



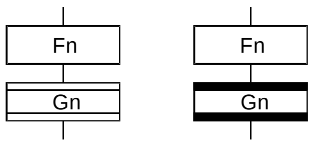
- The operation is shifted to the succeeding step when the shifting condition Gn is established after the servo program Kn (started at the motion control step) has ended.

(c) WAITON/WAITOFF + Motion control step



- The operation is started as soon as the designated bit device of WAITON/WAITOFF is turned ON/OFF.

(d) Combination with operation control step



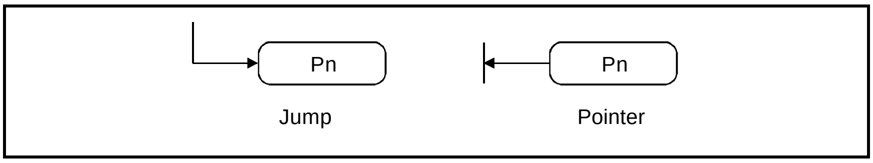
- The SHIFT and WAIT operate in the same manner. Operation is shifted to the succeeding step when the shifting condition is established after operation control program Fn is executed.

(2) Precautions

(a) Always set transition together with motion control step in a pair. If the step after the WAITON/WAITOFF is not a motion control step, execution of the SFC program will be interrupted when an error is detected.

(b) When a major or minor error occurs at the start of the servo program designated at the motion control step, thereby resulting in start failure, the SFC program is executed continuously, and shifted to the succeeding step. If the SFC program must be stopped when an error occurs, input the error detection conditions at the transition (shifting condition).

5.7 Jump/pointer



(1) Description of operation

(a) "JUMP" functions to jump to the designated pointer Pn within local program.

(b) The pointer can be set to a step, transition, branch point or connection point.

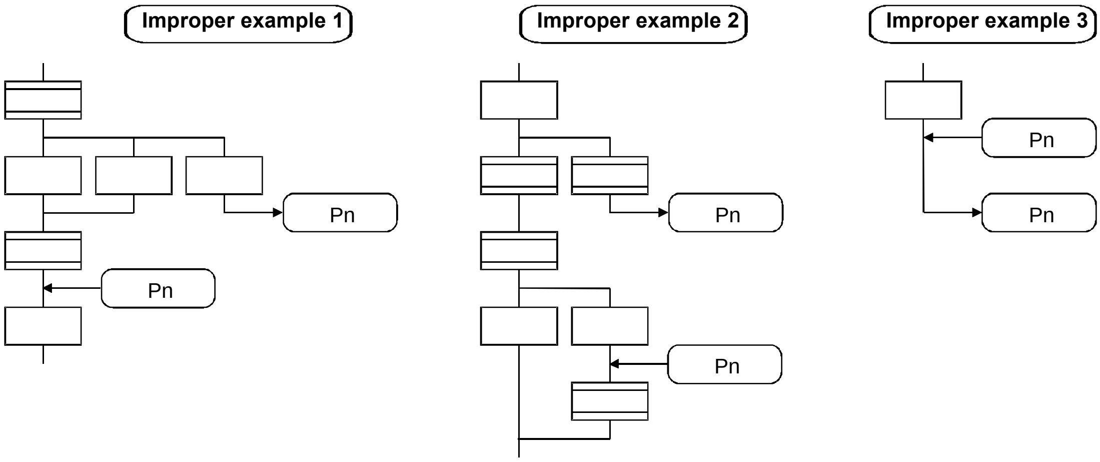
(c) Pointer Pn can be set between P0 and P1683 in each program.

(2) Precautions

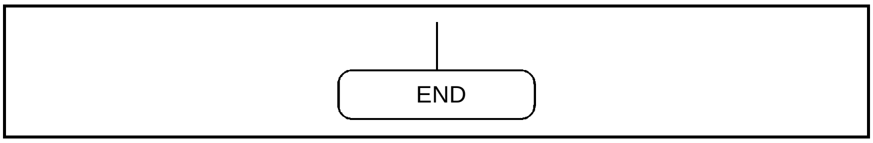
(a) JUMP cannot be set to exit a parallel branch - parallel connection. (Improper example 1 shown below)

(b) JUMP cannot be set from a parallel branch - parallel connection to a parallel branch - parallel connection. (Improper example 2 shown below)

(c) A setting that continues the label and jump cannot be set. (Improper example 3 shown below)



5.8 END



(1) Description of operation

(a) The program is ended.

(b) When the subroutine is called, the program is returned to the calling source SFC program.

(2) Precautions

(a) Multiple ENDs can be set in one program.

(b) END cannot be set between a parallel branch - parallel connection.

(c) The output is held even when the SFC program is completed with END.

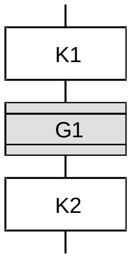
5.9 Branch/connection

5.9.1 Parallel shifting

The execution is shifted to the step or transition connected in series.

(1) To start a servo program or subroutine, and shift to the succeeding step before the operation is completed

Set the transition to SHIFT.
In this case, it is possible to omit the transition (shift).
When the transition is omitted, shifting is executed unconditionally.



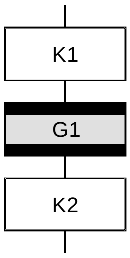
The servo program K1 is started.

The operation is shifted to the succeeding step when the condition set for transition G1 is established without waiting for the servo program K1 to end.

The servo program K2 is started.

(2) To start the servo program or subroutine and advance to the succeeding step after completion of operation

Set the transition to WAIT.



The servo program K1 is started.

The operation is shifted to the succeeding step when the axis started by servo program K1 is stopped (start accept flag: OFF) and the condition designated by transition G1 is established.

The servo program K2 is started.

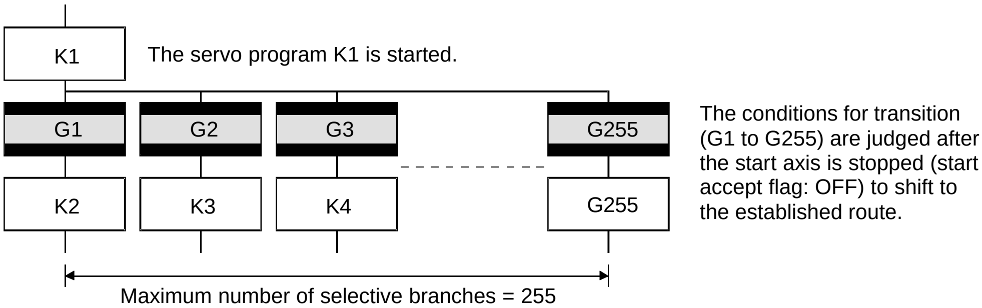
5.9.2 Selective branch/connection

(1) Selective branch

The conditions for multiple transitions (connected in parallel) are judged to execute only the route for which conditions are established first.

The transition is limited to SHIFT or WAIT.

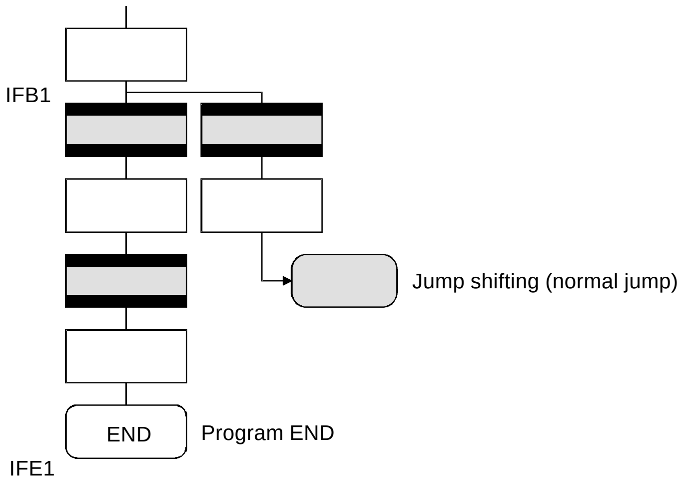
Example: WAIT



POINT
(1) The transition conditions are not always judged from left sequentially.

(2) Selective connection

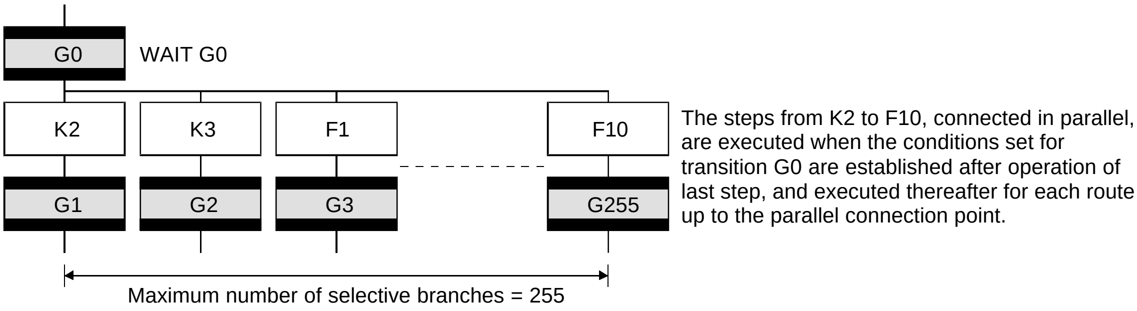
A selective connection is made to connect to one route after selective branching and processing of each route. It is also possible to set so that the route is not connected.



5.9.3 Parallel branch/parallel connection

(1) Parallel branch

Multiple steps, connected in parallel, are executed at the same time. Either step or transition may be used at the head of parallel branch destination.

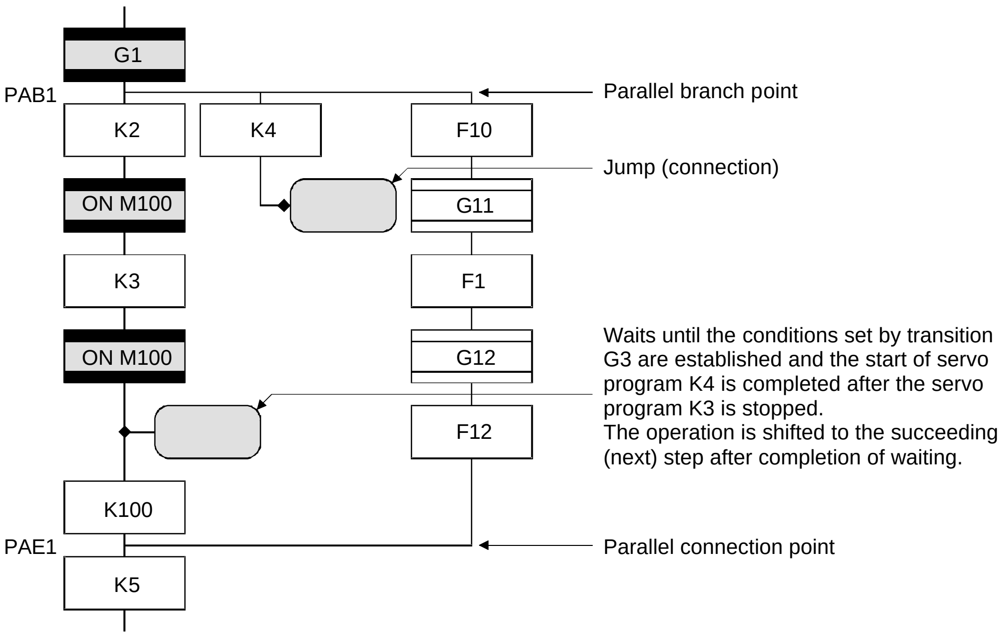


POINT
"SHIFT" or "WAIT" can be set for the transition just before parallel branch, but "WAITON" and "WAITOFF" cannot be set.

(2) Parallel connection

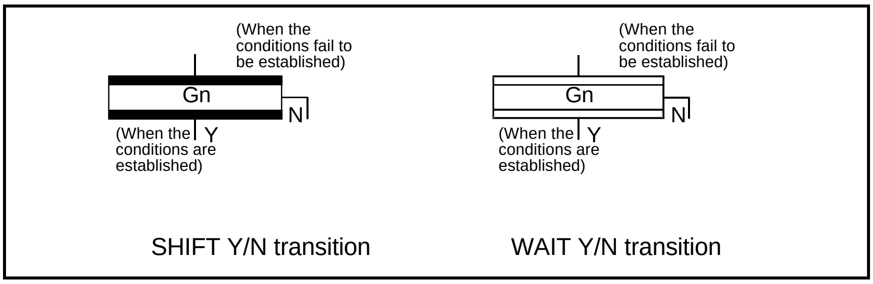
Always connect with a parallel connection when using parallel branching. Jumping to another branch route can be set between the parallel branch – parallel connection. In this case, the jump destination will be a parallel connection point in the middle (connection jump).

A jump that exits the parallel branch – parallel connection cannot be set.



5.10 Y/N transition

Use the "SHIFT Y/N transition", "WAIT Y/N transition" when the route needs to be branched according to the establishment of the shift conditions.



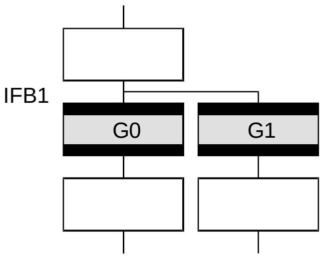
(1) Description of operation

(a) The operation is shifted to the lower step when the shifting conditions set by Gn are established, and to the step connected from right when the shifting conditions are not established.

(b) The difference between "SHIFT Y/N" and "WAIT Y/N" is the same as that between "SHIFT" and "WAIT".

(c) This allows the following selective branch program to be described easily.

<Conventional description>



- The G0 program and G1 program are limited only when the affirmation/negation of conditional expression is different.

(Example 1)

[G 0]
M0

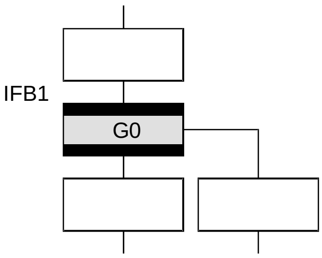
[G 1]
!M0

(Example 2)

[G 0]
D0!=K100

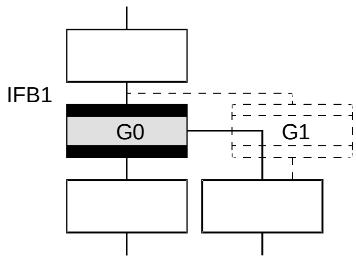
[G 1]
D0==K100

<Current description>



- Set the G0 program shown in the above (Example 1) and (Example 2).
- The converted SFC program list code is the same as the conventional description. (Only the SFC diagram expression differs.)

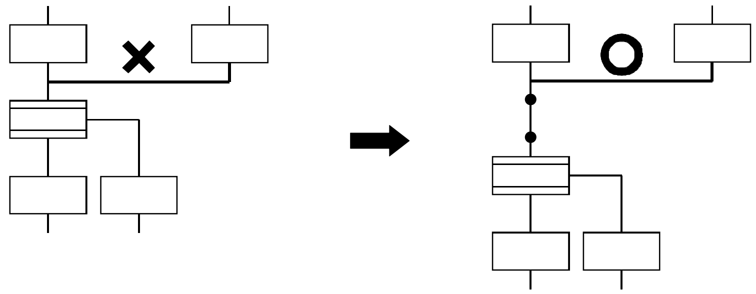
  Consequently, the program in which the part of "Automatic search → Conditional expression of vacant G-No." is logically negated, is created automatically, causing two G-programs to be occupied.



(2) Precautions

(a) To connect to just before "SHIFT Y/N" or "WAIT Y/N", insert it between "connection – branch".

- It is not possible to connect directly to "SHIFT Y/N" or "WAIT Y/N".
- Insert "connection - branch" in-between.



5.11 Task operation

The timing to execute the SFC program can be set once for each program with the program parameter. The task is roughly classified into three types as shown in the following table.

Task type	Contents
Normal task	Execution at motion main cycle (dead time)
Event task	1. Execution at constant cycle (0.8ms, 1.7ms, 3.5ms, 7.1ms, 14.2ms) 2. Execution when input set for event task, of external interrupt (16 points of QI60), is turned ON 3. Execution only by interrupt from PLC
NMI task (Non-Maskable Interrupt)	Execution when input set for NMI task, of external interrupt (16 points of QI60), is turned ON

(1) Normal task

The SFC program is executed at main cycle (dead time) of the Q motion CPU process.

POINT

(1) Set the SFC program (including motion control step) to "Normal task".

(2) Execution of normal task is interrupted while the event task/NMI task is being executed.
Note that when using a normal task, an event task prohibit command (DI) can be described in the operation control step. Thus, interruption of the event task can be prohibited in the section enclosed by the event task prohibit command (DI) and event task permit command (EI).

(2) Event task

The event task is used to execute the SFC program when an event occurs.
The event includes the following:

(a) Constant cycle
The SFC program is executed periodically at a cycle of 0.8ms, 1.7ms, 3.5ms, 7.1ms or 14.2ms.

(b) External interrupt (16 points of I0 to I15)
The SFC program is executed when the QI16 (16-point interrupt unit mounted in motion slot) input set to the event task turns ON.

(c) PLC interrupt
The SFC program is executed when the GINT command is executed by the sequence program for another units Q-PLC CPU.

POINT
(1) It is possible to set the multiple events in one SFC program. However, multiple constant cycles cannot be set. (2) The motion control step cannot be executed during an event task.

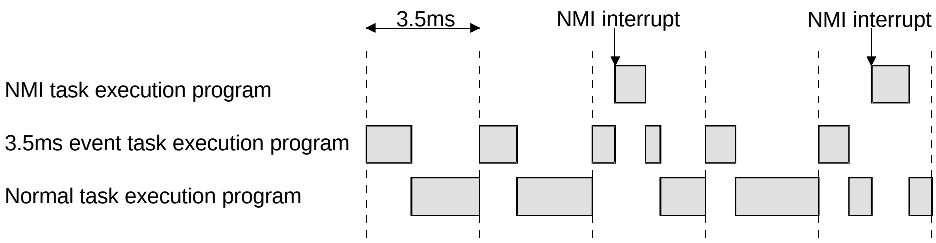
(3) NMI task

The SFC program is executed when the external interrupt (16 points of QI60) input set for the NMI task turns ON.

POINT
(1) The NMI task has the highest priority in normal task/event task/NMI task. (2) Even if the event task prohibit (DI) is executed by the normal task, the NMI task interrupt is not masked, but is executed.

(4) Execution status example

The following shows an SFC program execution example when the SFC program is executed by multiple tasks.



When there is a program to be executed by the NMI task, a program to be executed by the 3.5ms constant cycle event task and a program to be executed by the normal task, each task is executed as shown below.

(a) 3.5ms constant cycle event task is executed every 3.5ms
(b) NMI task is executed at highest priority when the NMI interrupt is input
(c) Normal task is executed during dead time

5.12 SFC parameters

The SFC parameters include "task parameters" used to control the tasks (normal task, event task, NMI task) and "program parameters" set for each SFC program.

5.12.1 Task parameters

<table>
<tr><th>No.</th><th colspan="2">Item</th><th>Setting range</th><th>Initial value</th><th>Remarks</th></tr>
<tr><td>1</td><td>Number of continuous shift lines</td><td>Normal task</td><td>1 to 30</td><td>3</td><td rowspan="2">These parameters are retrieved at the rising edge of the PLC READY signal (M2000). Then control is executed.
Turn the PLC READY signal (M2000) OFF before setting or changing those parameters.</td></tr>
<tr><td>2</td><td colspan="2">Interrupt setting</td><td>Set either "Event task" or "NMI task" for external interrupt input (I0 to I15).</td><td>Event task</td></tr>
</table>

5.12.2 Program parameters

The following parameters are set for each SFC program.

<table>
<tr><th>No.</th><th>Item</th><th colspan="2">Setting range</th><th>Initial value</th><th>Remarks</th></tr>
<tr><td>1</td><td>Start-up setting</td><td colspan="2">Execution/non-execution of automatic start</td><td>Non-execution of automatic start</td><td rowspan="6">These parameters are retrieved at the rising edge of the PLC READY signal (M2000). Then control is executed.
Turn the PLC READY signal (M2000) OFF before setting or changing those parameters.</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3">Execution task</td><td></td><td>–</td><td rowspan="3">Normal task</td></tr>
<tr><td>Event task</td><td>* When the event task is set, set the event to be validated.
Be sure to set one of following (1 to 3).
1. Constant cycle
0.8ms, 1.7ms, 3.5ms, 7.1ms or 14.2ms or none
2. External interrupt (selection from the events set for event task)
It is possible to set the multiple events from I0 to I15.
3. PLC interrupt

It is possible to set the multiple events for 1 to 3 shown above.
The same event may be shared by multiple SFC programs.</td></tr>
<tr><td>NMI task</td><td>* When the NMI task is set, set the interrupt to be validated.
1. External interrupt (select from the events set for event task)
It is possible to set the multiple events from I0 to I15.</td></tr>
<tr><td>3</td><td colspan="2">Number of continuous shift lines</td><td>1 to 10
* Set the number of continuous shift lines set for the program set in the event task or NHI task.</td><td>1</td></tr>
<tr><td>4</td><td colspan="2">END operation</td><td>End/continue
* Set the operation mode of END step for the program set in the event task or NMI task.</td><td>END</td></tr>
</table>

5.13 SFC program start method

The SFC program is executed while the PLC READY M2000 signal is ON.
The SFC program can be started by the following three methods.

(1) Automatic start
(2) Start from SFC program
(3) Start from PLC

The start method is set for each SFC program using the program parameter.

(1) Automatic start

The SFC program is started automatically when the PLC READY M2000 is turned ON.

(2) Start from SFC program

The SFC program is started when the subroutine is called or the start step is executed within the SFC program.

(3) Start from PLC

The SFC program is started when the SFCS command is executed within the sequence program.

5.14 SFC program end method

The SFC program is ended in the following three methods:

(1) End program by executing END set within SFC program.
(2) End SFC program by turning PLC READY signal M2000 OFF.
(3) End with clear step.

POINT
(1) Multiple ENDs can be set within one SFC program. (2) The SFC program will end even if "Automatic Start" is set.

Chapter 6 SV22 Servo Programs

6.1 Servo program

The servo program is used to designate the type of positioning control and the positioning data required for carrying out positioning control. The servo program configuration and designation methods are explained in this section.

With the SV13 and SV22, the servomotor is controlled with this servo program. However, the servo commands that can be used are listed in the "List of servo commands".

6.1.1 Servo program configuration

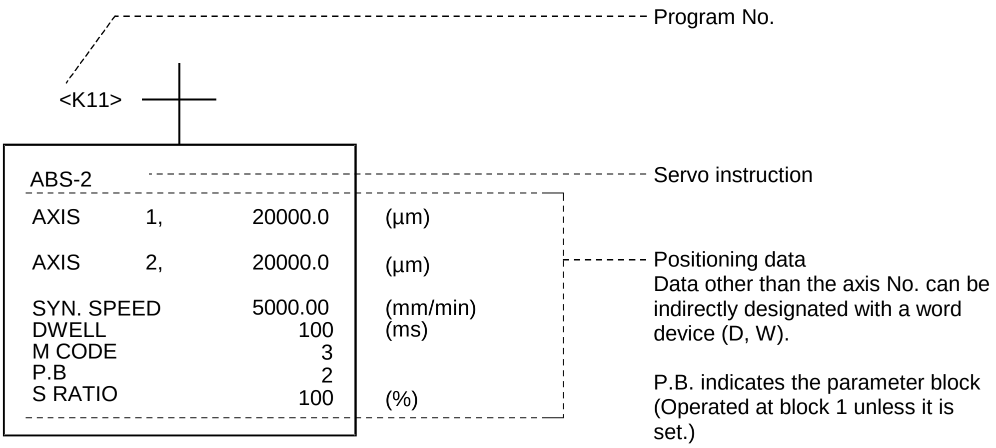
One servo program is configured of the parts (1) to (3) shown below.

(1) Program No. .... This number designates the start request in the sequence program. A random number between 0 and 4095 can be set.

0 to 4095

(2) Servo command .... This indicates the type of positioning control.

(3) Positioning data .... This is data required to execute the servo commands. The data required for execution is determined or each servo command.



(4) Servo program area

1) The servo program area is used to store the servo program created with a peripheral device. The memory in the positioning CPU is large enough to store 14334 steps (14K steps), and is backed up by the SRAM battery.
(This is not the sequence program storage internal memory.)

6.1.2 List of servo commands

The commands listed below are available, but the usage validity differs according to the CPU OS.

Positioning control		Instruction symbol	Processing details	Number of steps	Positioning data: Common settings: Parameter block No.	Axis	Address/Travel value	Command speed	Dwell time	M code	Torque limit value	Circular/helical: Auxiliary point	Radius	Center point	Number of pitches	Parameter block: Control unit	Speed limit value	Acceleration time	Deceleration time	Sudden stop deceleration time	Torque limit value	Deceleration processing on STOP input	Allowable error range for circular interpolation	S curve ratio
Linear control	1-axis	ABS-1	Absolute 1-axis positioning	4	△	○	○	○	△	△	–	–	–	–	–	–	△	△	△	△	△	△	–	△
		INC-1	Incremental 1-axis positioning	4	△	○	○	○	△	△	–	–	–	–	–	–	△	△	△	△	△	△	–	△
	2-axis	ABS-2	Absolute 2-axis linear interpolation	5	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		INC-2	Incremental 2-axis liner interpolation	5	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
	3-axis	ABS-3	Absolute 3-axis linear interpolation	7	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		INC-3	Incremental 3-axis linear interpolation	7	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
	4-axis	ABS-4	Absolute 4-axis linear interpolation	8	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		INC-4	Incremental 4-axis linear interpolation	8	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
Circular interpolation control	Auxiliary point designation	ABS	Absolute circular interpolation by auxiliary point designation	7	△	○	○	○	△	△	–	○	–	–	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by auxiliary point designation	7	△	○	○	○	△	△	–	○	–	–	–	△	△	△	△	△	△	△	–	△
	Radius designation	ABS	Absolute circular interpolation by radius designation, within CW180°	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		ABS	Absolute circular interpolation by radius designation, CW180° and greater	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		ABS	Absolute circular interpolation by radius designation, within CCW180°	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		ABS	Absolute circular interpolation by radius designation, CCW180° and greater	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by radius designation, within CW180°	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by radius designation, CW180° and greater	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by radius designation, within CCW180°	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by radius designation, CCW180° and greater	6	△	○	○	○	△	△	–	–	○	–	–	△	△	△	△	△	△	△	–	△
	Center point designation	ABS	Absolute circular interpolation by center point designation, CW	7	△	○	○	○	△	△	–	–	–	○	–	△	△	△	△	△	△	△	–	△
		ABS	Absolute circular interpolation by center point designation, CCW	7	△	○	○	○	△	△	–	–	–	○	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by center point designation, CW	7	△	○	○	○	△	△	–	–	–	○	–	△	△	△	△	△	△	△	–	△
		INC	Incremental circular interpolation by center point designation, CCW	7	△	○	○	○	△	△	–	–	–	○	–	△	△	△	△	△	△	△	–	△

Operates as one block if not set

○: Item that must be set
△: Item set as necessary
– : Item that cannot be set
* : Possible

<Number of steps>
The number of steps in each command is the value when the ○ mark items are set. The step is decremented by one each time a △ mark item is set.

	Others: Repeat condition	Others: Program No.	Others: Cancel	Others: Start	Speed change		SV13 for transfer assembly	SV22 for automatic machine Real mode	SV22 for automatic machine Virtual mode
	–	–	–	–	*		○	○	○
	–	–	–	–	*				
	–	–	–	–	*				
	–	–	–	–	*				
	–	–	–	–	*				
	–	–	–	–	*				
	–	–	–	–	*		○	○	○
	–	–	–	–	*				
	–	–	–	–	–		○	○	○
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				

○ mark: Usable
× mark: Not usable

Positioning control		Instruction symbol	Processing details	Number of steps	Positioning data																			
					Common settings							Circular/helical				Parameter block								
					Parameter block No.	Axis	Address/Travel value	Command speed	Dwell time	M code	Torque limit value	Auxiliary point	Radius	Center point	Number of pitches	Control unit	Speed limit value	Acceleration time	Deceleration time	Sudden stop deceleration time	Torque limit value	Deceleration processing on STOP input	Allowable error range for circular interpolation	S curve ratio
Fixed dimension feed	1 axis	FEED-1	1 axis fixed-dimension feed start	4	△	○	○	○	△	△	–	–	–	–	–	–	△	△	△	△	△	△	–	△
	2 axis	FEED-2	2 axis linear interpolation Fixed-dimension feed start	5	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
	3 axis	FEED-3	3 axis linear interpolation Fixed-dimension feed start	7	△	○	○	○	△	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
Speed control	Forward rotation	VF	Speed control Forward rotation start	3	△	○	–	○	–	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
	Reverse rotation	VR	Speed control Reverse rotation start	3	△	○	–	○	–	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
Speed control II	Forward rotation	VVF	Speed control II Forward rotation start	3	△	○	–	○	–	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
	Reverse rotation	VVR	Speed control II Reverse rotation start	3	△	○	–	○	–	△	–	–	–	–	–	△	△	△	△	△	△	△	–	△
Speed/position changeover control	Forward rotation	VPF	Speed/position changeover control Forward rotation start	4	△	○	○	○	△	△	△	–	–	–	–	△	△	△	△	△	△	△	–	△
	Reverse rotation	VPR	Speed/position changeover control Reverse rotation start	4	△	○	○	○	△	△	△	–	–	–	–	△	△	△	△	△	△	–	–	△
	Re-start	VPSTART	Speed/position changeover control Restart	2	–	○	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Speed changeover control (Max. 3 axis)		VSTART	Speed changeover control, start	1	△	–	–	–	–	–	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		VEND	Speed changeover control, end	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
		VABS	Absolute designation of speed changeover point	4	–	–	○	○	–	△	△	–	–	–	–	–	–	–	–	–	–	–	–	–
		VINC	Incremental designation of speed changeover point	4	–	–	○	○	–	△	△	–	–	–	–	–	–	–	–	–	–	–	–	–
Position follow-up control		PFSTART	Position follow-up control start	4	△	○	○	○	–	△	△	–	–	–	–	–	△	△	△	△	△	△	–	△
Constant speed control		CPSTART1	1 axis constant speed control start	3	△	○	–	○	–	–	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		CPSTART2	2 axis constant speed control start	3	△	○	–	○	–	–	–	–	–	–	–	△	△	△	△	△	△	△	△	△
		CPSTART3	3 axis constant speed control start	4	△	○	–	○	–	–	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		CPSTART4	4 axis constant speed control start	4	△	○	–	○	–	–	–	–	–	–	–	△	△	△	△	△	△	△	–	△
		CPEND	Uniform speed control end	1	–	–	–	–	△	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
For repetition of uniform control, speed changeover and uniform speed control		FOR-TIMES	Repetition range head setting	2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	△
		FOR-ON																						
		FOR-OFF																						
		NEXT	Repetition range end setting	3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Simultaneous start		START	Simultaneous start (Max. 3 program)	3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	△
Zero point return		ZERO	Zero point return start	7	–	○	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
High-speed oscillation		OSC	High-speed oscillation	5	△	○	○	○	–	△	–	–	–	–	–	–	–	–	–	–	△	–	–	–

↑ Operates as one block if not set (Parameter block No.)

○ : Item that must be set
△ : Item set as necessary
– : Item that cannot be set
* : Possible

	Others: Repeat condition	Others: Program No.	Others: Cancel	Others: Program	Speed change		SV13 for transfer assembly	SV22 for automatic machine Real mode	SV22 for automatic machine Virtual mode
	–	–	–	–	*		○	○	○
	–	–	–	–	*				
	–	–	–	–	*				
	–	–	–	–	*		○	○	×
	–	–	–	–	*				
	–	–	–	–	*		○	○	○
	–	–	–	–	*				
	–	–	–	–	*		○	○	×
	–	–	–	–	*				
	–	–	–	–	–				
	–	–	–	–	–		○	○	○
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	*		○	○	○
	–	–	–	–	*		○	○	○
	–	–	–	–	*		○	○	○
	–	–	–	–	*				
	–	–	–	–	*		○	○	○
	–	–	–	–	–		○	○	○
	○	–	–	–	–		○	○	○
	–	–	–	–	–		○	○	○
	–	○	–	–	*				
	–	–	–	–	–				×
	–	–	–	–	–				

○ mark: Usable
× mark: Not usable

<Number of steps>
The number of steps in each command is the value when the ○ mark items are set. The step is decremented by one each time a △ mark item is set.
<Torque limit value of common item>
For VPF/VPR commands, it is validated after changing to "Position control".
For VABS/VINC commands, it is validated after changing to "Speed control".
<START command>
It is unable to set the program No. indirectly.

Positioning control		Instruction symbol	Processing details	Number of steps	Positioning data																			
					Common settings							Circular/helical				Parameter block								
					Parameter block No.	Axis	Address/Travel value	Command speed	Dwell time	M code	Torque limit value	Auxiliary point	Radius	Center point	Number of pitches	Control unit	Speed limit value	Acceleration time	Deceleration time	Sudden stop deceleration time	Torque limit value	Deceleration processing on STOP input	Allowable error range for circular interpolation	S curve ratio
Helical interpolation control	Auxiliary point designation	ABS	Absolute helical interpolation by auxiliary point designation	10	△	○	○	○	△	△	–	○	–	–	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by auxiliary point designation	10	△	○	○	○	△	△	–	○	–	–	○	△	△	△	△	△	△	△	–	△
	Radius designation	ABS	Absolute helical interpolation by radius designation, within CW180°	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		ABS	Absolute helical interpolation by radius designation, CW180° and greater	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		ABS	Absolute helical interpolation by radius designation, within CCW180°	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		ABS	Absolute helical interpolation by radius designation, CCW180° and greater	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by radius designation, within CW180°	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by radius designation, CW180° and greater	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by radius designation, within CCW180°	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by radius designation, CCW180° and greater	9	△	○	○	○	△	△	–	–	○	–	○	△	△	△	△	△	△	△	–	△
	Center point designation	ABS	Absolute helical interpolation by center point designation, CW	10	△	○	○	○	△	△	–	–	–	○	○	△	△	△	△	△	△	△	–	△
		ABS	Absolute helical interpolation by center point designation, CCW	10	△	○	○	○	△	△	–	–	–	○	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by center point designation, CW	10	△	○	○	○	△	△	–	–	–	○	○	△	△	△	△	△	△	△	–	△
		INC	Incremental helical interpolation by center point designation, CCW	10	△	○	○	○	△	△	–	–	–	○	○	△	△	△	△	△	△	△	–	△
3D orthogonal interpolation		CPSTART XYZ	3D orthogonal interpolation CP start		△	○	–	○	–	–	–	–	–	–	–	△	△	△	△	△	△	△	△	△
		CPEND	3D orthogonal interpolation CP end		–	–	–	–	△	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Current value change	Servo	CHGA	Servo/virtual servo current value change	3	–	○	○	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Encoder	CHGA-B	Encoder current value change	3	–	○	○	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Cam	CHGA-C	Cam shaft current value change	3	–	○	○	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

↑ Operates as one block if not set (Parameter block No.)

○: Item that must be set
△: Item set as necessary
–: Item that cannot be set
*: Possible

<Number of steps>

The number of steps in each command is the value when the ○ mark items are set. The step is decremented by one each time a △ mark item is set.

	Others				Speed change		SV13 for transfer assembly	SV22 for automatic machine Real mode	SV22 for automatic machine Virtual mode
	Repeat condition	Program No.	Cancel	Start					
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–		○	○	○
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
	–	–	–	–	–				
							×	×	×
							○	○	○
							×	×	○
							×	×	○

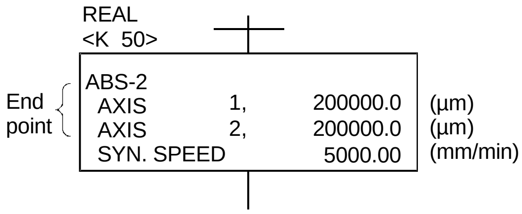
○ mark: Usable
× mark: Not usable

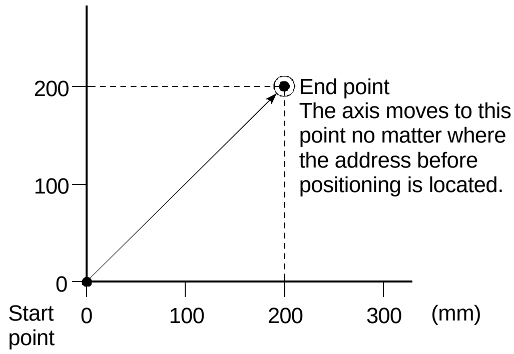
6.1.3 Linear control

1 to 4-axis control with ABS-1 to ABS-4 (absolute method)

(1) Using the zero point as a reference, positioning control is carried out from the current stopped address (address before positioning) to the designated address.

(2) The movement direction is determined according to the currently stopped address and designated address.





1 to 4-axis linear control with INC-1 to INC-4 (increment method)

(1) Positioning control is carried out from the currently stopped position's address by the designated movement amount.

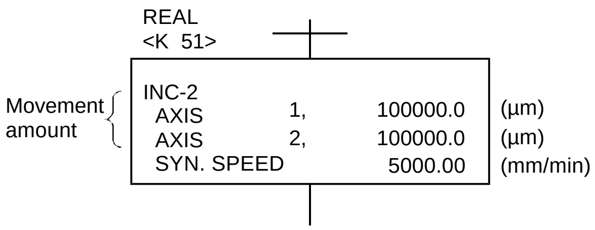
(2) The movement direction is determined according to the movement amount sign (+/–).

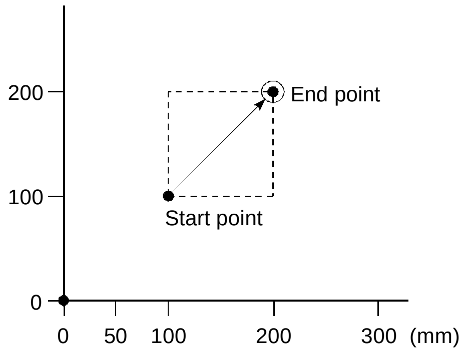
1) When movement direction is positive

Forward direction (address increment direction) positioning

1) When movement direction is negative

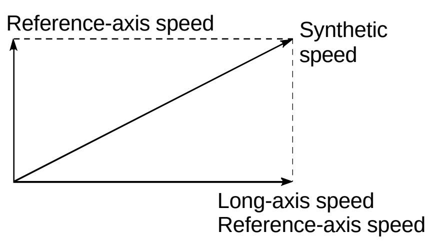
Reverse direction (address decrement direction) positioning





Speed designation for linear 2-axis, 3-axis and 4-axis interpolation control (according to speed type)

1. Synthetic speed
   This designates the speed to move with interpolation.
2. Long-axis speed
   This designates the speed for the interpolation axis with the longest movement distance.
   (The longest axis is automatically determined and processed.)
3. Reference-axis speed
   This designates the speed for the interpolation axis to be used as the reference.



6.1.4 Circular interpolation control using auxiliary point designation

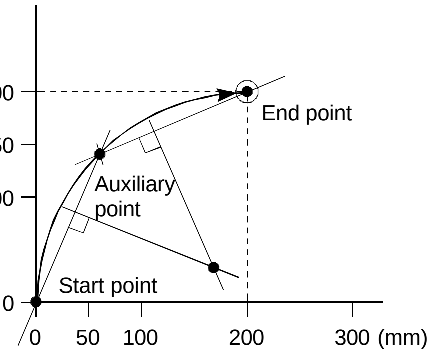
2-axis control with ABS (absolute method)

(1) Circular interpolation from the current stop address (address before positioning) through the designated auxiliary point address to the end point address, using the zero point as the reference.

(2) The center of the arc is the point of intersection of the perpendicular bisectors of the start point address (current stop address) to the auxiliary point address, and the auxiliary point address to the end point address.

REAL
<K 52>

	ABS			
End point	AXIS	1,	200000.0	(µm)
	AXIS	2,	200000.0	(µm)
	SPEED		5000.00	(mm/min)
	PAS-PT.	1,	50000.0	(µm)
	PAS-PT.	2,	150000.0	(µm)



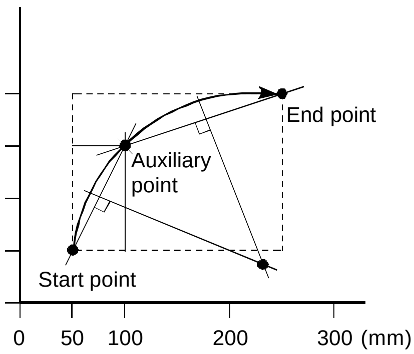
2-axis control with INC (increment method)

(1) Circular interpolation takes place from the current stop address, past the designated auxiliary point to the end point.

(2) An arc, having as its center point the intersection of the vertical bisector consisting of the start point (current stopped position) to auxiliary point, and auxiliary point to end point, is formed.

REAL
<K 53>

	INC			
End point	AXIS	1,	200000.0	(µm)
	AXIS	2,	150000.0	(µm)
	SPEED		5000.00	(mm/min)
	PAS-PT.	1,	50000.0	(µm)
	PAS-PT.	2,	100000.0	(µm)



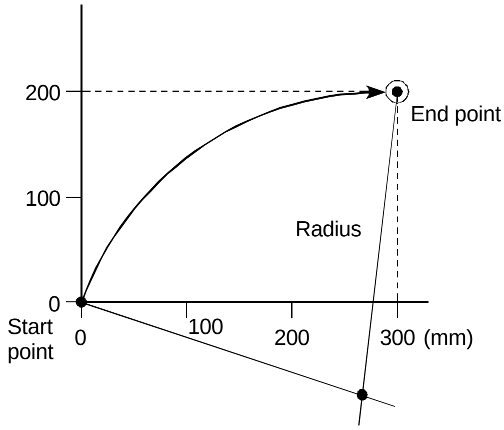
6.1.5 Circular interpolation control using radius designation

2-axis control with ABS , ABS , ABS , and ABS (absolute method)

(1) Circular interpolation of an arc of the designated radius from the current stop address (address before positioning) to the designated end point address, using the home position as the reference.

(2) The center of the arc lies at the point of intersection of the designated radius and the perpendicular bisector of the start point address (present stop address) to the end point address.

```
            REAL
            <K 54>
        ABS
End       AXIS      1,     300000.0   (µm)
point     AXIS      2,     200000.0   (µm)
          SPEED              8000.00  (mm/min)
          RADIUS           300000.0   (µm)
```

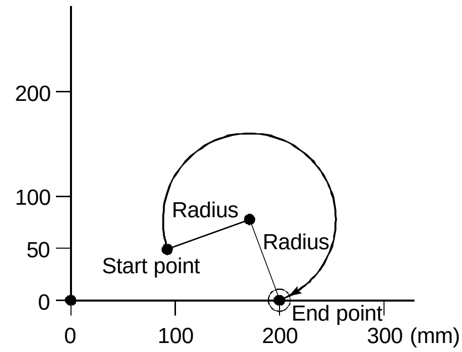


2-axis control with INC , INC , INC , and INC (increment method)

(1) Circular interpolation of an arc of the designated radius from the current stop address (0, 0) to the designated end point address.

(2) The center of the arc lies at the point of intersection of the designated radius and the perpendicular bisector of the start point address (present stop address) to the end point address.

```
            REAL
            <K 55>
          INC
Movement    AXIS     1,     100000.0   (µm)
amount      AXIS     2,     -50000.0   (µm)
            SPEED              8000.00 (mm/min)
            RADIUS            70000.0  (µm)
```

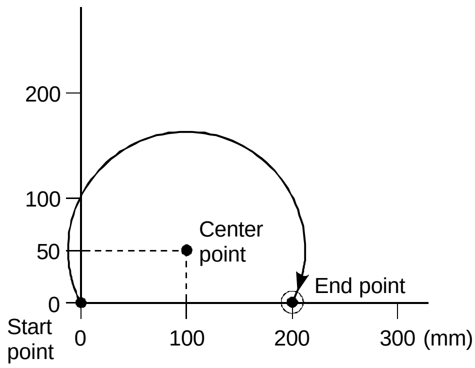


6.1.6 Circular interpolation control using center point designation

2-axis control with ABS ↻, ABS ↺ (absolute method)

(1) Using the currently stopped address (address before positioning) having the zero point as its reference as the start point, circular interpolation is carried out to the end point address with an arc having a radius that is the distance to the center point.

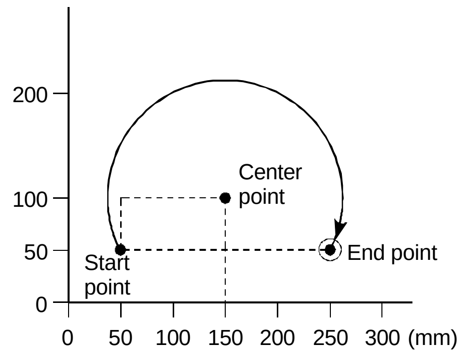
```
         REAL
         <K  56>
         ABS ↻
End    { AXIS      1,   200000.0   (µm)
point  { AXIS      2,        0.0   (µm)
         SPEED               5000.00   (mm/min)
         CENTER    1,   100000.0   (µm)
         CENTER    2,    50000.0   (µm)
```



2-axis control with INC ↻, INC ↺ (incremental method)

(1) Using the currently stopped address as the start point (0,0), circular interpolation using the movement amount is carried out to the end point with an arc having a radius that is the distance to the center point.

```
          REAL
          <K  57>
          INC ↻
Movement { AXIS      1,   200000.0   (µm)
amount   { AXIS      2,        0.0   (µm)
          SPEED               5000.00   (mm/min)
          CENTER    1,   100000.0   (µm)
          CENTER    2,    50000.0   (µm)
```



6.1.7 Fixed-dimension feed control

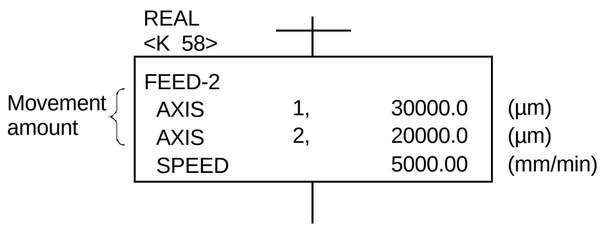
1-axis to 3-axis control with FEED-1, FEED-2 and FEED3 (incremental method)

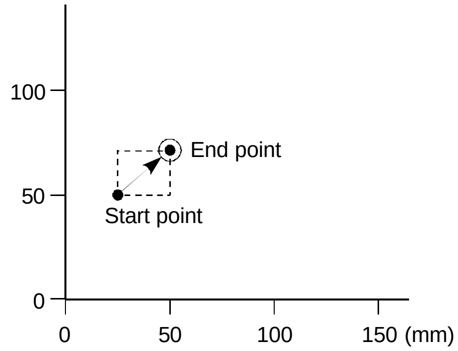
(1) Positioning control is executed for the designated movement amount from the current stop position (0).

(2) The travel direction is designated by the sign of the travel value, as follows:

  (a) Positive movement amount .... Forward direction (increased address)

  (b) Negative movement amount... Reverse direction (decreased address)





6.1.8 Speed control

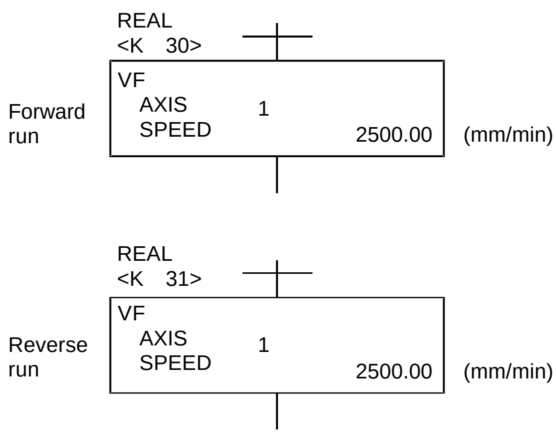
1-axis control with VF, VR, VVF, VVR

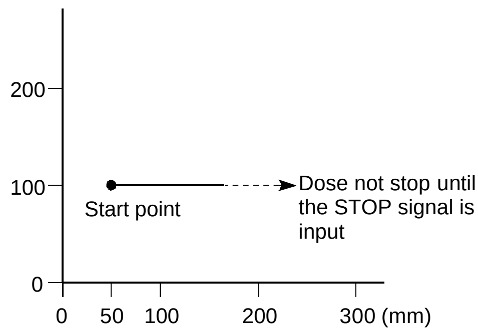
(1) After the servomotor is started, control is carried out at the designated speed until the stop command is input.

  (a) VF ......Forward direction start
  (b) VR......Reverse direction start
  — Executes the control including position loop for control of servo amplifier.

  (c) VVF....Forward direction start
  (d) VVR ...Reverse direction start
  — Executes the speed control excluding position loop for control of servo amplifier. Consequently, the deviation will not be excessive, and it is applicable to the stopper, etc.

(2) The current value remains 0 and does not change.





6.1.9 Speed/position changeover control

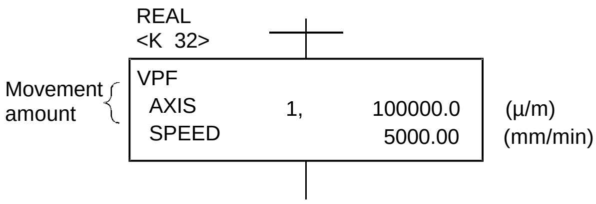
1-axis control with VPF, VPR (increment method)

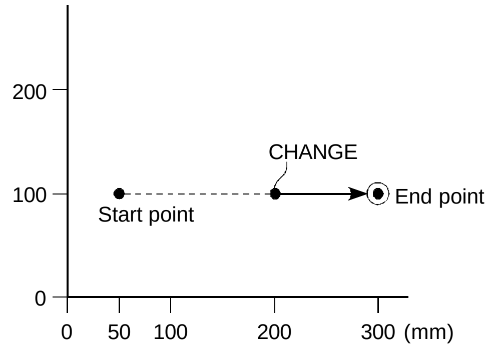
(1) After the servomotor starts, speed control is carried out. When the speed/position changeover enable signal (M3205/axis 1) is ON, the control will change from speed control to position control with the CHANGE (speed/position changeover) signal from an external source, and positioning will take place for the designated movement amount.

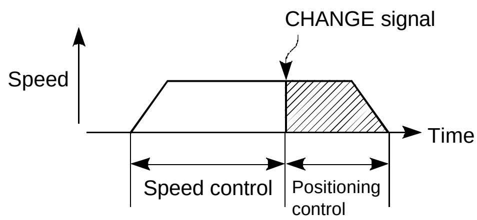
(a) VPF....Forward direction (address increment direction) start

(b) VPR ...Reverse direction (address decrement direction) start

(2) The designated positioning is carried out with the increment method when the CHANGE external signal is input.







REMARK

The response is not delayed after the CHANGE external signal is input.

6.1.10 Speed changeover control

1-axis to 3-axis control using VSTART, ABS-1, ABS-2, ABS-3, VEND (absolute method)

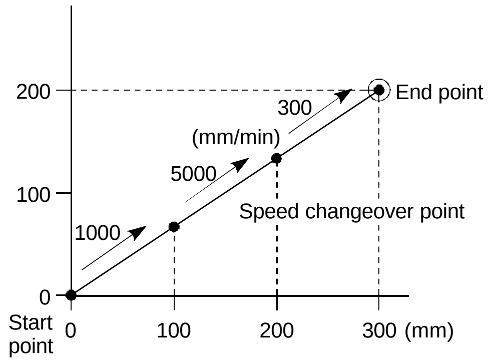
(1) Using the currently stopped address, having the zero point as the reference as the start point, positioning control is carried out to the end point while relaying the speed changeover point.
(Speed changeover point is only for VABS)

Absolute method

An address that results in reverse run cannot be designated.

REAL
<K 59>

	VSTART			
	ABS-2			
End point	AXIS	1,	300000.0	(µm)
	AXIS	2,	200000.0	(µm)
	SPEED		300.00	(mm/min)
	VABS			
Speed change point	AXIS	1,	100000.0	(µm)
	SPEED		1000.00	(mm/min)
	VABS			
Speed change point	AXIS	1,	200000.0	(µm)
	SPEED		5000.00	(mm/min)
	VEND			



1-axis to 3-axis control using VSTART, INC-1, INC-2, INC-3, VEND (increment method)

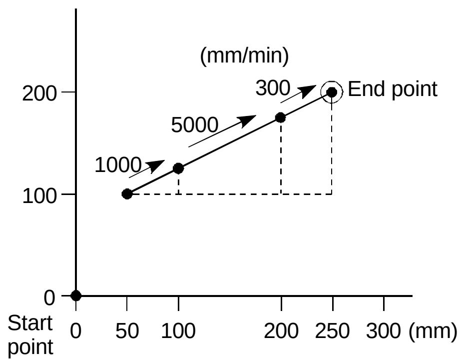
(1) Using the currently stopped address as the start point (0,0), positioning control is carried out to the end point while relaying the speed changeover point.
(Speed changeover point is only for VINC)

Increment method

An address that results in reverse run cannot be designated.

REAL
<K 60>

	VSTART			
	INC-2			
Movement amount	AXIS	1,	200000.0	(µm)
	AXIS	2,	100000.0	(µm)
	SPEED		300.00	(mm/min)
	VINC			
Speed change point	AXIS	1,	50000.0	(µm)
	SPEED		1000.00	(mm/min)
	VINC			
Speed change point	AXIS	1,	100000.0	(µm)
	SPEED		5000.00	(mm/min)
	VEND			



6.1.11 Constant-speed control

1-axis to 4-axis control with CPSTART1 to CPSTART4, CPEND

(1) With one start, positioning control is carried out at a uniform speed to the end point address while relaying the pass point.

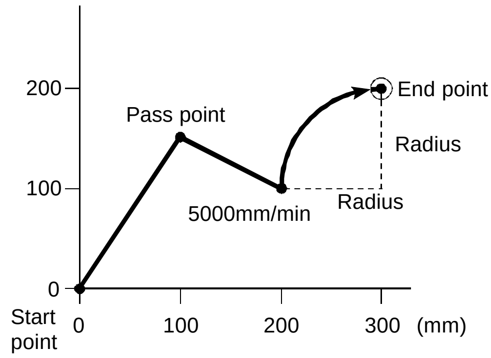
Pass point

ABS-2, ABS-3, ABS-4, ABS, ABS, ABS, ABS,
ABS, ABS, ABS, ABH, ABH, ABH,
ABH, ABH, ABH, ABH
INC-2, INC-3, INC-4, INC, INC, INC, INC,
INC, INC, INC, INH, INH, INH,
INH, INH, INH, INH

With the absolute method and incremental method, the pass point command is determined by ABS/INC, and both can be used.

REAL
<K 61>

	CPSTART2			
Speed designation	AXIS	1,		
	AXIS	2,		
	SPEED		5000.00	(mm/min)
Pass point	ABS-2			
	AXIS	1,	100000.0	(µm)
	AXIS	2,	150000.0	(µm)
	INC-2			
	AXIS	1,	100000.0	(µm)
	AXIS	2,	-50000.0	(µm)
The end point is before CPEND	INC			
	AXIS	1,	100000.0	(µm)
	AXIS	2,	100000.0	(µm)
	RADIUS		100000.0	(µm)
	CPEND			



6.1.12 Repeated control (for speed changeover control and uniform speed control)

1-axis to 4-axis control using FOR-TIMES, FOR-ON, FOR-OFF, NEXT

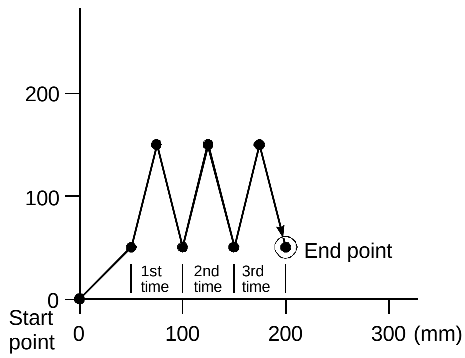
(1) The speed changeover point VABS and VINC commands for speed changeover control are repeatedly executed.

(2) The uniform speed control pass point ABS and INC commands are repeatedly executed.

(3) Designating the number of repetitions
FOR-TIMES designates the number of repetitions as K1 to K32767 and a number, or indirectly designates with D, W.
FOR-ON designates the bit device X, Y, M, L, B, F to be repeated until it turns ON.
FOR-OFF designates the bit device X, Y, M, L, B, F to be repeated until it turns OFF.

```
REAL
<K  62>
CPSTART2
  AXIS        1,
  AXIS        2,
  SPEED                5000.00    (mm/min)
ABS-2
  AXIS        1,       50000.0    (µm)
  AXIS        2,       50000.0    (µm)
FOR-TIMES
                       K      3
INC-2
  AXIS        1,       30000.0    (µm)
  AXIS        2,      100000.0    (µm)
INC-2
  AXIS        1,       20000.0    (µm)
  AXIS        2,     -100000.0    (µm)
NEXT
CPEND
```

Repeat three times.



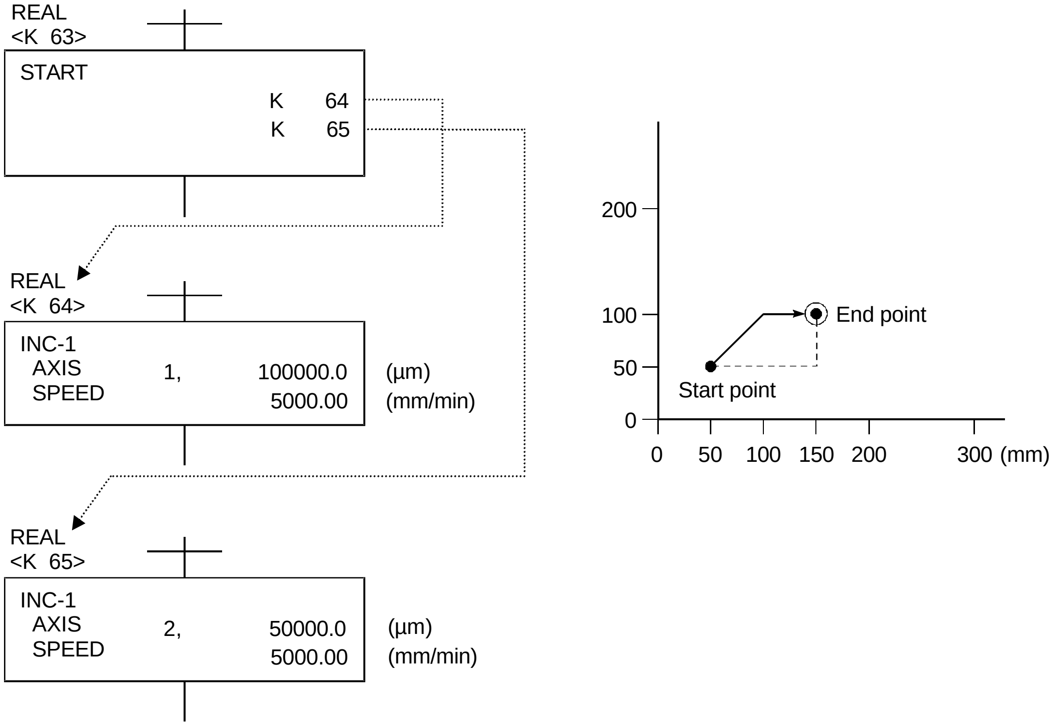
6.1.13 Simultaneous start

Simultaneous start control using START

(1) Two to three types of servo programs (excluding START) are simultaneously started.

(3) If three servo programs are for 1-axis to 4-axis control, up to 12 axes can be simultaneously started.

(3) The servo program No. designated with the START command cannot be indirectly designated with a word device (D, W).



6.1.14 Zero point return

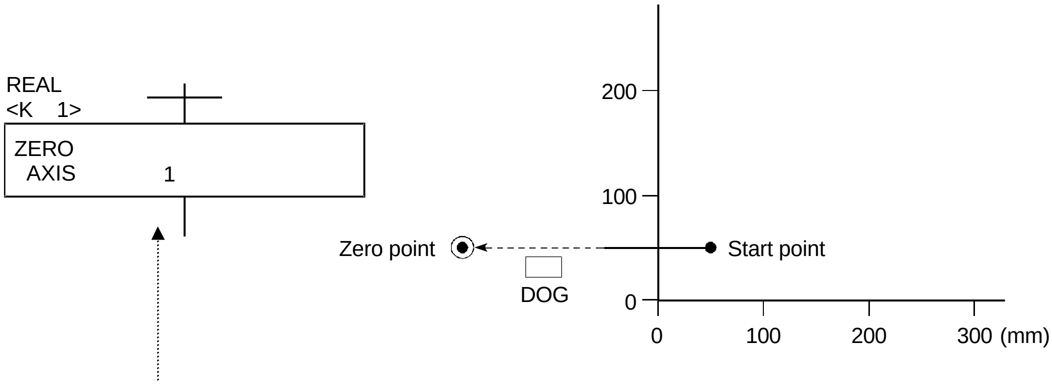
1-axis zero point return using ZERO

(1) Zero point return is executed from the currently stopped position using the method designated in the zero point return data.

(2) When the near-point dog type or count type is designated, the axis will move in the return direction designated in the zero point return data.

(3) When the data set type is designated, the stopped address will be used as the zero point and the axis will not move.

(4) The axis No. cannot be designated indirectly.



Only one axis can be designated.
Zero point return for another axis is created with a separate servo program.

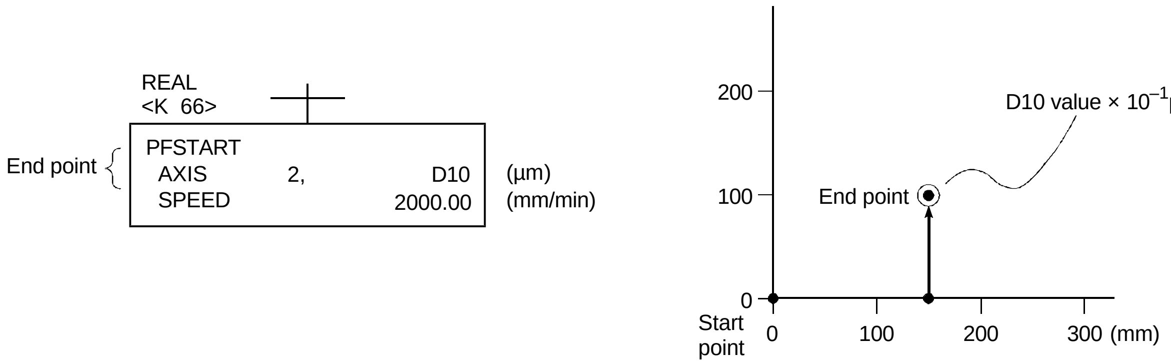
REMARK

Zero point return can be simultaneously started by calling the ZERO command subprogram with the START command.

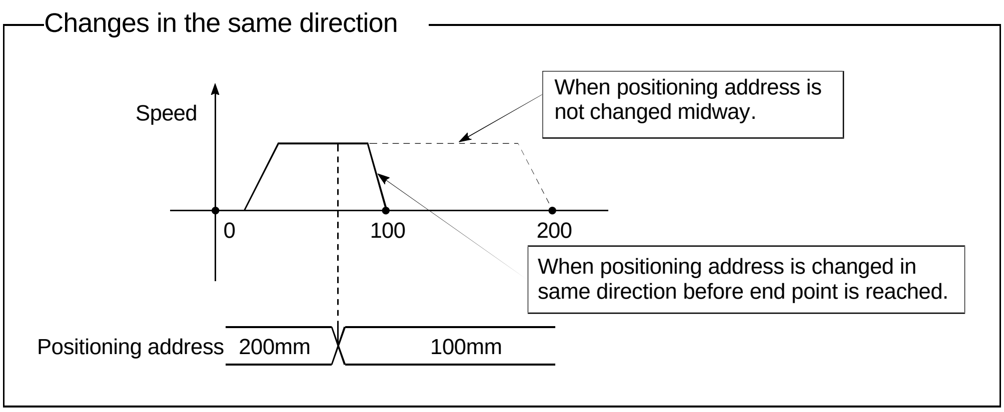
6.1.15 Position follow-up control

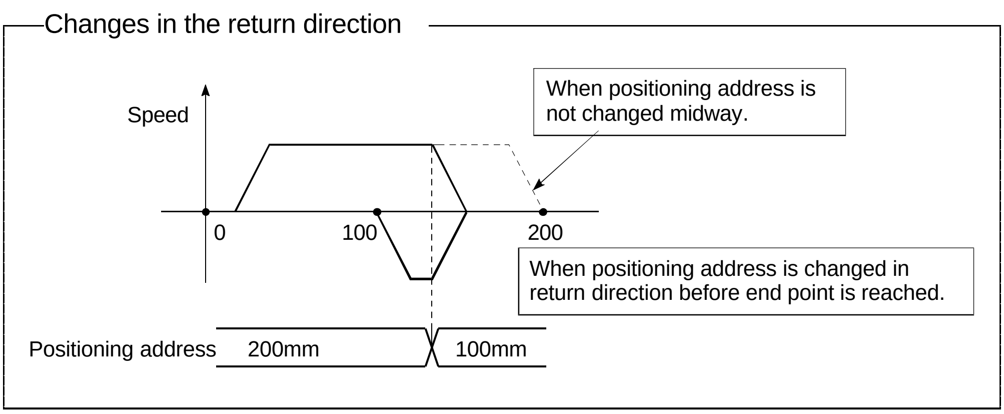
1-axis control using PFSTART (absolute method)

(1) With the first start, the axis is positioned to the address word device (D, W even number) set in the servo program.
(The axis will follow-up if the contents of D, W and # are changed.)



(2) The axis will move as follows if the contents of the word device are changed midway.



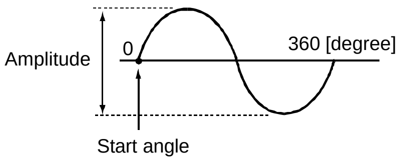


(3) The position follow-up control is executed continuously until the stop command is input.

6.1.16 High-speed oscillation control

1-axis control using OSC (increment method)

(1) The designated axis reciprocates in a sine wave form designated with 1) to 3) below.
Acceleration/deceleration is not carried out.
After starting, the axis will continue repeated reciprocation until stop is input.

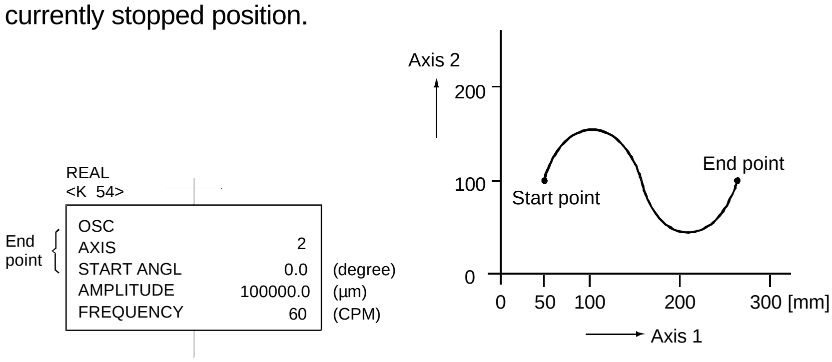


1) Start angle
Designate at which angle of the sine curve the start angle for starting is located.
The setting range is 0 to 359.9 [degree].
Servo program setting error "26" will result if the range is exceeded outside the designated range.

2) Amplitude
Designate the amplitude for reciprocation as a set unit.
Set the amplitude in the range of 1 to 2147483647.
Servo program setting error "25" will result if the range is exceeded outside the designated range.

3) Frequency
Set how many cycles to move the sine curve in one minute.
The setting range is 1 to 5000 [CPM].
Servo program setting error "27" will result if the range is exceeded outside the designated range.

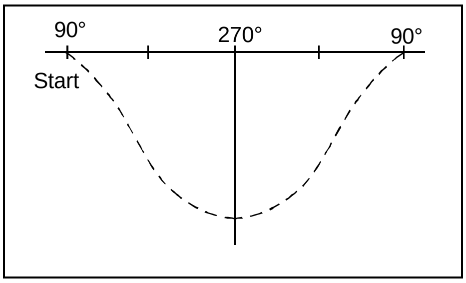
(2) The axis reciprocates the amplitude amount with incremental movement from the currently stopped position.

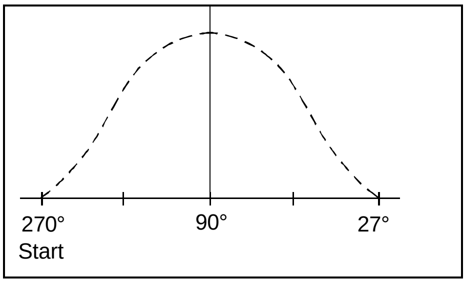


* This example shows axis 1 with linear control to show the state of the axis 2 high-speed oscillation moving with a sine waveform.

(3) As sudden starting is not possible because acceleration/deceleration is not carried out, set the start angle as 90 or 270.

Start angle 90° Start angle 270°


90°
270°
90°
Start


270°
90°
27°
Start

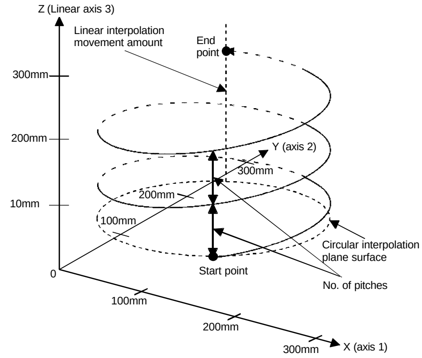
6.1.17 Helical interpolation control with auxiliary point designated

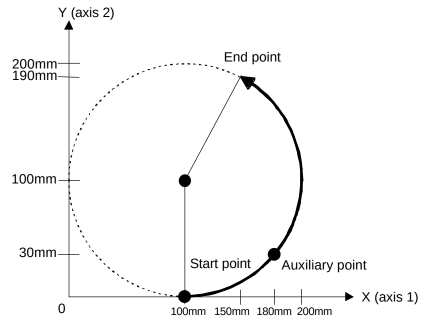
3-axis control by ABH (absolute method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address (address before positioning) based on the zero point, through the designated auxiliary point address on the arc end point address and linear axis end point address.

REAL
<K 70>

	ABH			
Circular interpolation end point	AXIS	1,	150000.0	(µm)
	AXIS	2,	190000.0	(µm)
Linear axis end point	LINEAR AXIS	3,	250000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	AUXILIARY P.	1,	180000.0	(µm)
	AUXILIARY P.	2,	30000.0	(µm)





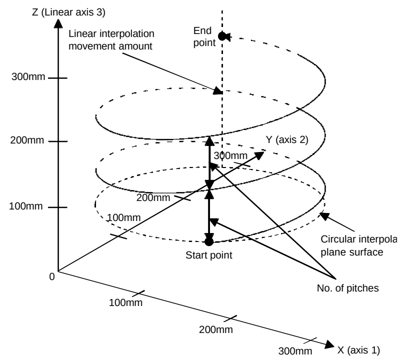
Circular interpolation plane surface

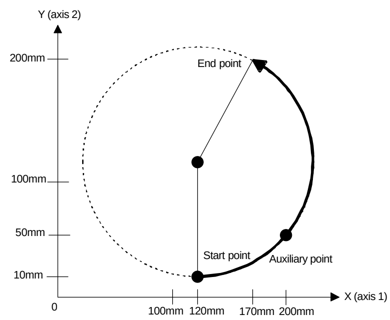
3-axis control by INH (increment method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address through the designated auxiliary point address on the arc end point address and linear axis end point address.

REAL
<K 71>

	INH			
Circular interpolation movement amount	AXIS	1,	50000.0	(µm)
	AXIS	2,	190000.0	(µm)
Linear axis movement amount	LINEAR AXIS	3,	250000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	AUXILIARY P.	1,	80000.0	(µm)
	AUXILIARY P.	2,	40000.0	(µm)





Circular interpolation plane surface

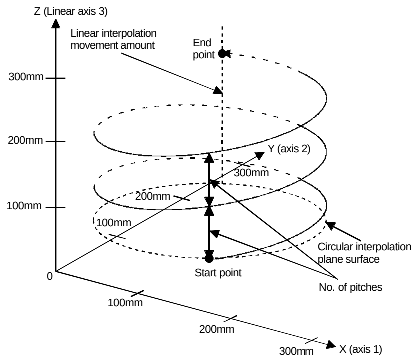
6.1.18 Helical interpolation control with radius designated

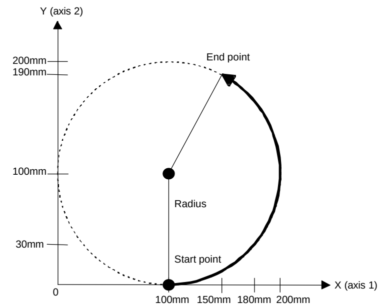
3-axis control by ABH /ABH /ABH /ABH (absolute method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address (address before positioning) based on the zero point, at the designated radius to the arc end point address and linear axis end point address.

REAL
<K 72>

	ABH			
Circular interpolation end point	AXIS	1,	150000.0	(µm)
	AXIS	2,	190000.0	(µm)
Linear axis end point	LINEAR AXIS	3,	250000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	RADIUS		100000.0	(µm)





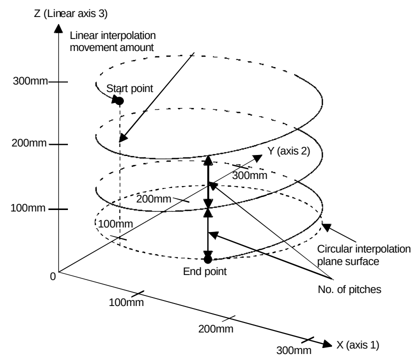
Circular interpolation plane surface

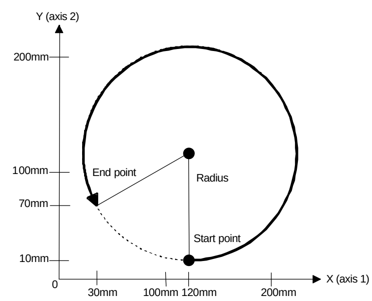
3-axis control by INH /INH /INH /INH (increment method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address as the starting point [0, 0] to the arc end point address designated with the radius and to the linear axis end point address.

REAL
<K 73>

	INH			
Circular interpolation movement amount	AXIS	1,	–90000.0	(µm)
	AXIS	2,	60000.0	(µm)
Linear axis movement amount	LINEAR AXIS	3,	280000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	RADIUS		100000.0	(µm)





Circular interpolation plane surface

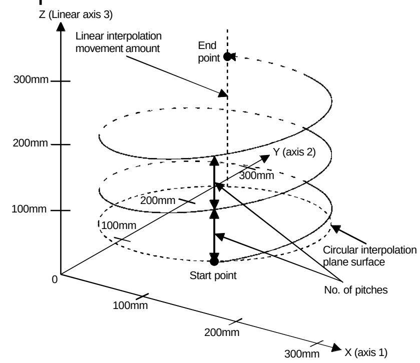
6.1.19 Helical interpolation control with center point designated

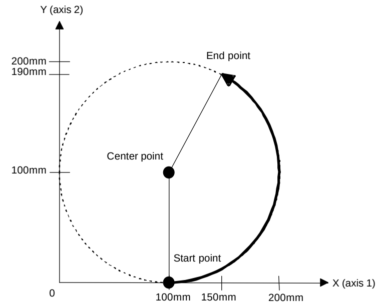
3-axis control by ABH /ABH (absolute method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address (address before positioning) with an arc, having a radius to equal to the distance to the center point. Interpolation is executed to the arc end point address and linear axis end point address.

REAL
<K 74>

	ABH			
Circular interpolation end point	AXIS	1,	150000.0	(µm)
	AXIS	2,	190000.0	(µm)
Linear axis end point	LINEAR AXIS	3,	250000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	CENTER	1,	100000.0	(µm)
	CENTER	2,	100000.0	(µm)





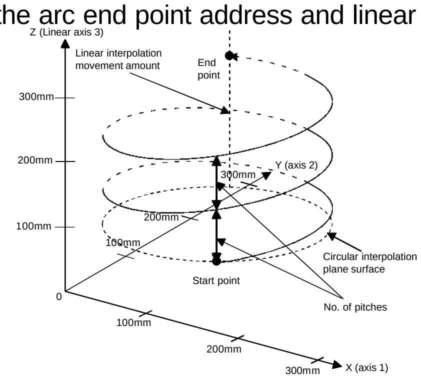
Circular interpolation plane surface

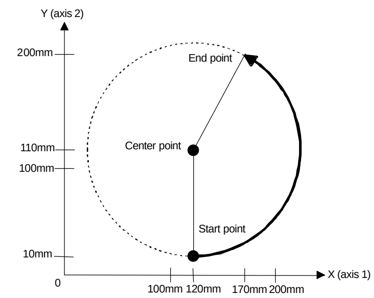
3-axis control by INH /INH (increment method)

(1) Helical interpolation control is realized by linearly interpolating the linear axis while executing circular interpolation from the current stop address as the starting point [0, 0]. Interpolation is executed with an arc, having a radius equal to the distance to the center point. Interpolation ends at the arc end point address and linear axis end point.

REAL
<K 75>

	INH			
Circular interpolation movement amount	AXIS	1,	50000.0	(µm)
	AXIS	2,	190000.0	(µm)
Linear axis movement amount	LINEAR AXIS	3,	250000.0	(µm)
	SPEED		5000.00	(mm/min)
	NUMBER OF PITCHES		2	
	CENTER	1,	110000.0	(µm)
	CENTER	2,	120000.0	(µm)



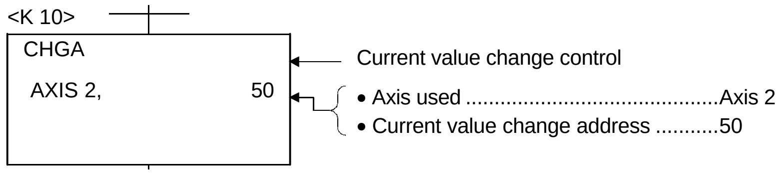


Circular interpolation plane surface

6.1.20 Current value change

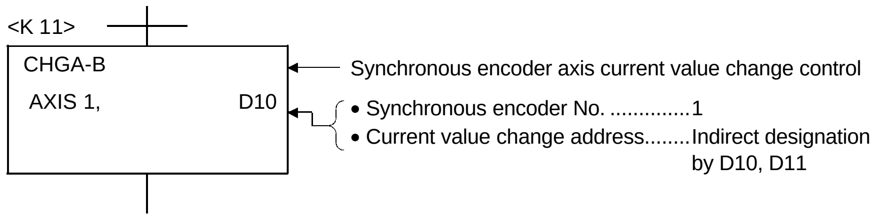
CHGA Servomotor/virtual servomotor axis current value change control

(1) When the real mode is selected, the current value of the designated axis is changed.
(2) When the virtual mode is selected, the current value of the designated virtual servomotor is changed.



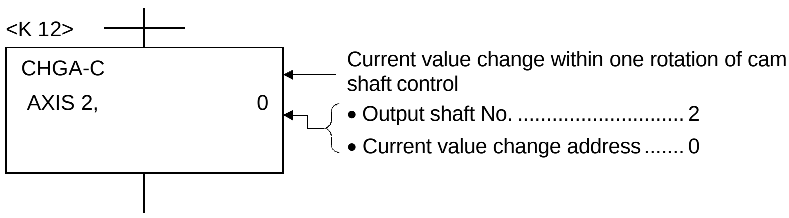
CHGA-E Synchronous encoder axis current value change control

(1) The current value of the designated synchronous encoder axis is changed to the designated address.



CHGA-C Current value change within one rotation of cam shaft control

(1) When the CHGA-C command is executed, the current value within one rotation of the designated cam shaft is changed to the designated address.
(2) In this case, the cam shaft may be rotating.



Memo

Chapter 7 Operation Control Program

A substitute operational expression, dedicated motion function and bit device control command can be set with the operation control program.
Multiple blocks can be set in one operation control program, however, the shifting condition can be set only to the transition program.
The operational expressions that can be described by the operation control program and transition program are described in this section.

7.1 Order of operator and function priority

The order of the operator and function priority is shown below.
The operation order can be designated randomly by using parentheses.

Priority order	Items (operator, function)
	Calculation within parentheses ((...))
High	Standard function (SIN, COS, etc.), type conversion (USHORT, LONG, etc.)
	Bit inversion (¯), logical negation (!), sign inversion (–)
	Multiplication (*), division (/), remainder (%)
	Addition (+), subtraction (–)
	Left bit shift (<<), right bit shift (>>)
	Comparator: Below (<), less than: (<=), over: (>), more than (>=)
	Comparator: Equal (=), Not equal (!=)
	Bit logical product (&)
	Bit exclusive logical sum (^)
	Bit logical sum (｜)
	Logical product (*)
	Logical sum (+)
Low	Substitute (=)

7.2 List of operation control and transition commands

Division	Symbol	Function	Format	Number of basic steps	Usable program		Usable expression			
					F/FS	G	Calculation expression	Bit conditional expression	Relation condition expression	Y/N transition conditional expression
Dyadic operation	=	Substitute	(D) = (S)	4	○	○	○	–	–	–
	+	Addition	(S1) + (S2)	4	○	○	○	–	–	–
	–	Subtraction	(S1) – (S2)	4	○	○	○	–	–	–
	*	Multiplication	(S1) * (S2)	4	○	○	○	–	–	–
	/	Division	(S1) / (S2)	4	○	○	○	–	–	–
	%	Remainder	(S1) % (S2)	4	○	○	○	–	–	–
Bit operation	¯	Bit inversion (complement)	¯ (S)	2	○	○	○	–	–	–
	&	Bit logical product	(S1) & (S2)	4	○	○	○	–	–	–
	|	Bit logical sum	(S1) | (S2)	4	○	○	○	–	–	–
	^	Bit exclusive logical sum	(S1) ^ (S2)	4	○	○	○	–	–	–
	>>	Bit right shift	(S1) >> (S2)	4	○	○	○	–	–	–
	<<	Bit left shift	(S1) << (S2)	4	○	○	○	–	–	–
Code	–	Code inversion (complement of 2)	– (S)	4	○	○	○	–	–	–
Standard function	SIN	Sine	SIN (S)	2	○	○	○	–	–	–
	COS	Cosine	COS (S)	2	○	○	○	–	–	–
	TAN	Tangent	TAN (S)	2	○	○	○	–	–	–
	ASIN	Inverse sine	ASIN (S)	2	○	○	○	–	–	–
	ACOS	Inverse cosine	ACOS (S)	2	○	○	○	–	–	–
	ATAN	Inverse tangent	ATAN (S)	2	○	○	○	–	–	–
	SQRT	Square root	SQRT (S)	2	○	○	○	–	–	–
	LN	Natural logarithm	LN (S)	2	○	○	○	–	–	–
	EXP	Exponential operation	EXP (S)	2	○	○	○	–	–	–
	ABS	Absolute value	ABS (S)	2	○	○	○	–	–	–
	RND	Round off	RND (S)	2	○	○	○	–	–	–
	FIX	Round down	FIX (S)	2	○	○	○	–	–	–
	FUP	Round up	FUP (S)	2	○	○	○	–	–	–
	BIN	BCD to BIN conversion	BIN (S)	2	○	○	○	–	–	–
	BCD	BIN to BCD conversion	BCD (S)	2	○	○	○	–	–	–
Type conversion	SHORT	Conversion to 16-bit integer type (with code)	SHORT (S)	2	○	○	○	–	–	–
	USHORT	Conversion to 16-bit integer type (without code)	USHORT (S)	2	○	○	○	–	–	–
	LONG	Conversion to 32-bit integer type (with code)	LONG (S)	2	○	○	○	–	–	–
	ULONG	Conversion to 32-bit integer type (without code)	ULONG (S)	2	○	○	○	–	–	–
	FLOAT	Conversion 64-bit floating decimal type with data regarded as "Data with code"	FLOAT (S)	2	○	○	○	–	–	–
	UFLOAT	Conversion of 64-bit floating decimal type with data regarded as "Data without code"	UFLOAT (S)	2	○	○	○	–	–	–
Bit device status	(None)	ON (A contact)	(Bit conditional expression)	2	○	○	–	○	–	○
	!	OFF (B contact)	! (Bit conditional expression)	2	○	○	–	○	–	○

Division	Symbol	Function	Format	Number of basic steps	Usable program F/FS	Usable program G	Usable expression: Calculation expression	Usable expression: Bit conditional expression	Usable expression: Relation condition expression	Usable expression: Y/N transition conditional expression
Bit device control	SET	Device setting	SET (D)	3	○	○	–	○	–	–
			SET (D) = (Conditional expression)	4	○	○	–	○	○	–
	RST	Device resetting	RST (D)	3	○	○	–	○	–	–
			SET (D) = (Conditional expression)	4	○	○	–	○	○	–
	DOUT	Device output	DOUT (D), (S)	4	○	○	–	○	–	–
	DIN	Device input	DIN (D), (S)	4	○	○	–	○	–	–
	OUT	Bit device output	OUT (D) = (Conditional expression)	4	○	○	–	○	○	–
Logical operation	(None)	Logical affirmation	(Conditional expression)	0	○	○	–	○	○	○
	!	Logical negation	! (Conditional expression)	2	○	○	–	○	○	○
	*	Logical product	(Conditional expression) * (Conditional expression)	4	○	○	–	○	○	○
	+	Logical sum	(Conditional expression) + (Conditional expression)	4	○	○	–	○	○	○
Com-parison operation	==	Equal	(Calculation expression) == (Calculation expression)	4	○	○	–	–	○	○
	!=	Not equal	(Calculation expression) != (Calculation expression)	4	○	○	–	–	○	○
	<	Below	(Calculation expression) < (Calculation expression)	4	○	○	–	–	○	○
	<=	Less than	(Calculation expression) <= (Calculation expression)	4	○	○	–	–	○	○
	>	Over	(Calculation expression) > (Calculation expression)	4	○	○	–	–	○	○
	>=	More than	(Calculation expression) >= (Calculation expression)	4	○	○	–	–	○	○
Dedicated motion function	CHGV	Speed change request	CHGV ((S1), (S2))	4	○	○	–	–	–	–
	CHGT	Torque limit value change request	CHGT ((S1), (S2))	4	○	○	–	–	–	–
Others	EI	Event task permit	EI	1	○	○	–	–	–	–
	DI	Event task prohibit	DI	1	○	○	–	–	–	–
	NOP	Non-processing	NOP	1	○	○	–	–	–	–
	BMOV	Block transfer	BMOV (D), (S), (n)	7	○	○	–	–	–	–
	TIME	Time waiting	TIME (S)	7	–	○	–	–	–	–
	MULTW	Writing of data to shared memory of local machine	MULTW (D), (S), (n), (D1)	8	○	○	–	–	–	–
	MULTR	Reading of data from shared memory of other machine	MULTR (D), (S1), (S2), (n)	7	○	○	–	–	–	–
	TO	Writing of word data to intelligent function unit/special unit	TO (D1), (D2), (S), (n)	7	○	○	–	–	–	–
	FROM	Reading of word data from intelligent func-tion unit/special unit	FROM (D), (S1), (S2), (n)	7	○	○	–	–	–	–

Expression for approximating size of one program code in operation control program and transition program

> 2 + (1 + Total number of basic steps in one block)
> + Number of 32-bit constants/block × 1
> + Number of 64-bit constants/block × 3) × Number of blocks (steps)
> (1 step = 2 bytes)

POINT

- The shifting conditions must always be set in the final block of the transition program.

7.3 Dedicated motion functions (CHGV, CHGT)

Speed change request: CHGV

Format	Setting data	Contents	Result data type
CHGV ((S1), (S2))	(S1)	Axis No. requesting for speed change	–
	(S2)	Designated speed	

(1) The speed is changed with the following procedure.
   1) The "speed change flag" (M2061 to M2092) corresponding to the axis designated by (S1) is turned ON.
   2) The speed of axis designated by (S1) is changed to the speed designated by (S2).
   3) The "Speed change flag" is turned OFF.

(2) The range of axis numbers set in (S1) is shown below.

Q172CPU	Q173CPU
1 to 8	1 to 32

(3) The operation is performed as follows depending on the code of the designated speed set in (S2).

Code of designated speed	Operation
Positive	Change of speed
0	Temporary stop
Negative	Backward

(4) The range of designated speed set in (S2) is shown below.
   1) Real mode

	mm		inch		degree		PULSE	
	Setting range	Units	Setting range	Units	Setting range	Units	Setting range	Units
Speed change request	0 to 600000000	$\times 10^{-2}$ mm/min	0 to 600000000	$\times 10^{-3}$ inch/min	0 to 2147483647	$\times 10^{-3}$ degree/min	0 to 10000000	PLS/sec
Backward request	–1 to –600000000	$\times 10^{-2}$ mm/min	–1 to –600000000	$\times 10^{-3}$ inch/min	–1 to –2147483647	$\times 10^{-3}$ degree/min	–1 to –10000000	PLS/sec

   2) Virtual mode

	PULSE	
	Setting range	Units
Speed change request	0 to 10000000	PLS/sec
Backward request	–1 to –10000000	PLS/sec

(5) When the speed change is requested with a negative speed during start, deceleration can be started at that point, and the axis returned in the reverse direction upon completion of deceleration.

Program example

Program to change positioning speed of axis 2

```
CHGV (K2, K10)
```

Reverse program to change positioning speed of axis 1 to a negative value

```
CHGV (K1, K-1000)
```

Torque limit value change request: CHGT

Format	Setting data	Contents	Result data type
CHGT ((S1), (S2))	(S1)	Axis No. to request for torque limit value change	–
	(S2)	Designated torque limit value	

(1) The torque limit value of the axis designated by (S1) is changed to the torque limit value designated by (S2).

(2) If the servo has been started for the axis in the real mode, the torque limit value can be changed regardless of whether operation has been started or stopped, or whether the servo is ON or OFF.

(3) The range of axis numbers set in (S1) is shown below.

Q172CPU	Q173CPU
1 to 8	1 to 32

(4) The range of torque limit value that can be set in (S2) is 1 to 500 [%].

(5) The relation of the torque limit value and torque limit value designated by the servo program is shown below.

At start

The torque limit value is commanded to the servo of the start axis according to the "P. torque" set with the servo program or the "torque limit value" in the designated parameter block.
When interpolation is started, the torque value is limited for the interpolation axis.

↓

When the CHGT command is executed, the setting torque limit value is commanded only to the designated axis.

↓

The torque limit value commanded to the servo when starting the servo program or when starting JOG is valid thereafter only when it is less than the torque limit value changed by CHGT.
This torque limit value is clamped for each axis.

During start

1) Even when the following setting is made, the torque limit value is not changed to the torque limit value higher than that changed by the CHGT command.
   - Torque limit value at halfway point during constant speed control or speed change control
   - Torque limit value when speed/position changeover control is changed to position control
   - Torque limit value in speed control II
2) It is possible for CHGT command to change the torque limit value to a value higher than the torque limit value set by the servo program or parameter block.

Program example

Program to change torque limit value of axis 2

```
CHGT (K2, K10)
```

7.4 Other commands

Event task permit: EI

Format
EI

(1) Execution of event task is permitted.

(2) It is applicable only to the normal task.

Program example

Execution of event task is permitted.

```
EI
```

Event task prohibit: DI

Format
DI

(1) Execution of event task is prohibited.

(2) When an external interrupt or PLC interrupt occurs after execution of the DI command, the corresponding event task is executed once when the EI command is executed.

(3) The constant cycle event task is not executed during DI.

(4) The execution of NMI task cannot be prohibited.

Program example

Program to prohibit the execution of event task

```
DI
```

Non-processing: NOP

Format
NOP

(1) Since the command is a non-processing command, and will not bring about any influence upon last operation.

Block transfer: BMOV

Format	Setting data	Contents	Result data type
BMOV (D), (S), (n)	(D)	Head No. of transfer destination device	–
	(S)	Head No. of transfer source device	
	(n)	Number of words transferred	

(1) The contents of n-words from the word device designated by (S) are transferred in a batch to the n-words from the word device designated by (D).

(2) The data can be transferred even when the transfer source device is overlapped with the transfer destination device.

(3) When the Nn (cam No.) is designated in (D) and (S), the cam data can be transferred in a batch.

Program example

Program to transfer the contents of 5 words (from D0) to 5 words (from #10) in a batch

```
BMOV #10, D0, K5
```

#10	12	Batch transfer	D0	12
#11	34	←	D1	34
#12	56		D2	56
#13	78		D3	78
#14	90		D4	90

Program to transfer the contents of 2,048 words (from #0) to the data area of cam No.2 (resolution: 2048) in a batch

```
BMOV N2, #0, K2048
```

	Data of cam No.2			
0th stroke ratio	H0000	Batch transfer	#0	H0000
1st stroke ratio	H0005	←	#1	H0005
2nd stroke ratio	H000A		#2	H000A
:	:		:	:
2047th stroke ratio	H0000		#2047	H0000

Time waiting: TIME

Format	Setting data	Contents	Result data type
TIME (S)	(S)	Waiting time (0 to 2147483647) msec	Logical type (True/False)

(1) The system will wait for time designated by (S).
When the time elapsed is less than the setting time, the state becomes "False". When it is over the setting time, it becomes "True".

Program example

Program to wait for 60 sec. (when constant is designated)

```
TIME K60000
```

Comment: / /

Format
/ /

(1) The character string after // to the end of the block is a comment.

(2) Double-byte characters can be used.

Program example

Example to attach a comment to the substitute program

```
Substitute the value D0=D1//D0 (16-bit integer data) for "D1".
```

Writing of data to local machines shared memory: MULTW

Format	Setting data	Contents	Result data type
MULTW (D), (S), (n), (D1)	(D)	Shared memory address of local machine CPU at writing destination	–
	(S)	Head No. of device in which writing data is stored	
	(n)	Number of data items to be written	
	(D1)	Local machine device to turn ON after completion of writing	

(1) (n) word of device data is written from the local machine CPU's (S) to the CPU shared memory address designated local machine CPU's CPU.
When the writing is completed, the completion bit designated by (D1) is turned ON.

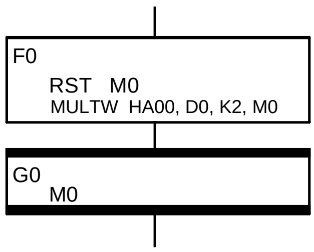
(2) Reset the completion bit using the user program.

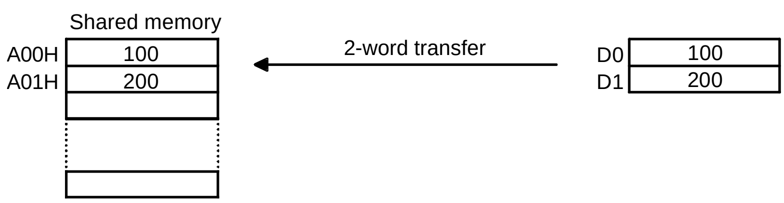
(3) Note that the other MULTW command cannot be executed until the MULTW command is executed and the completion bit is turned ON.

(4) Since the processing time increases in proportion to the number of writing data items (n), it is recommended to execute the processing to the normal task or event task (7.1ms or over), so as not to obstruct the motion operation.

Program example

Program to write 2-word from D0 to shared memory A00H, and shift to succeeding step after confirming the writing completion.





Reading of data from shared memory of other machine: MULTR

Format	Setting data	Contents	Result data type
MULTR (D), (S1), (S2), (n)	(D)	Head No. of device in which data read is stored	–
	(S1)	Head input No. of Q-CPU, motion-CPU to read	
	(S2)	Head address of shared memory of data to read	
	(n)	Number of reading data items	

(1) The (n) word of data is read from the address designated by (S2) of the CPU’s shared memory of other machine CPU designated by (S1) to store it in and after the device designated by (D).

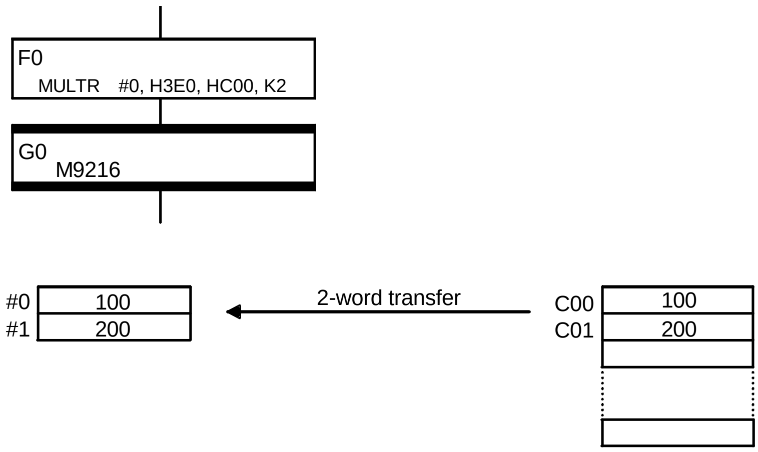
(2) When the reading of data from designated object machine is completed properly, the "Reading completed" flag corresponding to the object machine is turned ON.

(3) When the MULTR command is executed last to the same machine, the "Reading completed" flag of the object machine is turned ON/OFF depending on the results of MULTR executed last.

(4) Since the processing time increases in proportion to the number of reading data items (n), it is recommended to execute the processing to the normal task or event task (7.1ms or over), so as not to obstruct the motion operation.

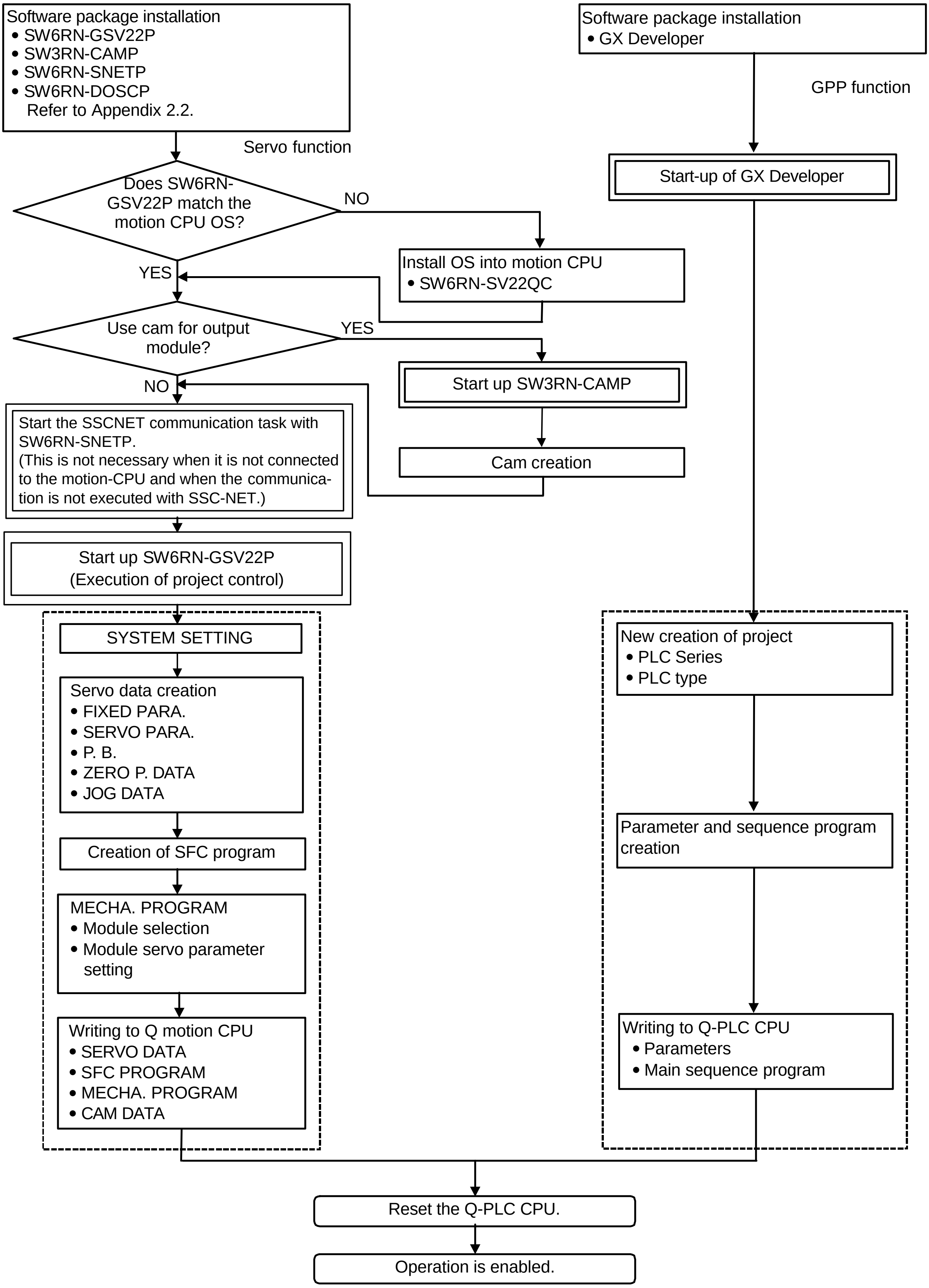
Program example

Program to shift to succeeding step after reading 2-word from the No.1 machine’s shared memory C00H to #0



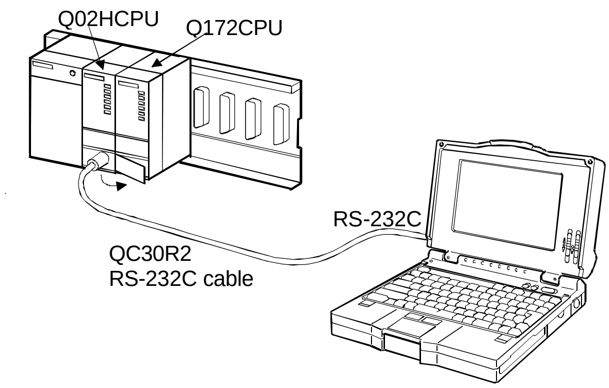
Chapter 8 Windows Personal Computer Operations

8.1 Flow of creating data for operating motion controller



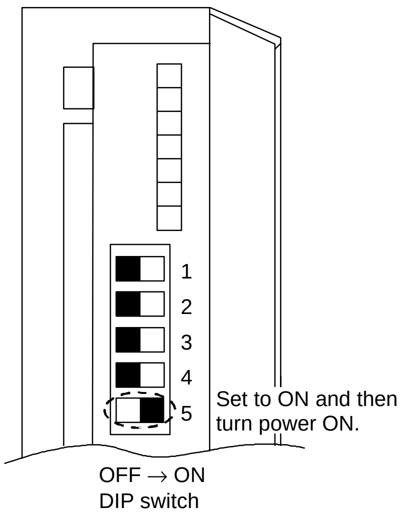
8.2 Registering the main unit OS

Register (install) the OS (SW6RN-SV22QC) for the Q motion CPU.



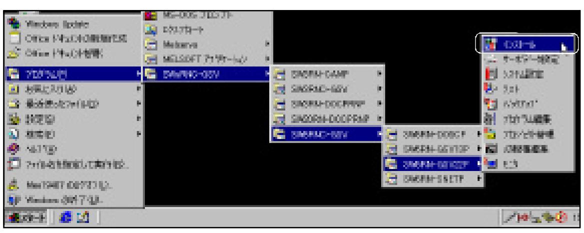
(1) Connect the RS-232C port of DOS/V personal computer to the RS-232C connector of Q02HCPU using QC30R2 cable, and turn ON the power supply. (If the system has already been started with the cable connected, start from step (2).)

(2) Turn the Q motion CPU power switch OFF, set the install switch to Installation enabled (ON), and then turn the power switch ON.



DIP switch 5	
OFF	ON
Installation disabled	Installation enabled

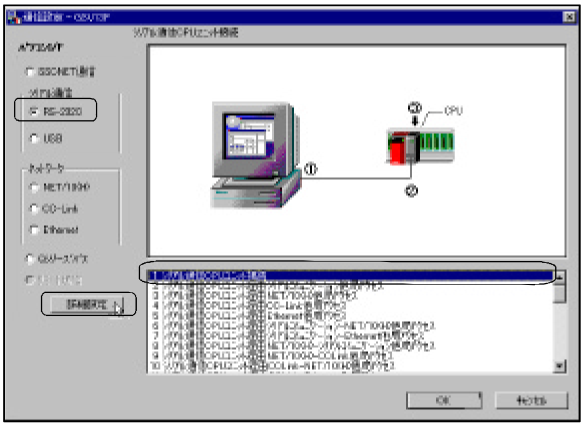
(3) Click on [Start], [Program], [SWnRNC-GSV], [SW6RNC-GSV], [SW6RN-GSV22P] and then [Install].



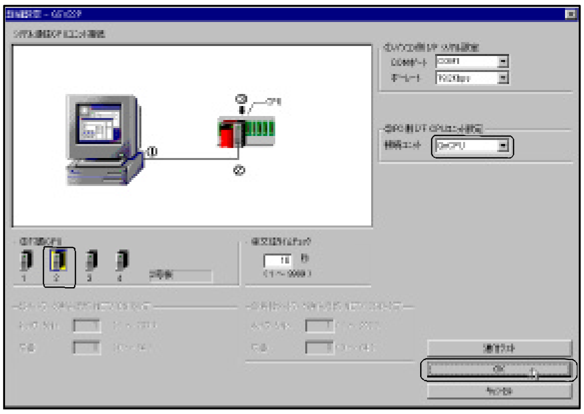
Continued on next page

Continued from previous page

(4) The INSTALL dialog box will open. Click on [Communication], and then the [Communication setting] menu.



(5) The COMMUNICATION SETTING dialog box will open. Check "RS-232C", select "1". Serial communication CPU connection", and click on the Detail setting button.

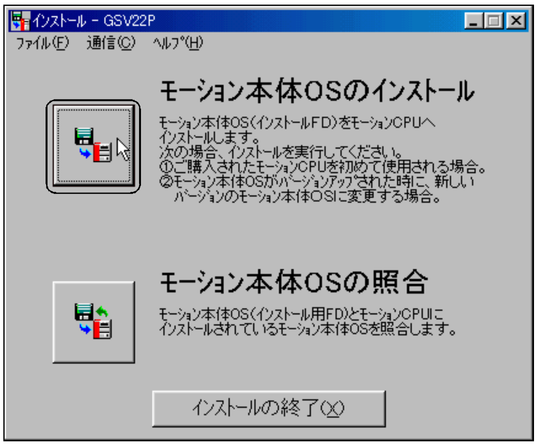


(6) The DETAIL setting dialog box will open. Select the "PC side I/F CPU setting" to 'QnCPU', and the "Object CPU" to '#2 machine'.
After setting, click on the OK button.
The Communication setting dialog box will open again, so click on the OK button.

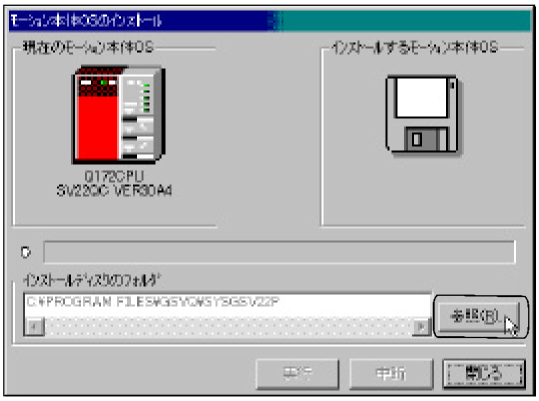
(7) When the message shown at the left appears, click on the Yes button.

Continued on next page

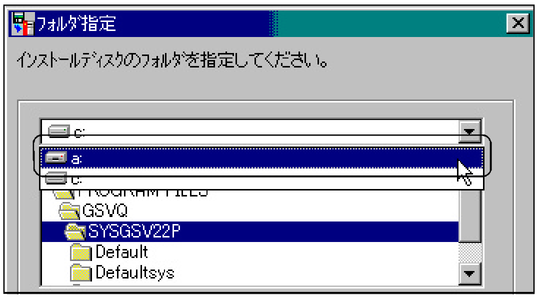
Continued from previous page



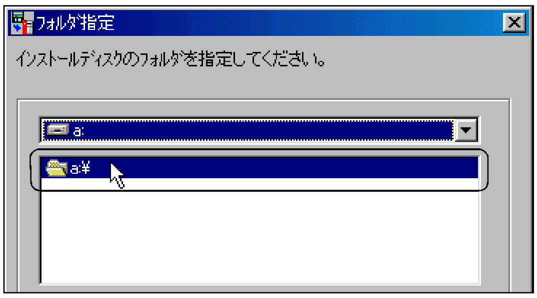
(8) The INSTALL dialog box will open. Click on the Motion main unit OS install button.



(9) The MOTION MAIN UNIT OS INSTALL dialog box will open. Click on the Reference button.



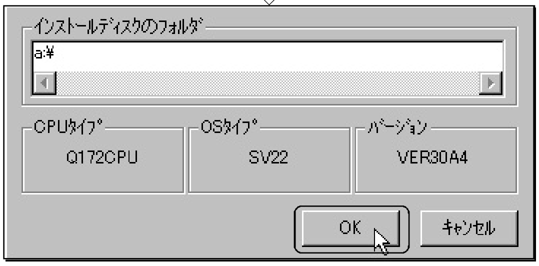
(10) Insert the OS FD (SW6RN-SV2QC-1/2) in the FD drive, and select 'a:' in the folder's designation dialog box.



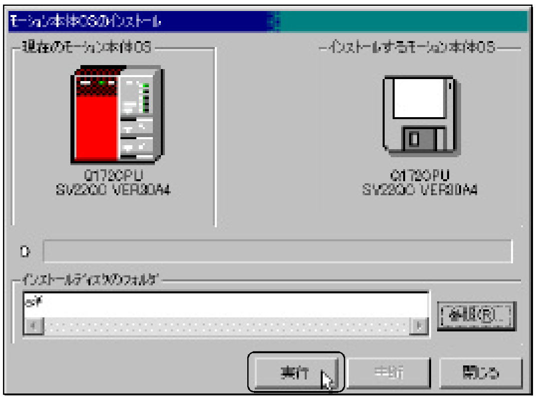
(11) Double-click on 'a:\'.

Continued on next page

Continued from previous page



(12) When 'SV22' appears at "OS type", click on the OK button.



(13) Click on the Execute button in the MOTION MAIN UNIT OS INSTALL dialog box.

(14) When the message shown at left appears, replace the FD with the second OS FD (SW6RN-SV2QC-2/2), and click on the OK button.

Note: It takes several minutes to install the main unit OS.

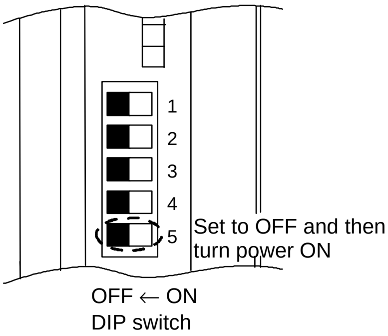
(15) When the message "Installation completed" appears, click the OK button.

(16) Click on the CLOSE button in MOTION MAIN UNIT OS INSTALL dialog box.

Continued on next page

Continued from previous page

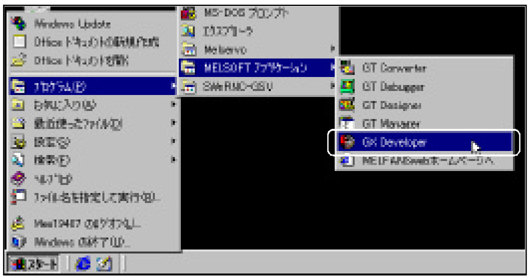
(17) Click on the Install End button.



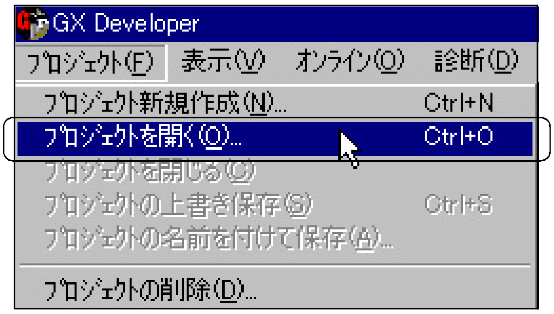
(18) Turn the Q motion CPU power OFF, set the install switch to Installation disabled, and then turn the power ON again. This completes registration of the main unit OS.

8.3 Setting the Q-PLC CPU

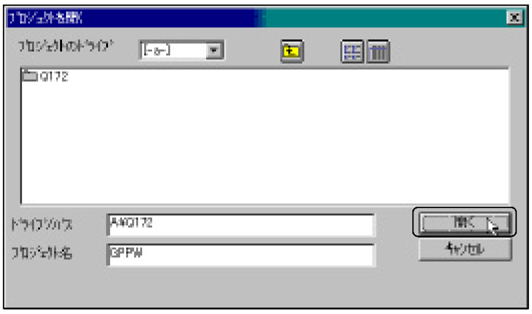
8.3.1 Reading the sequence program



(1) Click on [Start], [Program], [MELSOFT application] and then [GX Developer] in Windows.

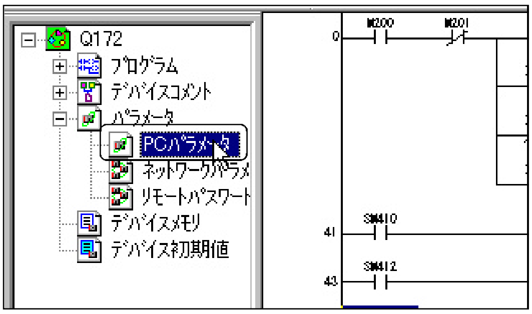


(2) The GX Developer will start, so click on [Project] and then the [Open project] menu.

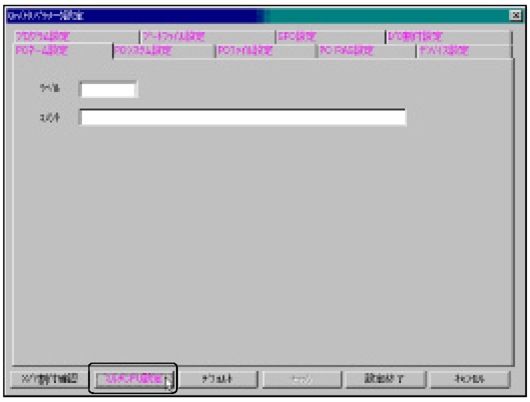


(3) Input 'A:\Q172\' in the "Drive path" in Open project dialog box, and 'GPPW' in "Project name" then click on the Open button.
The sequence program, PC parameter, etc., will be read in from the FD.

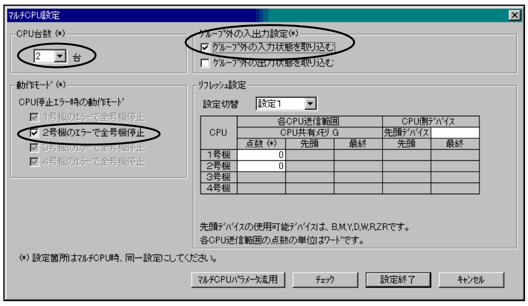
8.3.2 Setting the multi-CPU



(1) Double-click on [Parameter] and then [PC parameter] in project data list.



(2) The Qn(H) PARAMETER SETTING dialog box will open, so click on the Multi-CPU setting button.



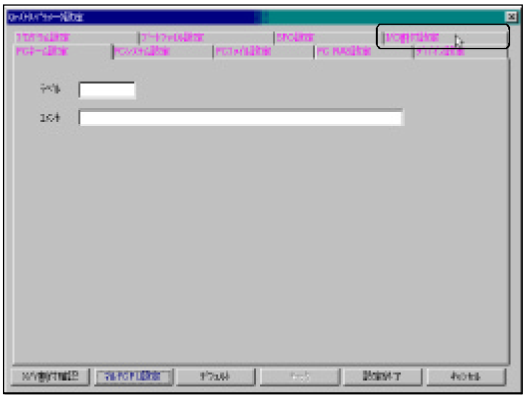
(3) The MULTI-CPU SETTING dialog box will open, set the "Number of CPUs" to '2 units', and check the "Retrieve out-of-group input status".
Confirm that 'All machines stopped due to error in #2 machine' of "Operation mode" is checked.

(4) Select the "Setting changeover" in "Refresh setting" to 'Setting 1', and set the following.
"Head device" : 'M0'
"Number of #1 machine points" : '8'
"Number of #2 machine points" : '8'

Continued on next page

Continued from previous page

(5) Select the "Setting changeover" in "Refresh setting" to 'Setting 2', and set the following.
"Head device" : 'D6000'
"Number of #1 machine points" : '4'
"Number of #2 machine points" : '4'
Click the Setting End button after setting.



(6) The Qn(H) PARAMETER SETTING dialog box will open again, so click on the I/O ASSIGNMENT SETTING tab.

(7) Click on the Detail setting button in the I/O ASSIGNMENT setting tab screen.

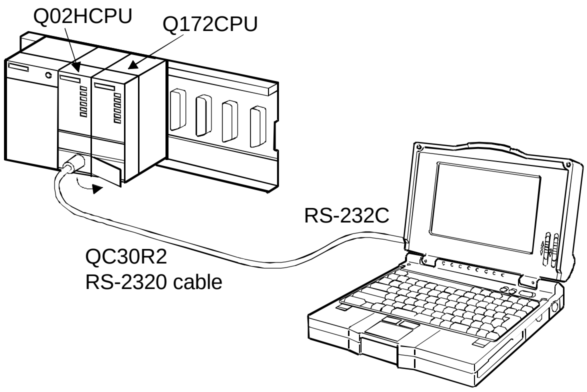
(8) The I/O UNIT, INTELLIGENT FUNCTION UNIT DETAIL SETTING dialog box will open. Select the "Control CPU" for "Slot 1 (*-1)" to '#2 machine', and the "Control CPU" for "Slot 2 (*-2)" to '#1 machine', and click on the Setting End button.

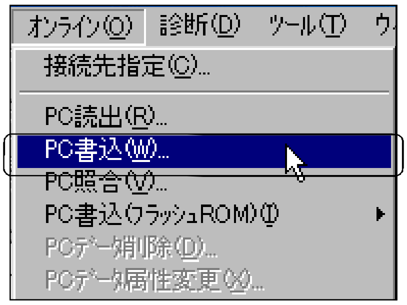
Continued on next page

Continued from previous page

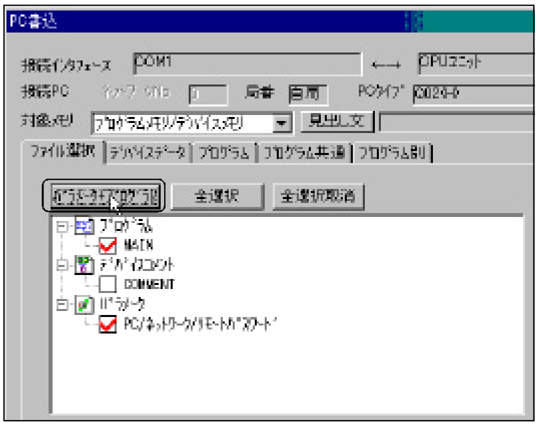
(9) The Qn(H) PARAMETER SETTING dialog box will open again, so click on the $\boxed{\text{Setting End}}$ button.

8.3.3 Writing the sequence program

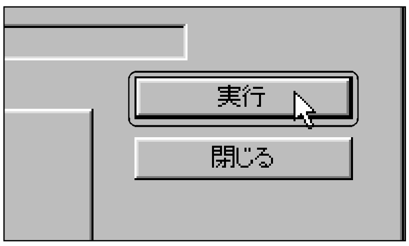




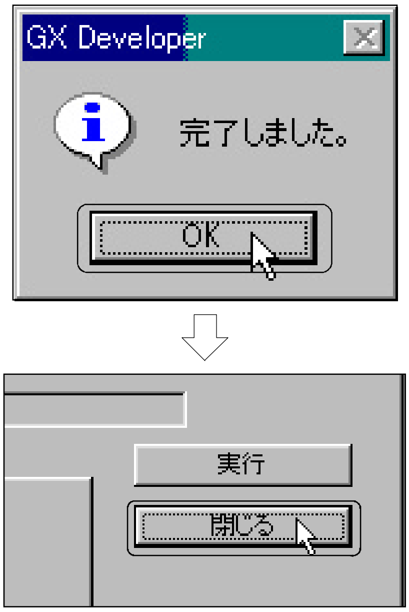
(1) Click on [Online], and then [Write to PC].



(2) The WRITE TO PC dialog box will open, so click on the Parameter + Program button.



(3) Click on the Execute button.

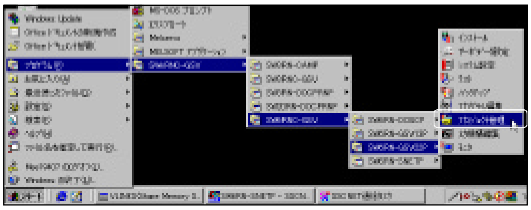


(4) The message "Completed." will appear when writing of data to the PC is completed, so click on the OK button.

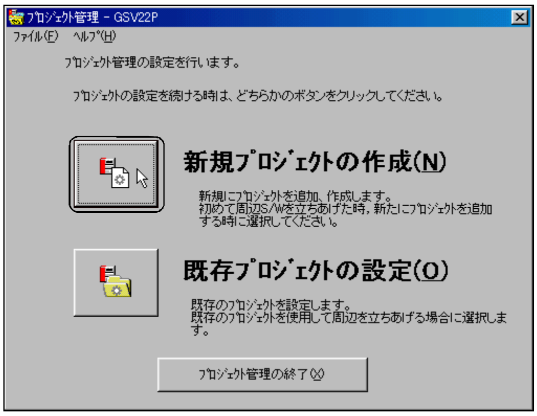
(5) Click on the Close button in the WRITE TO PC dialog box.

8.4 Starting up SW6RN-GSV22P

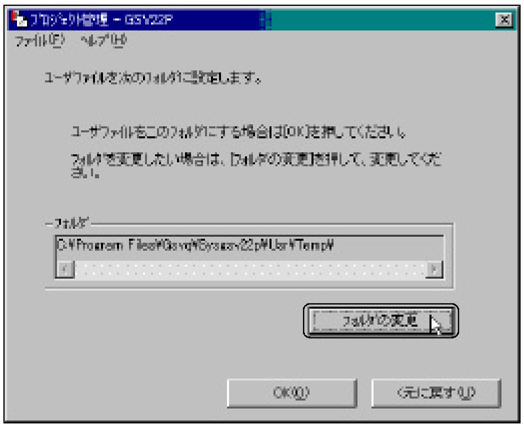
The operations from starting up the SW6RN-GSV22P to the creation of a new project are explained in this section.



(1) Click on [Start], [Program], [SWnRNC-GSV], [SW6RNC-GSV], [SW6RN-GSV22P] and then [Project control].

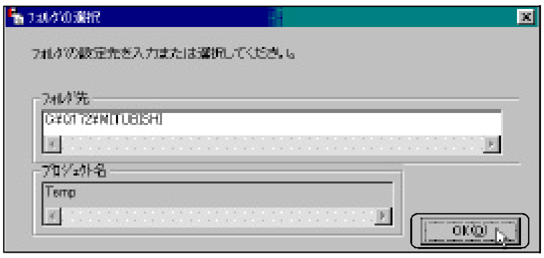


(2) Click on the Creation of new project button in the PROJECT CONTROL dialog box.



(3) After the screen changes, click on the Folder change button.

To return to the last screen, click the Undo button.



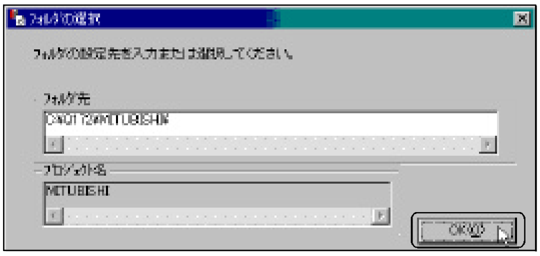
(4) Set 'C:\Q172\ (input of own name in English)' in "Folder destination" in the FOLDER SELECT dialog box, and click on the OK button.

Continued on next page

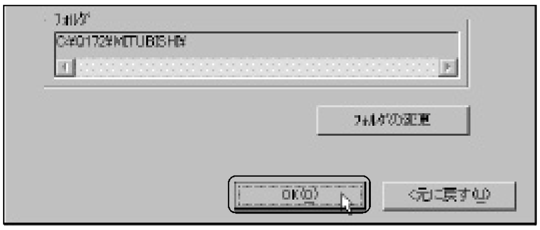
Continued from previous page

(5) When the message "Create a new project?" appears, click on the YES button.

(6) When your own name (English) appears at "Project name" in the FOLDER SELECT dialog box, click on the OK button.



(7) Click on the OK button in PROJECT CONTROL dialog box.



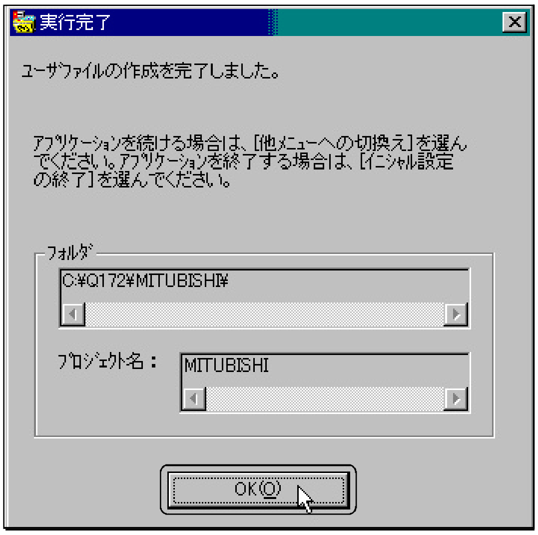
(8) After the screen changes, click on the New creation button.

Continued on next page

Continued from previous page

(9) When the NEW CREATION dialog box appears, select 'Q172' for “CPU select" and 'SW6-SV22QC (SFC)' for "OS select", then click on the OK button.

(10) When dialog box confirming creation of the user initial file appears, click on the YES button.



(11) When the EXECUTION COMPLETE dialog box appears, click on the OK button.
This completes start up and initialization.

(12) To use the other function of SW6RN-GSV22P in succession, Click on the Change to other menu button in the PROJECT CONTROL dialog box to use.

Continued on next page

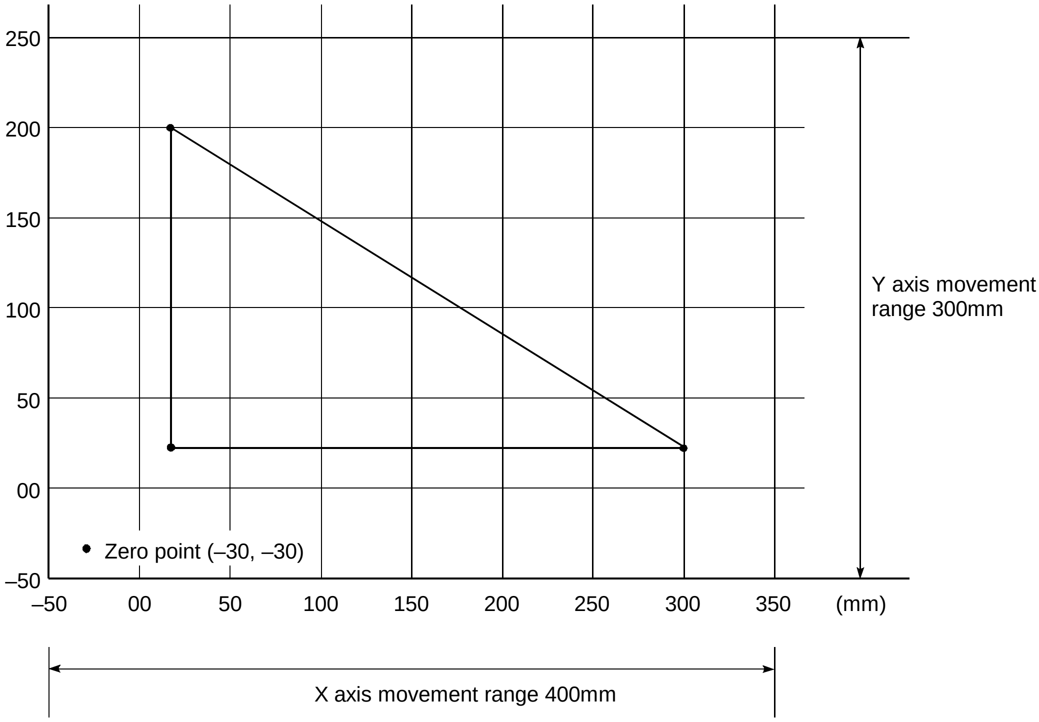
Continued from previous page

(13) The CHANGE TO OTHER MENU dialog box will appear.

Chapter 9 Basic Practice Using the SV22 Real Mode

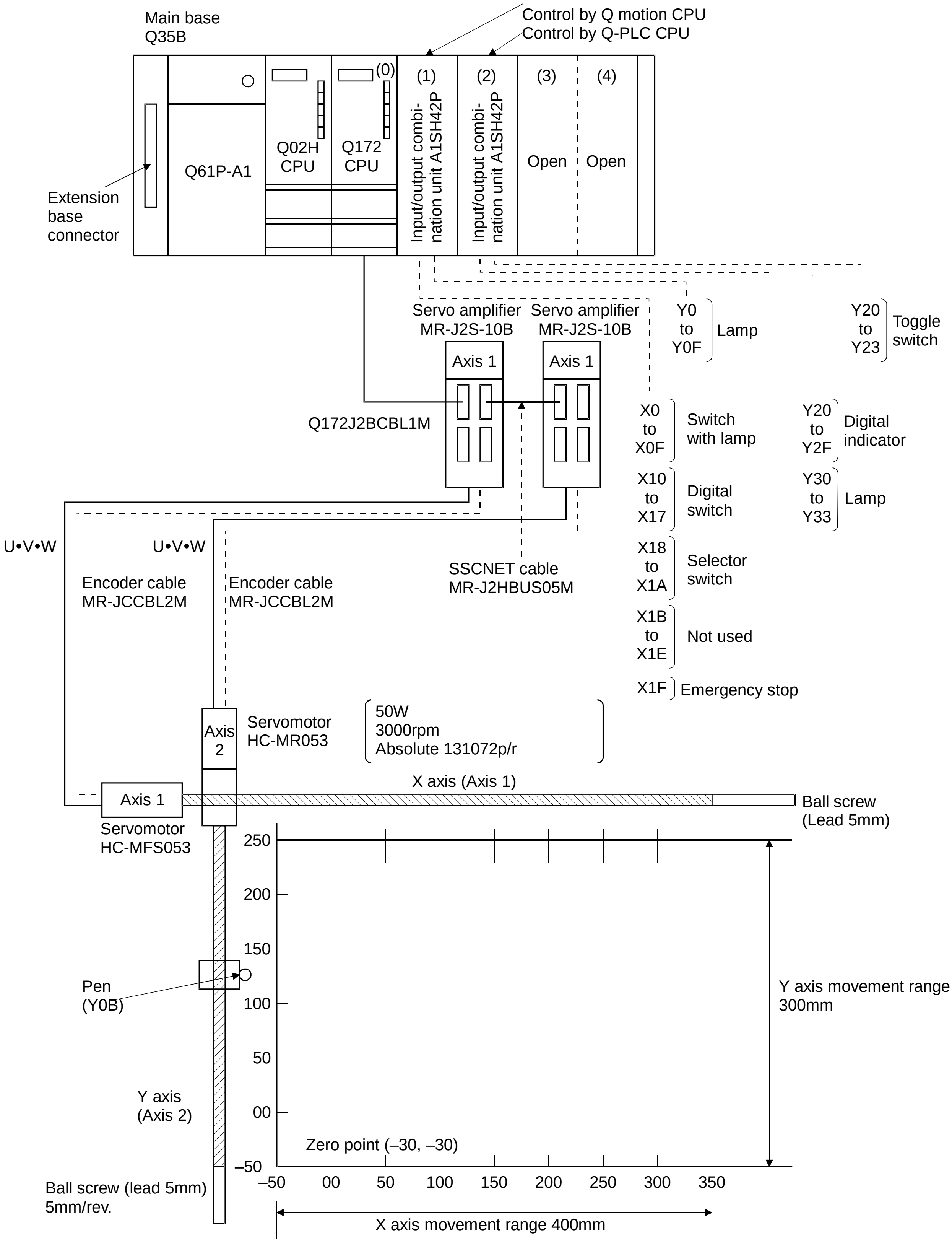
9.1 Details of practice

A triangle will be drawn on the X-Y table as a positioning path.
The SV13 is the same as the SV22's real mode so this practice session will be used for both.

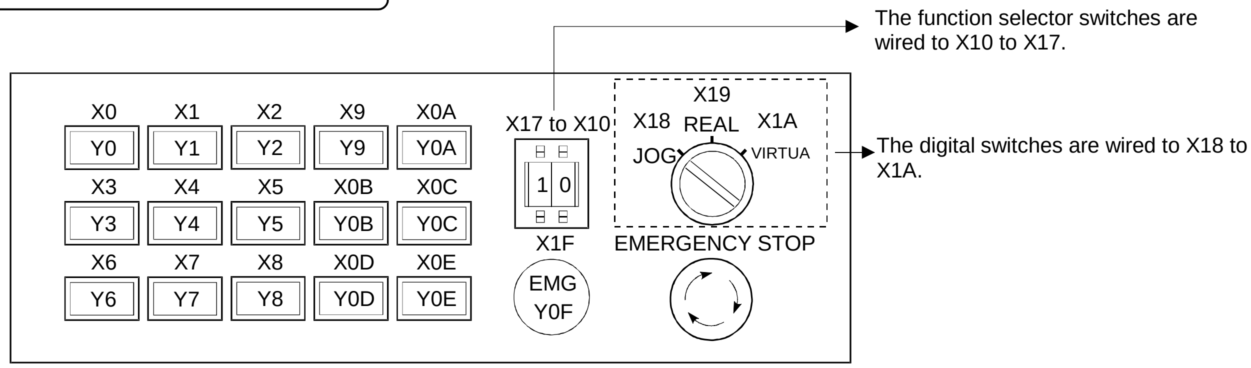


9.2 Q172CPU practice machine system configuration

Since the external signals (limit, DOG) are not used for this practice, the Q1272LX unit is omitted.



Practice machine operation panel



The input signal switches are wired to X0 to X0E.
The lamps are wired to Y0 to Y0E.

The basic settings for the Q motion CPU are set.

X0	X1	X2	X9	X0A
STANDBY POINT POSITION-ING	SET NO POSITION-ING	CONTINUE POSITION-ING		VIRTUAL MODE PRSENT VALUE
X3	**X4**	**X5**	**X0B**	**X0C**
SPEE CHANGE 2000	SPEE CHANGE 1000	SPEE CHANGE 300	PE UP/DOWN	ERROR RESET
X6	**X7**	**X8**	**X0D**	**X0E**
CAM DATA CHANGE REQUEST		CLUTCH OFF	STOP	SUDDEN STOP

X10
to
X17

DIGITAL
switch 2 digit

0	0

X18 ....JOG
X19 ....Real
X1A..............Virtual

Operation
selector switch

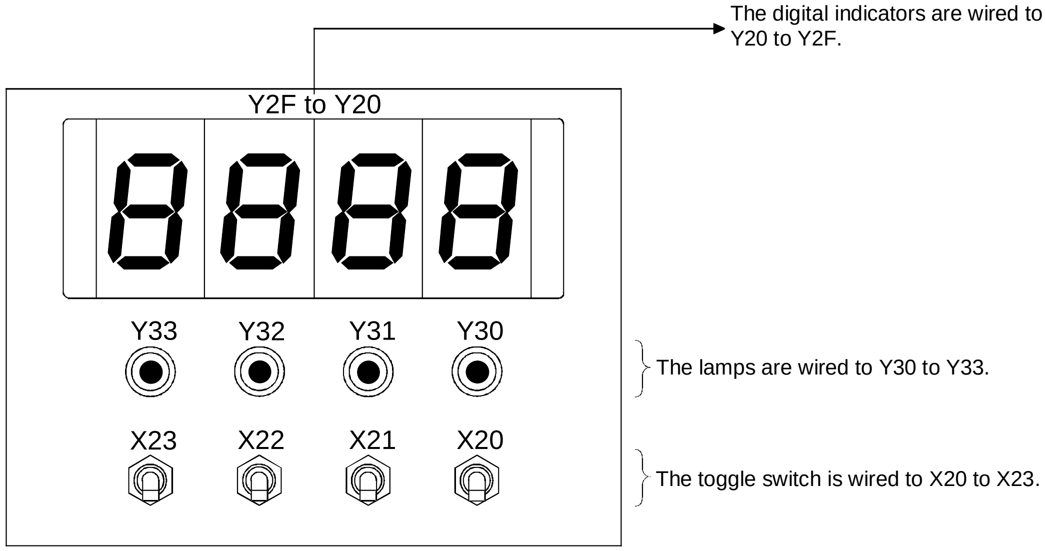
X1F ....Emergency stop

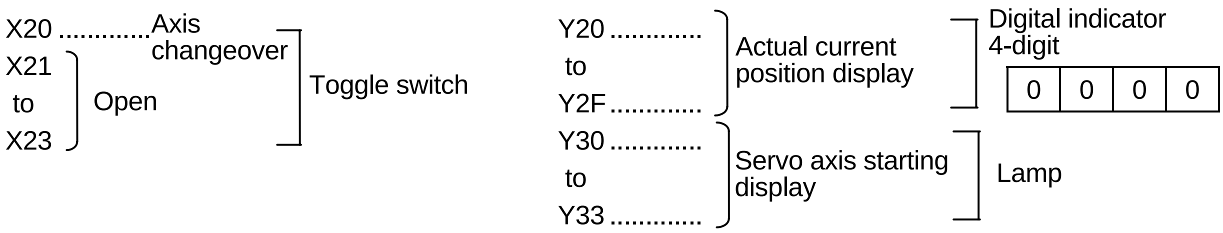
X0................
X1................
X2................
X3................
X4................
X5................
X6................
X7................
X8................

JOG operation
Positioning start
Speed change
command

X9................ Current value change
X0A .... Open
X0B .... Pen UP/DOWN
X0C .... Error reset
X0D .... Stop
X0E .... Sudden stop

Push-button
switch

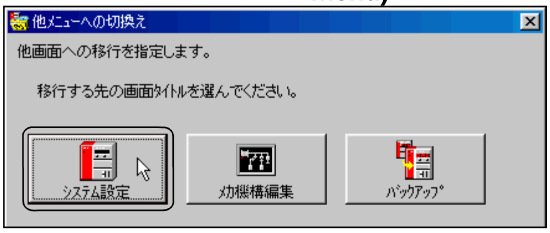

The digital indicators are wired to Y20 to Y2F.
Y2F to Y20
Y33
Y32
Y31
Y30
The lamps are wired to Y30 to Y33.
X23
X22
X21
X20
The toggle switch is wired to X20 to X23.


X20 ....Axis changeover
X21 to X23 Open
Toggle switch
Y20 .... to Y2F ....
Actual current position display
Digital indicator 4-digit
0 0 0 0
Y30 .... to Y33 ....
Servo axis starting display
Lamp

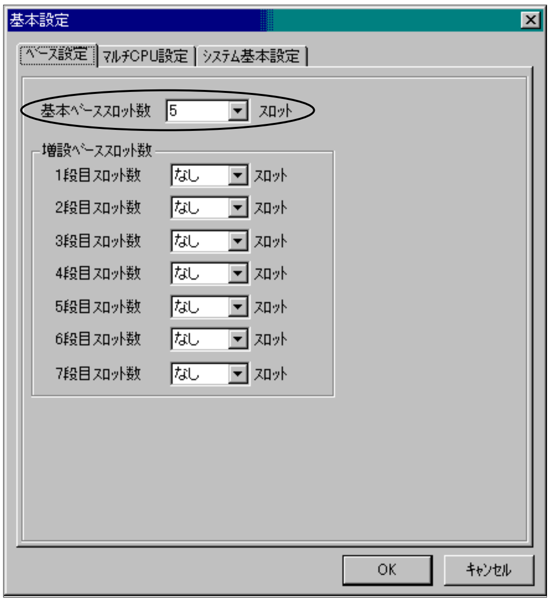
9.3 System setting

The system is set with SW6RN-GSV2P.

(1) System setting window display (when the menu is changed to another menu)



1) Open the CHANGE TO OTHER MENU dialog box from the PROJECT CONTROL menu displayed when the SW6RN-GSV22P is started, and click on the System setting button.

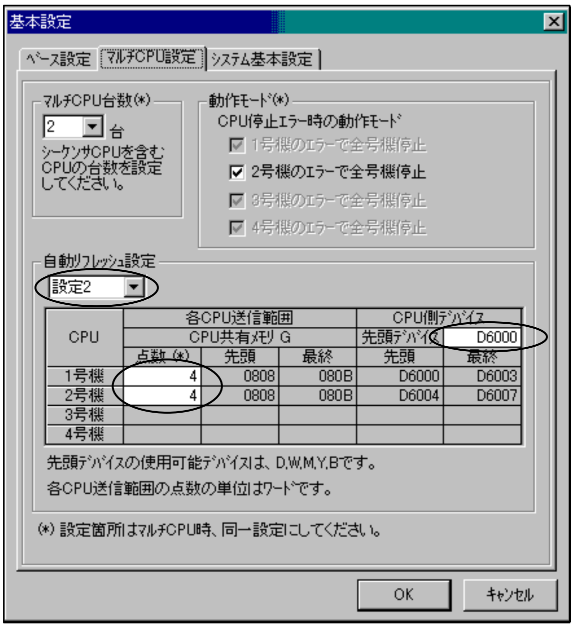


2) The SYSTEM SETTING window and BASIC SETTING dialog box will open.
Set the "Number of main base slots" to '5'. After setting, click on the MULTI-CPU SETTING tab in the Basic setting dialog box.

3) Set the "Number of multi-CPUs" to '2'.
Set the "Automatic refresh setting" to 'Setting 1', and set the following.
"Head device" : 'M0'
"Number of #1 machine points" : '8'
"Number of #2 machine points" : '8'

(Continued on next page)

(Continued from previous page)

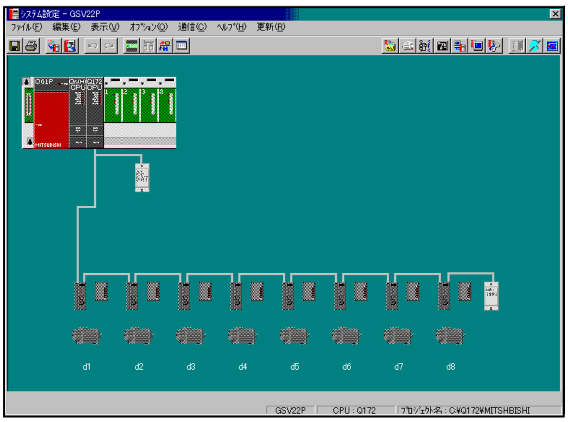


4) Set the "Automatic refresh setting" to 'Setting 2', and set the following.

"Head device"	: 'D6000'
"Number of #1 machine points"	: '4'
"Number of #2 machine points"	: '4'

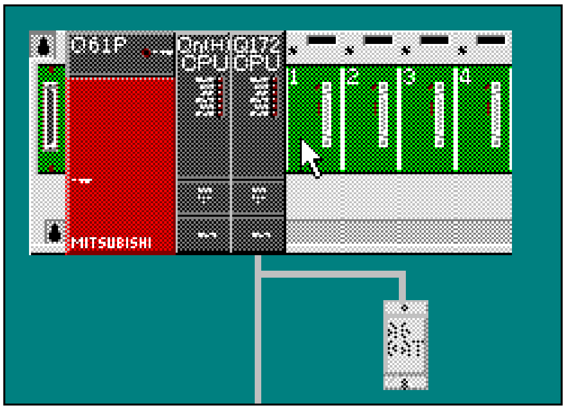
5) Confirm that 'All machines stopped due to error in #2 machine' under "Operation mode" is checked. After checking, click the SYSTEM SETTING tab i the BASIC SETTING dialog box.

6) Set the "Emergency stop input setting" to 'X(PX)', and input '1F' for X(PX).
Check that "Operation cycle setting" is selected under 'Automatic setting' and that "Turn M2000 ON at switch (STOP→RUN)" is under operation at STOP→RUN.
After setting, click on the $\boxed{\text{OK}}$ button in the BASIC SETTING dialog box.

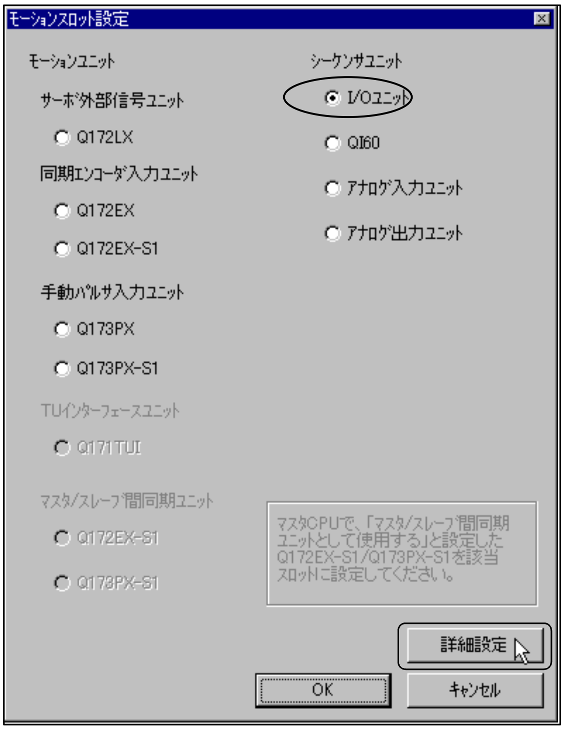


7) The SYSTEM SETTING window will open.
The settings for each unit are explained in (2), (3) and (4) on the following pages.

(2) Setting the motion slot



1) Double-click on the slot 1 of the main base on the SYSTEM SETTING screen to set the input/output hybrid unit in slot 1.

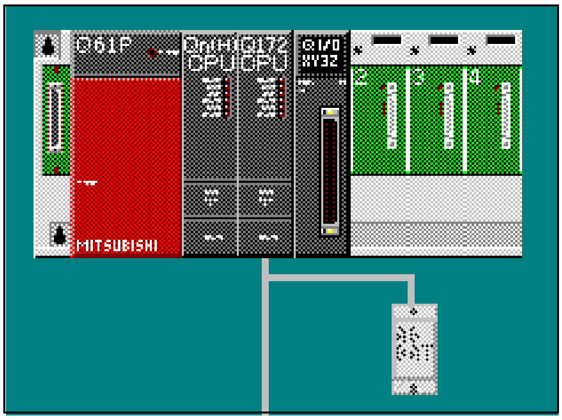


2) The MOTION SLOT setting dialog box will open, so select 'I/O unit' from under "PLC".
After setting, click on the Detail setting button.

3) The I/O UNIT SETTING dialog box will open. Se the "Unit type" to 'Hybrid (same No. for input/ output)', and the number of points to '32'.
After setting, click on the OK button.
The MOTION SETTING dialog box will open again, so click on the OK button.

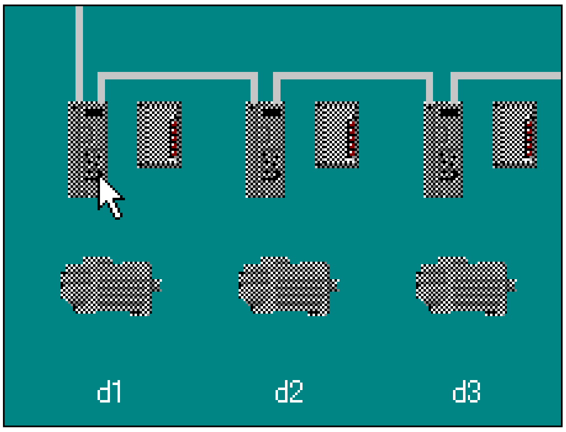
(Continued on next page)

(Continued from previous page)

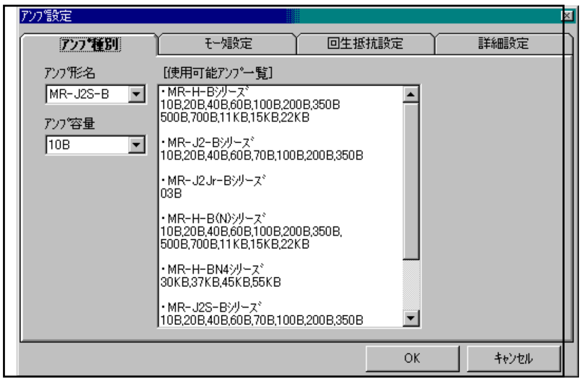


4) This completes setting of the slot 1 input/output hybrid unit.
Slot 2's input/output hybrid unit is controlled by the Q-PLC CPU and does not need to be set.)

(3) Setting the amplifier

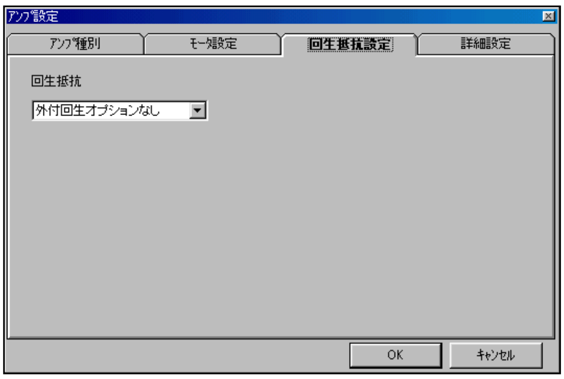


1) To set the first servo amplifier and servomotor, click on the servo amplifier (first [d1] amplifier from left) in the SYSTEM SETTING window.



2) The AMPLIFIER TYPE tab screen will open in the AMPLIFIER SETTING dialog box. Set the "Amplifier type name" to 'MR-J2S-B', and the "Amplifier capacity" to '10B'.
After setting, click on the MOTOR SETTING tab in the AMPLIFIER SETTING dialog box.

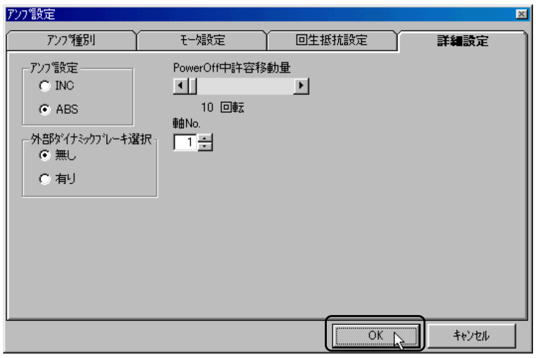
3) Set "Automatic Setting" for the "Motor series".
After setting, click on the REGENERATIVE RESISTOR SETTING tab in the AMPLIFIE SETTING dialog box.



4) Check that "Regenerative resistor" is set to 'No external regenerative option'.
After setting, click on the DETAIL SETTING tab in the AMPLIFIER SETTING dialog box.

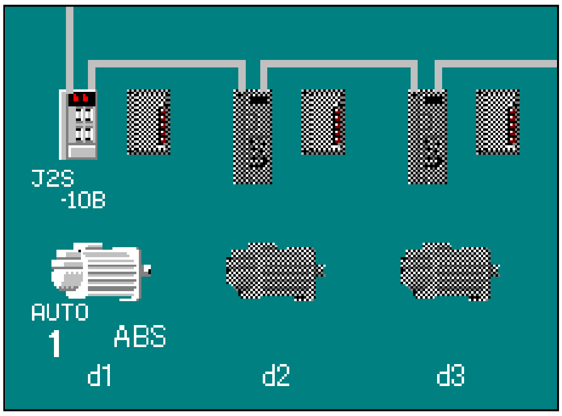
(Continued on next page)

(Continued from previous page)



5) Set the "Axis No." to '1' and check that the other items are set as follows.
"Amplifier setting" : 'ABS'
"External dynamic brake selection": 'None'
"Tolerable movement amount during power-off" : '10 rotations'

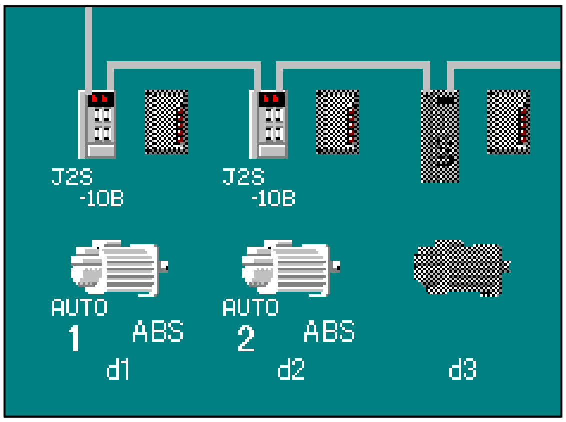
6) Click on the OK button in the AMPLIFIER SETTING dialog box.



7) Next, to set the second servo amplifier and servomotor, double-click on the second (d2) servo amplifier from the left on the SYSTEM SETTING screen.
Carry out the same steps 2) to 4) for the first unit.

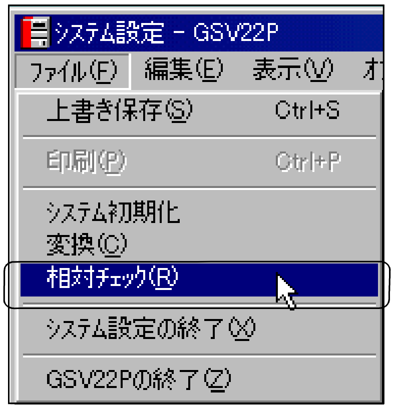
8) Set the "Axis No." in the DETAIL SETTING tab screen to '2'.
The settings other than "Axis No." are the same as the first unit.

9) Click on the OK button in the AMPLIFIER SETTING dialog box.



10) This completes setting of the first (d1) and second (d2) servo amplifiers and servomotors.

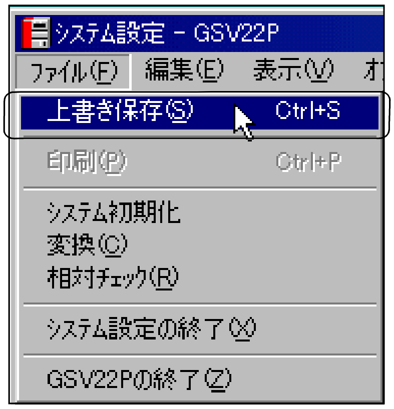
(4) Relative check, Conversion, and Save As



1) After setting the motion slot and amplifier, click on the [File] menu and then the [Relative check] menu in the SYSTEM SETTING window.

2) If the message "No error." is displayed for the check results, click on the OK button.
If the error contents and remedies are displayed, correct the settings, and execute the relative check again.

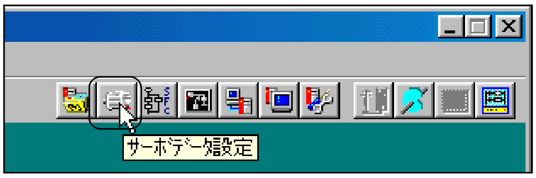
3) Click on the [File] menu and then the [Conversion] menu.
When the message "Completed." is displayed, click on the OK button.



4) Click on the [File] menu, and then the [Save As] menu.
This completes setting of the system.

9.4 Setting the servo data

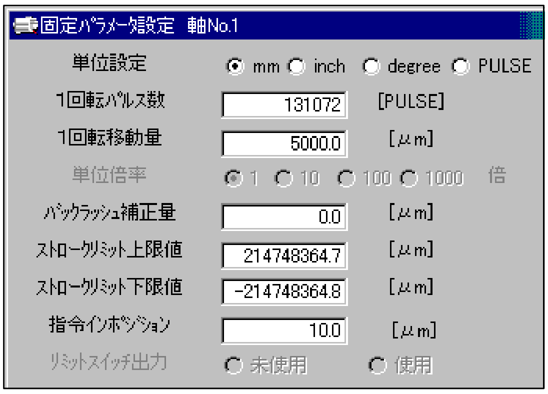
After ending the system settings, set the servo data.



1) Click on the Servo data setting tool button in the SYSTEM SETTING window.

Close the system setting window after the servo system setting window opens.
(Click ✕ at the upper right on the window.)

2) Double-click on the 1-axis section of the Fixed Parameters in the SERVO DATA SETTING window.



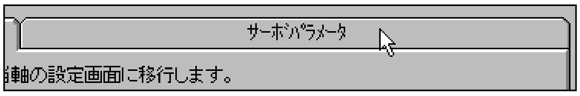
3) The FIXED PARAMETER SETTING/AXIS NO.1 dialog box will open, so set each item as shown on the left.
After setting, click on the OK button.

4) Using the same operations, set the 1-axis/2-axis fixed parameter data, zero point return data and JOG operation data as shown below.

		1軸	2軸
固定パラメータ	単位設定	mm	mm
	1回転パルス数	131072[PULSE]	131072[PULSE]
	1回転移動量	5000.0[μm]	5000.0[μm]
	単位倍率	-	-
	バックラッシュ補正量	0.0[μm]	0.0[μm]
	ストロークリミット上限値	214748364.7[μm]	214748364.7[μm]
	ストロークリミット下限値	-214748364.8[μm]	-214748364.8[μm]
	指令インポジション	10.0[μm]	10.0[μm]
	リミットスイッチ出力	-	-
原点復帰データ	原点復帰方向	逆方向	逆方向
	原点復帰方法	データセット式1	データセット式1
	原点アドレス	-30000.0[μm]	-30000.0[μm]
	原点復帰速度	-	-
	クリープ速度	-	-
	ドグ後の移動量	-	-
	パラメータブロック指定	-	-
JOG運転データ	JOG速度制限値	10000.00[mm/min]	10000.00[mm/min]
	パラメータブロック指定	1	2

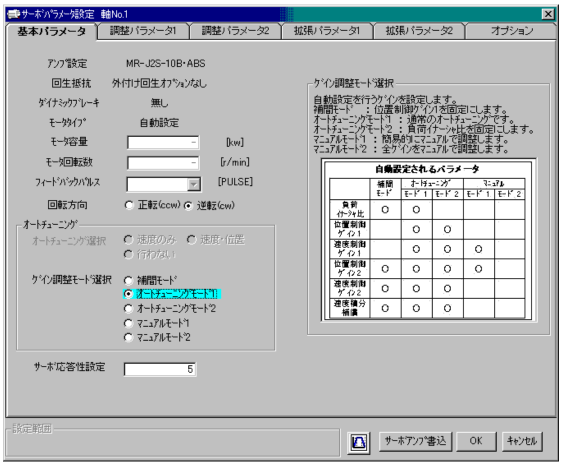
(Continued on next page)

(Continued from previous page)



5) Click on the [SERVO PARAMETER] tab in the SERVO DATA SETTING window.

6) Click on the 1-axis section of the Basic Parameters in the SERVO PARAMETER tab screen.



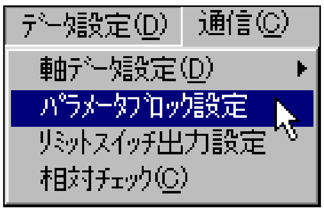
7) The SERVO PARAMETER SETTING/AXIS NO.1 dialog box will open, so set each item as shown below. (Only the basic parameters need to be set.)

Rotating direction : Reverse (CW)
Automatic tuning : Automatic tuning mode 1
Servo response setting : 5

After setting, click on the OK button.

8) Using the same operations, set the 2-axis servo parameter as shown below.

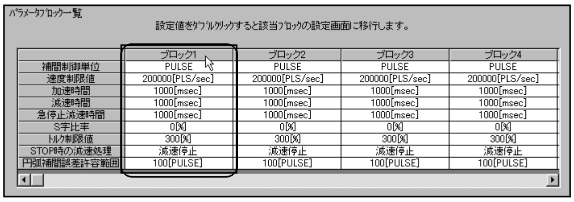
		1軸	2軸
基本ﾊﾟﾗﾒｰﾀ	ｱﾝﾌﾟ設定	MR-J2S-10B-ABS	MR-J2S-10B-ABS
	回生抵抗	外付け回生ｵﾌﾟｼｮﾝなし	外付け回生ｵﾌﾟｼｮﾝなし
	ﾀﾞｲﾅﾐｯｸﾌﾞﾚｰｷ	無し	無し
	ﾓｰﾀﾀｲﾌﾟ	自動設定	自動設定
	ﾓｰﾀ容量[kw]	-	-
	ﾓｰﾀ回転数[r/min]	-	-
	ﾌｨｰﾄﾞﾊﾞｯｸﾊﾟﾙｽ[PULSE]	-	-
	回転方向	逆転(cw)	逆転(cw)
	ｵｰﾄﾁｭｰﾆﾝｸﾞ		
	ｵｰﾄﾁｭｰﾆﾝｸﾞ選択	-	-
	ｹﾞｲﾝ調整ﾓｰﾄﾞ選択	ｵｰﾄﾁｭｰﾆﾝｸﾞﾓｰﾄﾞ1	ｵｰﾄﾁｭｰﾆﾝｸﾞﾓｰﾄﾞ1
	ｻｰﾎﾞ応答性設定	5	5



9) Click on the [Data setting] menu and then the [Parameter block setting] menu in the SERVO DATA setting window.

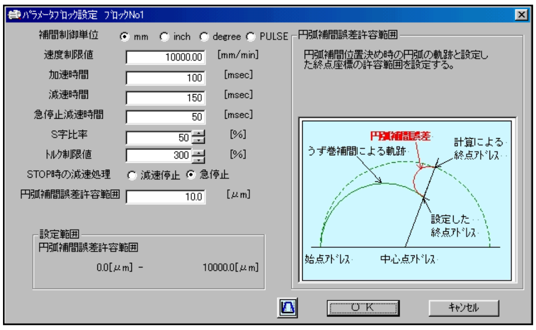
(Continued on next page)

(Continued from previous page)



	ブロック1	ブロック2	ブロック3	ブロック4
補間制御単位	PULSE	PULSE	PULSE	PULSE
速度制限値	200000[PLS/sec]	200000[PLS/sec]	200000[PLS/sec]	200000[PLS/sec]
加速時間	1000[msec]	1000[msec]	1000[msec]	1000[msec]
減速時間	1000[msec]	1000[msec]	1000[msec]	1000[msec]
急停止減速時間	1000[msec]	1000[msec]	1000[msec]	1000[msec]
S字比率	0[%]	0[%]	0[%]	0[%]
トルク制限値	300[%]	300[%]	300[%]	300[%]
STOP時の減速処理	減速停止	減速停止	減速停止	減速停止
円弧補間誤差許容範囲	100[PULSE]	100[PULSE]	100[PULSE]	100[PULSE]

10) The PARAMETER BLOCK SETTING BLOCK NO.1 dialog box will open when the "Block 1" section is double-clicked on, so set the following contents.
After setting, click on the OK button.



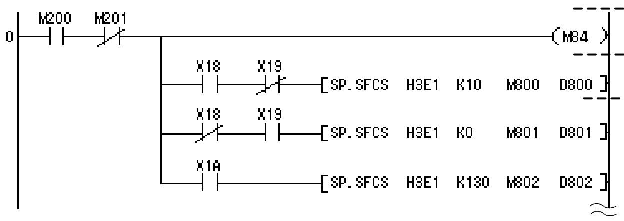
11) Set block 2 as follows in the same manner.

	ブロック1	ブロック2
補間制御単位	mm	mm
速度制限値	10000.00[mm/min]	10000.00[mm/min]
加速時間	100[msec]	400[msec]
減速時間	150[msec]	400[msec]
急停止減速時間	50[msec]	50[msec]
S字比率	50[%]	50[%]
トルク制限値	300[%]	300[%]
STOP時の減速処理	急停止	急停止
円弧補間誤差許容範囲	10.0[μm]	10.0[μm]

12) After setting all the servo data, click [File] and then the [Save As] menu.
This completes setting of the servo data.

9.5 Practice SFC programs

The sequence programs and SFC programs used for practice are listed below. Refer to the following explanations for details on each program.

Normal execution	Start by sequence program	Start by SFC program
• Sequence program 	• [JOG operation] SFC program No.10 • [Real mode main] SFC program No.0	
		• [Zero point return] SFC program No.20
		• [Servo program continuous] SFC program No. 80
• [Initial setting] SFC program No.210: Automatic start	• Virtual mode for practice	

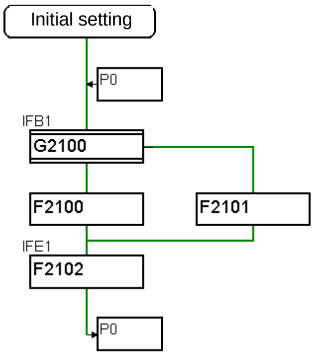
SFC program parameters

No.	Program name	Automatic start	END operation	Number of shifts	Execution timing
0	Real mode main	No			Normal
10	JOG operation	No			Normal
20	Zero point return	No			Normal
80	Servo program continuation	No			Normal
210	Initial setting	Yes			Normal

- **Normal execution program**

[Initial setting] program

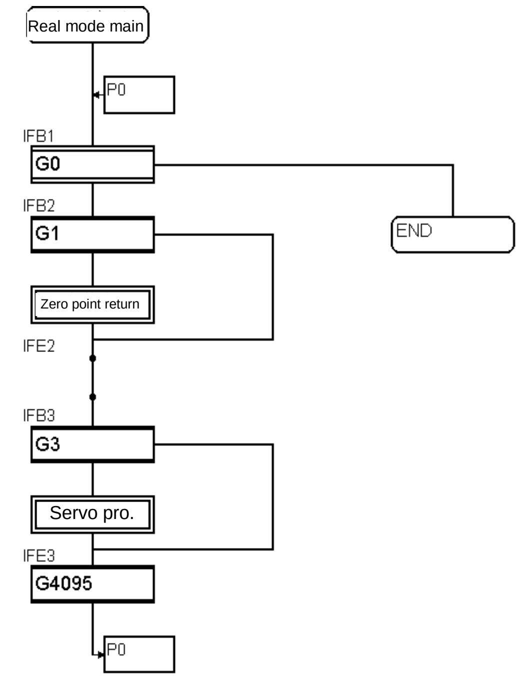
Started automatically. No. 210



- **Program started by sequence program**

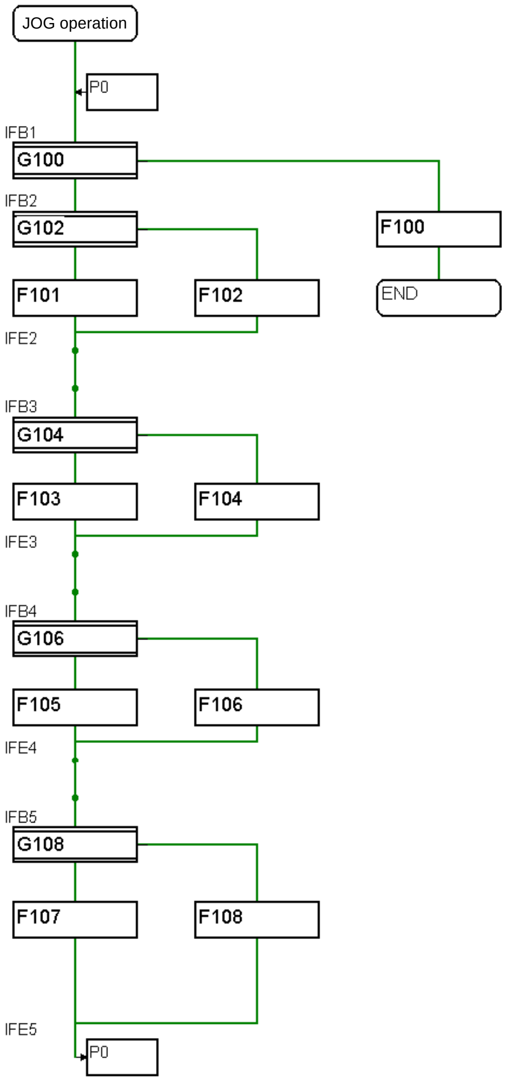
[Real mode main] program No. 0

Started by sequence program.



[JOG operation] program No.10

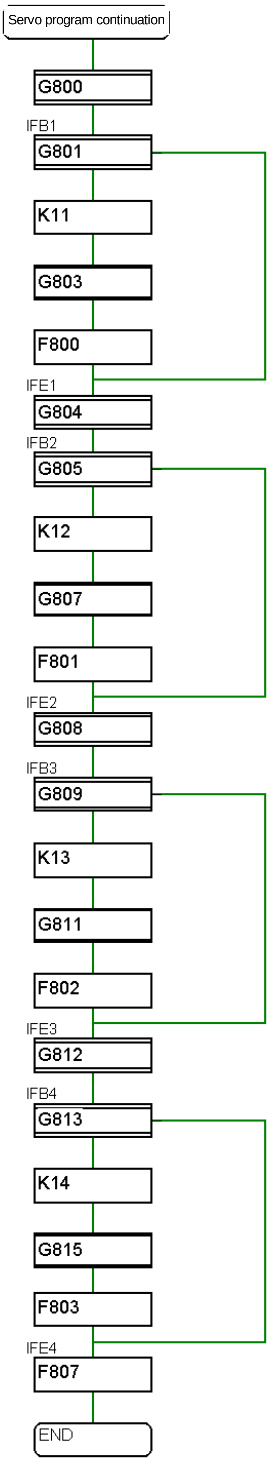
Started by sequence program.



• **Program started by SFC program**

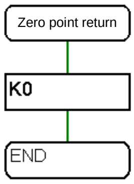
[Servo program continuation] program No. 80

Started by No. 0.

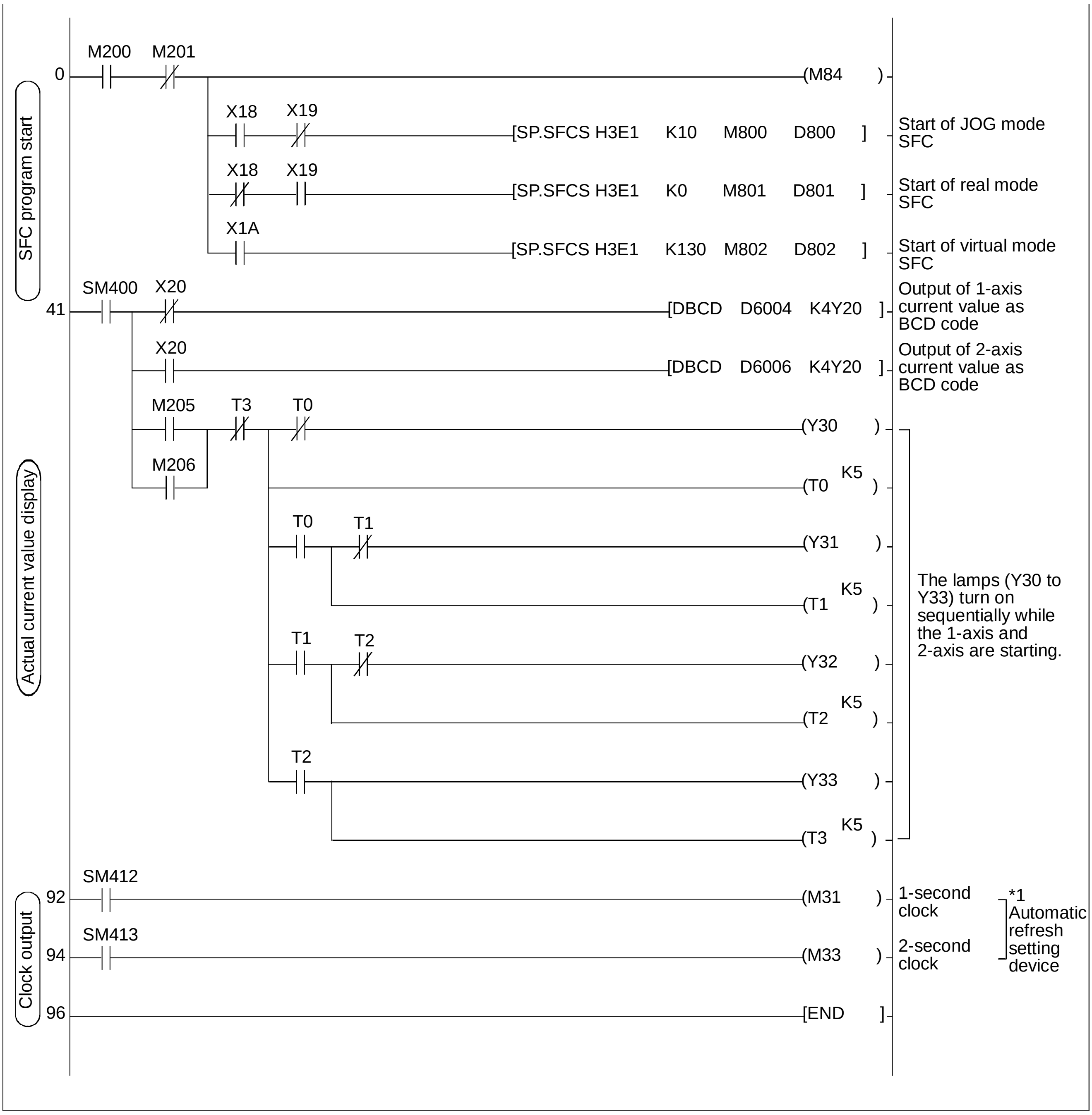


[Zero point return] program No. 20

Started by No. 0.



• **Q02H sequence program**



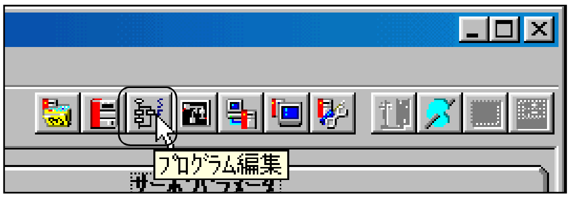
*1 When the clock relay is refreshed for automatic refresh, it may not be refreshed at correct timing depending on automatic refresh timing.

9.6 Creating SFC programs

Create the SFC program used to set the operation of motion control.

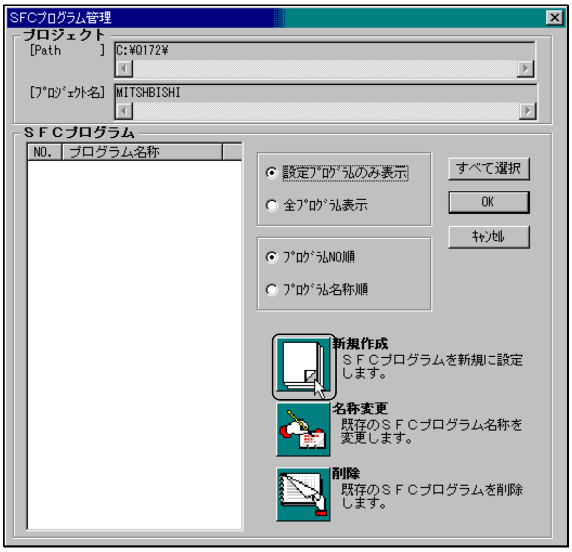
9.6.1 Creating a new SFC program

Creation of a new SFC program starts by assigning the [Program name].

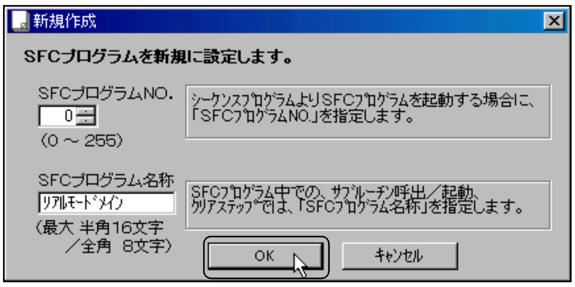


1) Click on the Program edit tool button in the SERVO DATA SETTING window.
The PROGRAM EDIT window and SFC PROGRAM CONTROL dialog box will open.

Close the servo data setting window after the PROGRAM EDIT window opens.
(Click ✕ at the upper right on the window.)



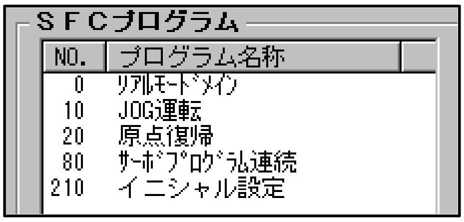
2) Click on the New creation button.



3) The NEW CREATION dialog box will open, so set the program No. of the SFC program to be created.
Input '0' for the "SFC program No.", and 'Real mode main' for the "SFC program name".
Click on the OK button after inputting.

(Continued on next page)

(Continued from previous page)



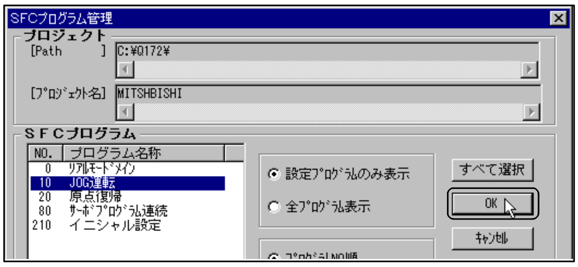
(The procedures for creating the SFC programs other than No.10 and 20 will not be explained here. Refer to the section on the SFC programs for operation and create the programs later.)

4) The set SFC programs will be listed.
Click the New creation button again to create the SFC programs shown below.

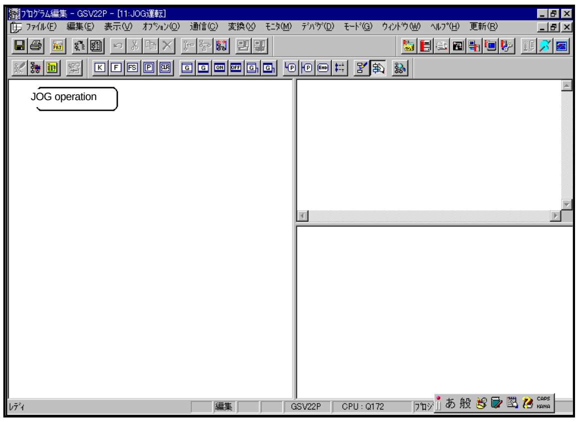
No.	Program name
0	Real mode main
10	JOG operation
20	Zero point return
80	Servo program continuation
210	Initial setting

9.6.2 Creating the SFC diagram

Arrange the SFC diagram symbols to create the SFC diagram.

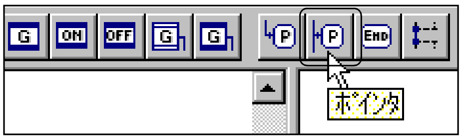


1) Select the "10 JOG operation" from the SFC program list in the SFC PROGRAM CONTROL dialog box, and click on the OK button.

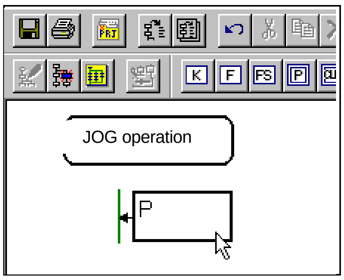


2) The program edit screen (used for creation of each SFC program) will open.

> When the SFC program is selected by mistake, causing another SFC program to be displayed, click on the SFC program control tool button to select the desired SFC program again.



3) Click on the Pointer tool button in the PROGRAM EDIT screen.

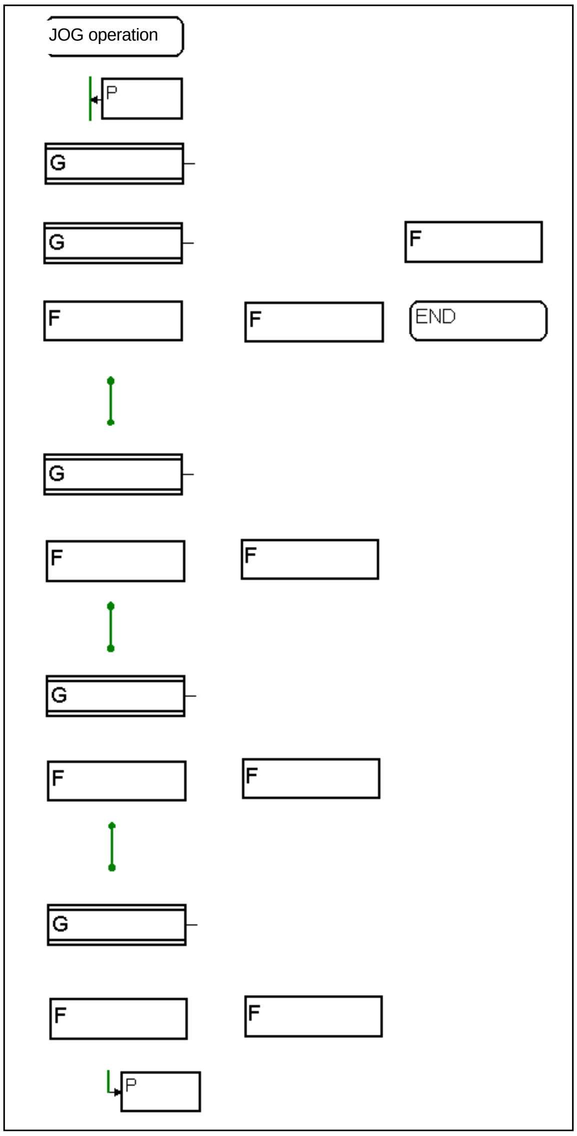


4) Click a random position to arrange the SFC diagram symbol pointer.

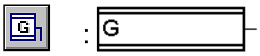
> Multiple points can be arranged.
> After arrangement, click the right mouse button to clear the SFC diagram symbol.

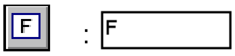
(Continued on next page)

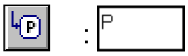
(Continued from previous page)

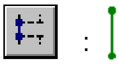


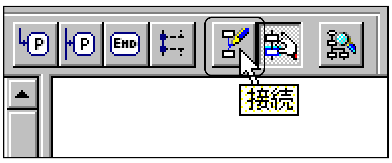
5) Click on each tool button thereafter in the similar manner to arrange each SFC diagram symbol as shown on the left.

 (SHIFT Y/N transition)

 (1-time execution type operation control step)

 (Jump)

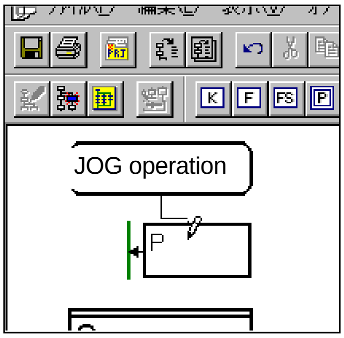
 (Continuation of connection - branch)



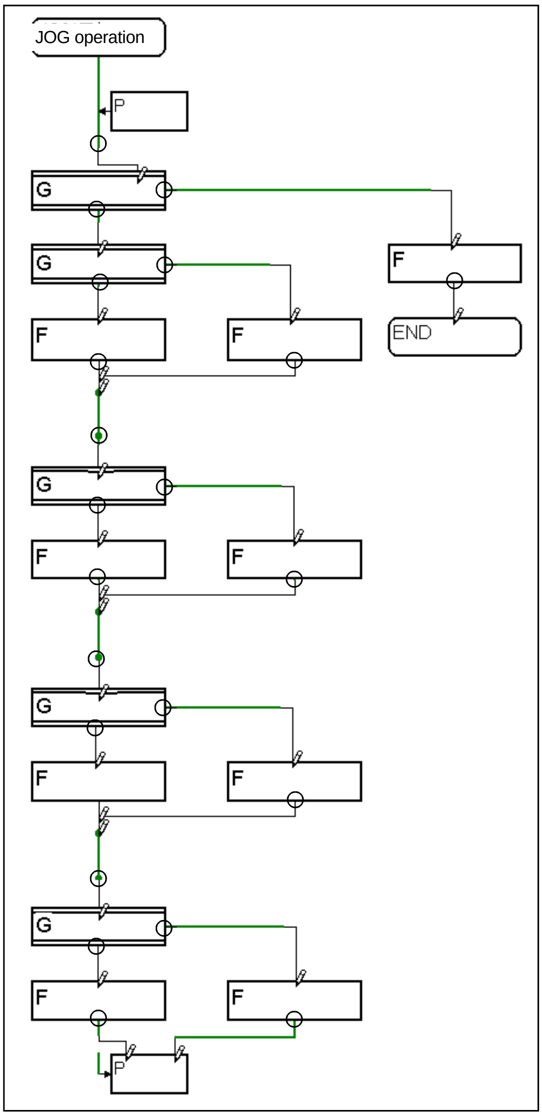
6) Connect the SFC diagram symbols arranged. Click on the Connection tool button in PROGRAM EDIT screen.

(Continued on next page)

(Continued from previous page)

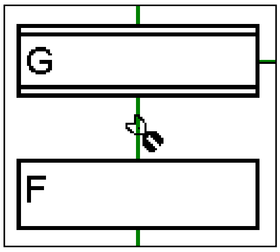


7) The shape of the mouse cursor will change when moved over the SFC diagram symbol.
Drag the start point of SFC program and the pointer to connect the SFC diagram symbols.



8) Connect the other SFC diagram symbols in the same manner.

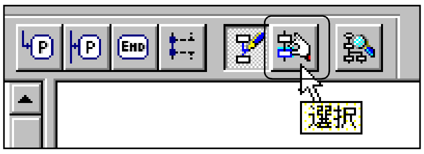
When the connection is made incorrectly, click on the Selection tool button () on the PROGRAM EDIT screen, and then move the mouse cursor over the connection line. Click the mouse to cut the line.



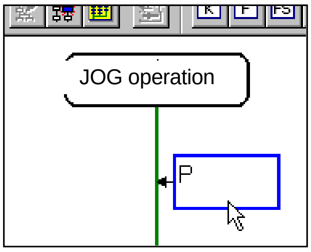
(Continued on next page)

(Continued from previous page)

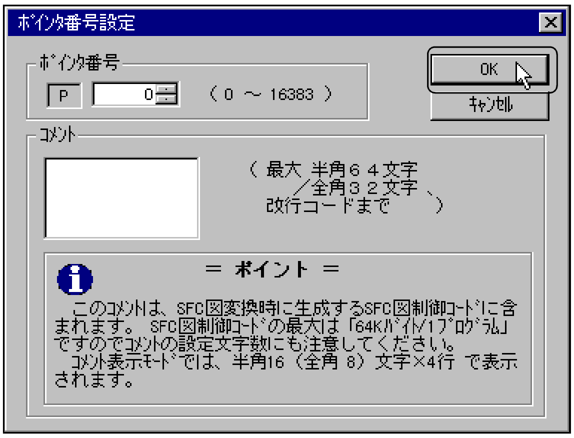
9) Click on [Edit] and then the [Alignment] menu in the PROGRAM EDIT window.
The arranged SFC diagram symbols will be aligned.



10) Set the program No. and pointer No. to the arranged SFC diagram symbols.
Click on the Selection tool button in the PROGRAM EDIT screen.



11) Double-click on the pointer (P).



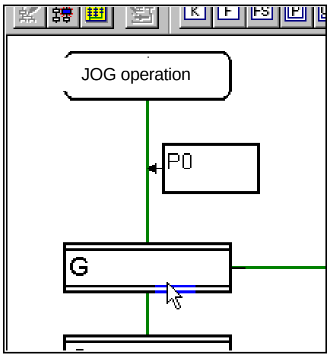
12) The pointer No. setting dialog box will open.
Input '0' for the "Pointer No.", and click on the OK button.

The pointer No. can be set between 0 and 16383 for each SFC program.

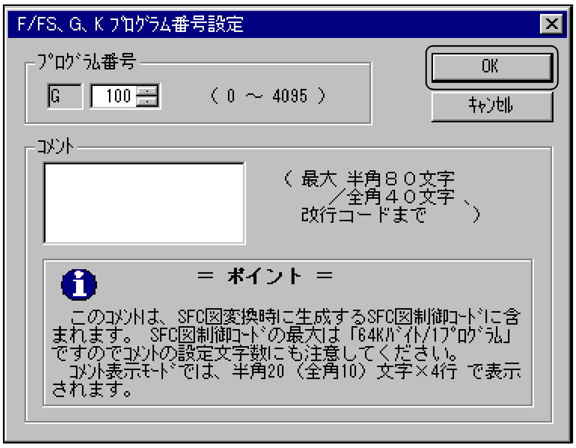
(Note that 'P0' for the SFC program No.0 differs from 'P0' for the SFC program No.10.)

(Continued on next page)

(Continued from previous page)



13) The pointer No. is set to '0'.
Next, double-click on the transition (G).

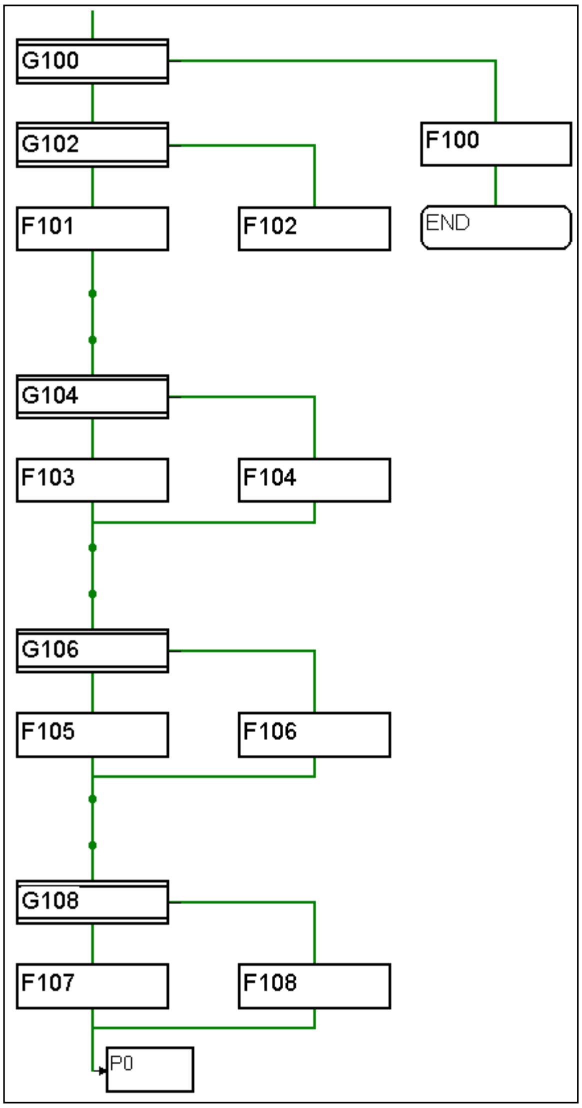


14) The PROGRAM NO. SETTING dialog box will open
Input '100' for the "Program No.", and click on the OK button.

> The program No. is common in the project.

(Continued on next page)

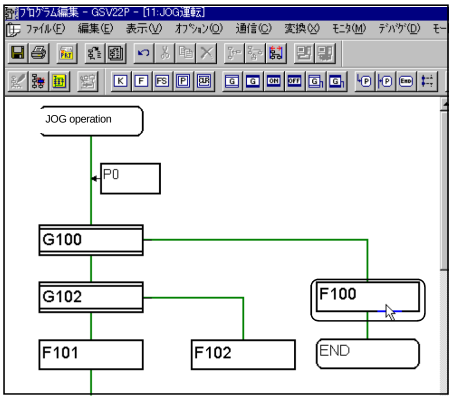
(Continued from previous page)



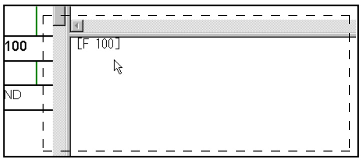
15) The program No. 'G100' is set to the transition.
Set the program No. and pointer No. for the other SFC diagram symbols shown at left in the same manner.

9.6.3 Inputting the transition and operation control step

Set the conditional expression and operation expression to the transition and operation control steps arranged in the SFC diagram.



1) Click on and select the operation control step 'F100'.



2) When '[F100]' appears at the area on the lower right of the screen (step's PROGRAM EDIT screen), double-click it.

3) The OPERATION CONTROL PROGRAM/ TRANSITION PROGRAM EDIT dialog box will open. Click on the Command selection button.

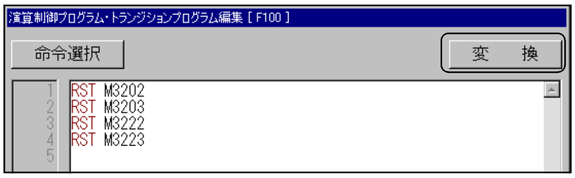
The command can also be input and set directly.
Skip to step 5) when inputting directly.

4) The COMMAND WIZARD dialog box will open. Select as shown below, and click on the Completion button.

Type : Bit device control
Description : RST
Description example: RST M0

(Continued on next page)

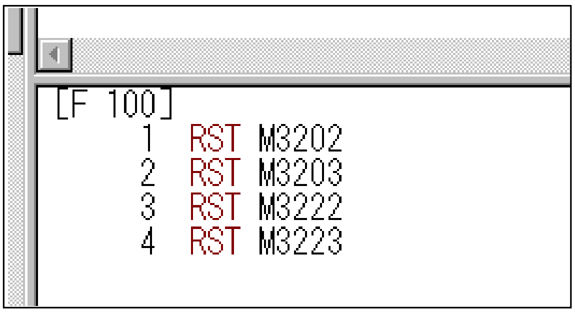
(Continued from previous page)



5) The command will be set as 'RST M0', so input 'M3202' for 'M0'.
Click on the Enter key to feed the line, and input 'RST M3203', 'RST M3222' and 'RST M3223'.
After inputting, click on the Conversion button.

6) When the message "Conversion completed." appears, click on the OK button.

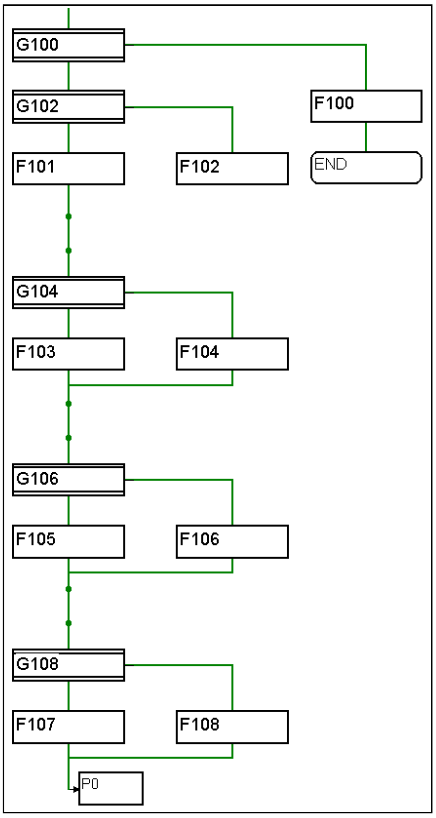
7) Click on the OK button.



8) The set command will appear on the step's PROGRAM EDIT screen.

(Continued on next page)

(Continued from previous page)

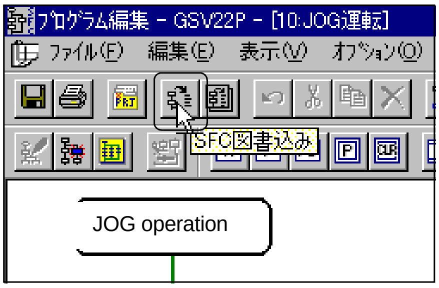


9) Set the operation expression and conditional expression for the following operation control program and transition program in the same manner.

[G100]	PX18
[G102]	PX5*!M3203
[G104]	PX3*!M3202
[G106]	PX1*!M3223
[G108]	PX7*!M3222
[F100]	RST M3202 RST M3203 RST M3222 RST M3223
[F101]	D640L=K100000 SET M3202
[F102]	RST M3202
[F103]	D640L=K100000 SET M3203
[F104]	RST M3203
[F105]	D642L=K100000 SET M3222
[F106]	RST M3222
[F107]	D642L=K100000 SET M3223
[F108]	RST M3223

*(Logical product: AND)　　M3202 (Axis 1 Forward)
!(OFF)　　M3222 (Axis 2 Forward)
M3203 (Axis 1 Reverse)　　M3223 (Axis 2 Reverse)

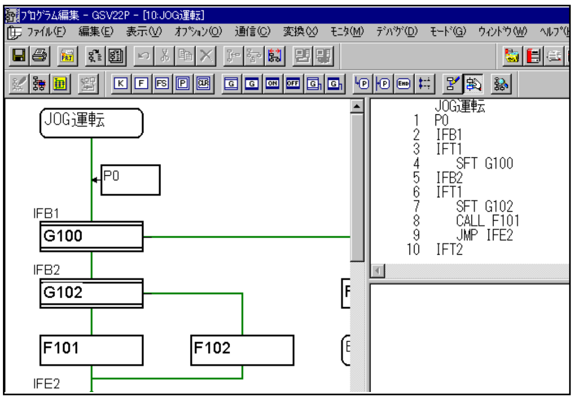
10) Click on the SFC diagram write tool button on the PROGRAM EDIT screen.



(Continued on next page)

(Continued from previous page)

11) When the message "Completed normally." appears, click on the OK button.



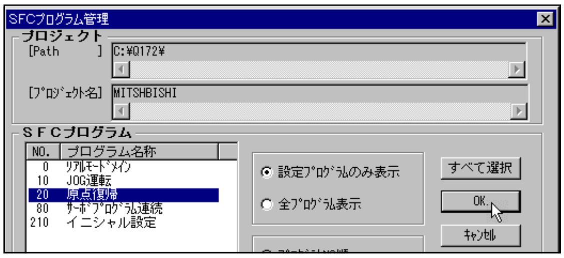
12) The SFC program will be listed on the right side on screen.

13) Click on [File], and then the [Save As] menu on the PROGRAM EDIT window.
This completes creation of the JOG operation SFC program.

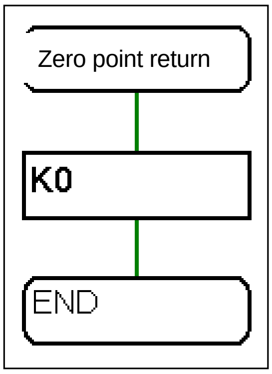
9.6.4 Inputting the motion control step

Set the motion control steps (used for positioning control, etc.).
Create the SFC program for zero point return first.

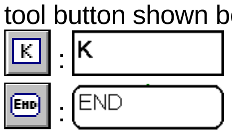
1) Create the SFC program for zero point return.
Click on the SFC program control tool button on the PROGRAM EDIT screen.



2) Select "20 Zero point return" from the SFC program list in the SFC PROGRAM CONTROL dialog box, and click on the OK button.

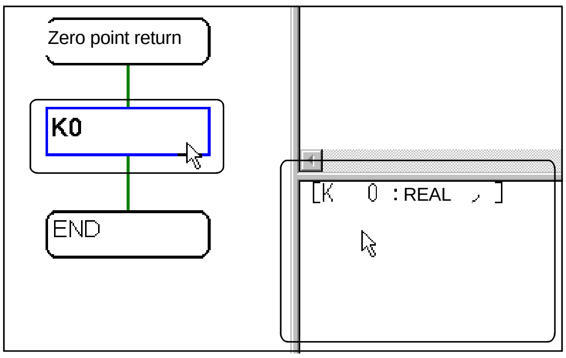


3) Create the zero point return SFC program with the following procedure.
   (a) Arrange the SFC diagram symbols.
   Arrange the SFC diagram symbols using each tool button shown below.



   : (Motion control step)
   : (END)

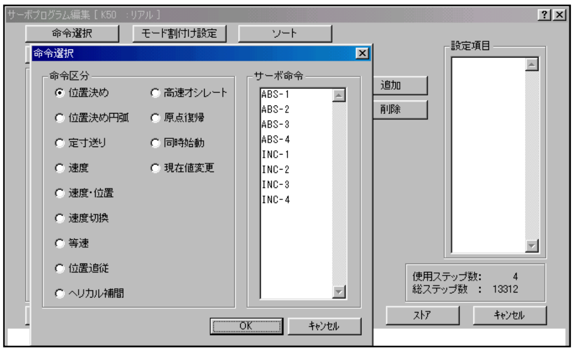
   (b) Connect the SFC diagram symbols using the Connection tool button.
   (c) After selecting the servo program No. using the Selection tool button, set the servo program No.



4) Click on the motion control step 'K0' to select the control step, and double-click on the PROGRAM EDIT screen.

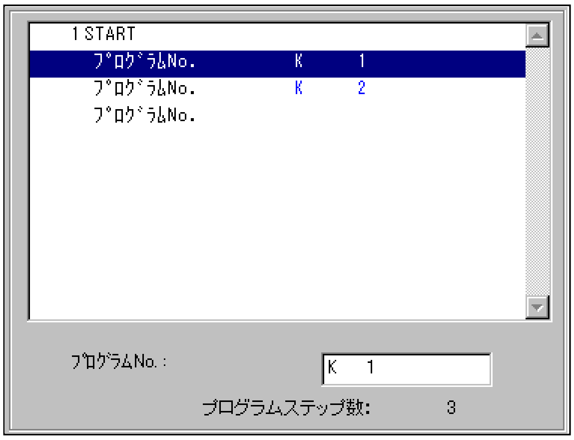
(Continued on next page)

(Continued from previous page)



5) The SERVO PROGRAM EDIT dialog box and COMMAND SELECTION dialog box will open

6) Set the " Command division " in the COMMAND SELECTION dialog box to 'Simultaneous start', and the "Servo command" to 'START'. Then click on the OK button.



7) Input the 'K1' in the "Program No.:" text box.
Click the Enter key to feed the line, and input 'K2'.

(Continued on next page)

(Continued from previous page)

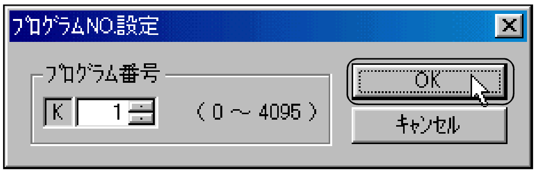
8) Click on the Store button.
This completes setting of the 'K0' motion control step.

9) Carry out the same procedure to create the motion control steps used for the other SFC programs shown on the following pages.

Outline of motion control step editing

1) Click on the Program No. setting button.

2) Input the motion control step No. to be edited next in the "Program No." in the PROGRAM NO. SETTING dialog box, and click the OK button.

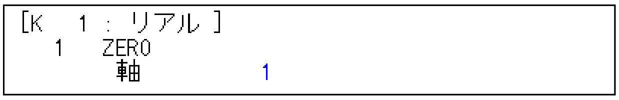


3) Note that the setting items required may differ depending on the type of command for Command selection → Axis No. → Address → Speed (→ M code, dwell time).
To set the M code or dwell time, select the "Setting item" on the right of the SERVO PROGRAM EDIT dialog box after selecting the command, and click the [Add] button. The setting will be enabled.
Note that the item displayed at "Setting item" may differ depending on the command selected.

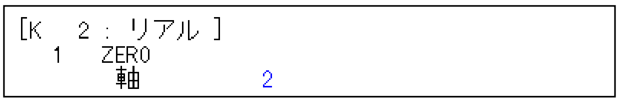
4) Click the Store button.

REAL
Simultane PROGRAM NO.
PROGRAM NO.

Zero point REAL
(Axis AXIS



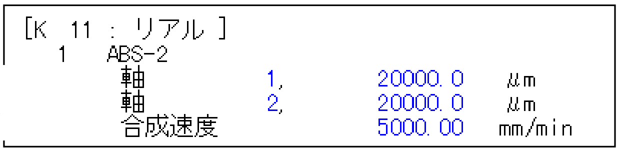
Zero point REAL
(Axis AXIS



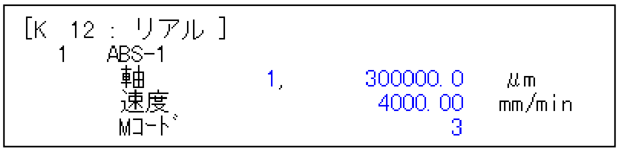
(Continued on next page)

(Continued from previous page)

REAL
Positioning AXIS
(2-axis lin AXIS
COMPOSITE VELOCITY

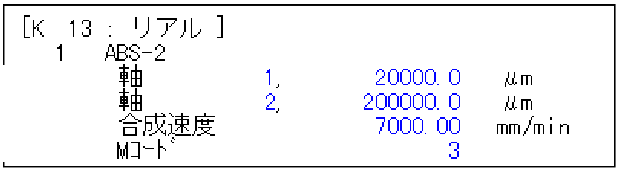


REAL
Positioning AXIS
(1-axis lin VELOCITY
M CODE



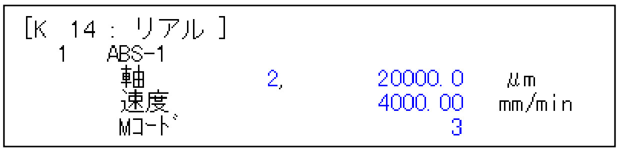
Select the "M code" from the "Setting item", and click on the Add button.

REAL
Positionin AXIS
(2-axis lin AXIS
COMPOSITE VELOCITY
M CODE



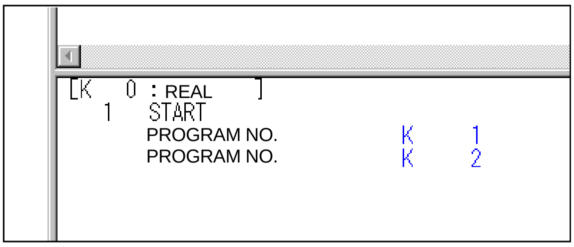
Select the "M code" from the "Setting item", and click on the Add button.

REAL
Positioning AXIS
(1-axis lin VELOCITY
M CODE



Select the "M code" from the "Setting item", and click on the Add button.

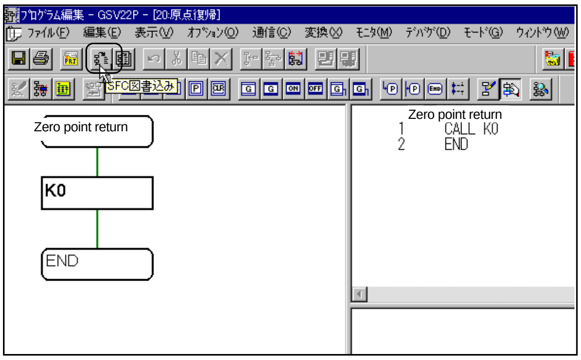
10) After creating the motion control step, click on the Cancel button to close the SERVO PROGRAM EDIT dialog box.



11) The set motion control step will appear on the PROGRAM EDIT screen.

(Continued on next page)

(Continued from previous page)



12) Click on the SFC diagram write tool button on the PROGRAM EDIT screen to convert the program into a SFC program.

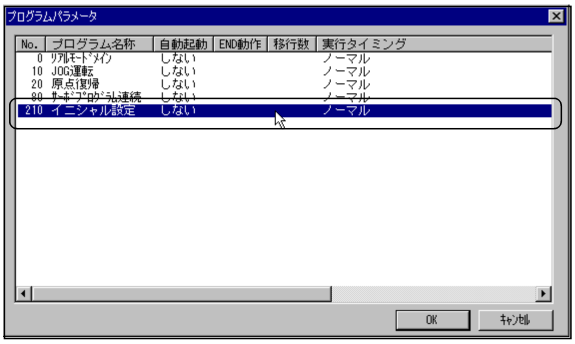
Refer to section 9.9 and create the SFC programs with the following numbers.
0 80 210

13) Click on [File], and then the [Save As] menu on the PROGRAM EDIT window.
This completes inputting of the motion control step.

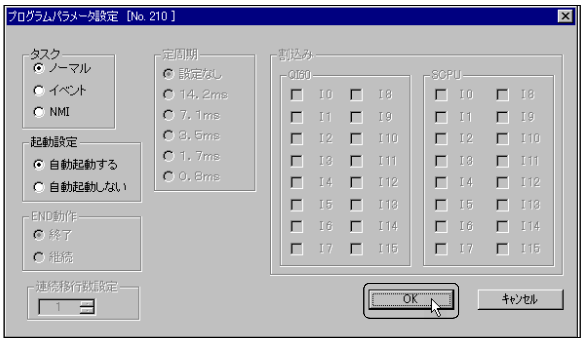
9.6.5 SFC program parameter setting and batch conversion

Set the parameters and convert them for the created SFC program.

1) Click on [Option], [SFC parameter setting] and then the [Program parameter] menu on the PROGRAM EDIT screen.



2) The PROGRAM PARAMETER screen will open. The created SFC program will be listed, so double-click on "Initial setting".

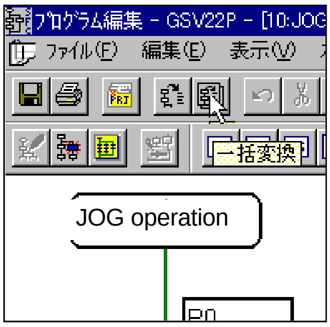


3) The PROGRAM PARAMETER SETTING dialog box will open.
Set "Start setting" to 'Automatic start'.
After setting, click on the OK button.

Task (execution timing) setting
1. Normal task
   Execute at motion cycle (dead time)
2. Event task
   - Execute at constant cycle (0.8ms, 1.7ms, 3.5ms, 7.1ms, 14.2ms)
   - Execute upon input of external interrupt QI60 I0 to I15
   - Execute upon interrupt from PLC (I0 to I15) (GINT command)
3. NMI task (Non-Maskable Interrupt)
   Execute upon input of external interrupt QI60 I0 to I15
   Since it has higher priority than external input of event task, it is executed preferentially even if the interrupt inhibit (DI) occurs.

(Continued on next page)

(Continued from previous page)



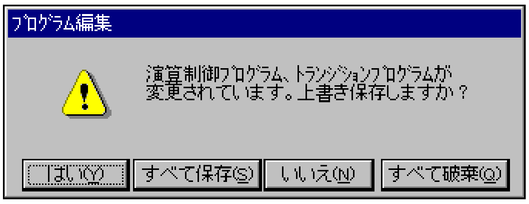
4) Convert the created SFC diagram into an SFC program as a batch.
Click on the Batch conversion tool button in the PROGRAM EDIT screen.

5) When the message "Completed normally." appears, click on the OK button.
This completes creation of the SFC program.

If a "CAUTION" message appears, correct the SFC program, or exit the GSV22P.

(Ending the SW6N-GV22P)

6) Click on [File] and then the [GSV22P end] menu on the PROGRAM EDIT screen.
(The [File], [GSV22P exit] menu is located in each function window.)



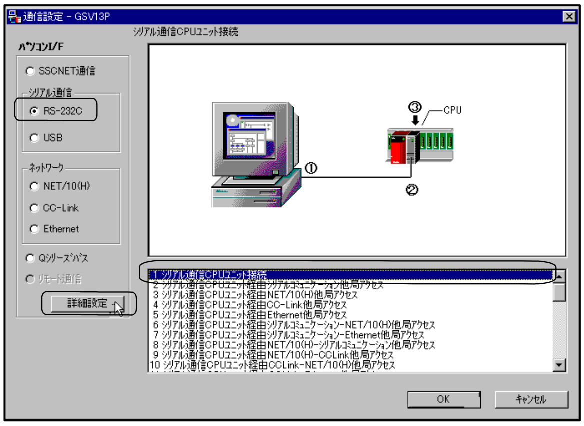
7) If the setting data has not been saved, a message to confirm overwriting will appear.
Click on the Yes button.
(The message for the PROGRAM EDIT screen is shown on the left.)

9.7 Writing to the motion CPU

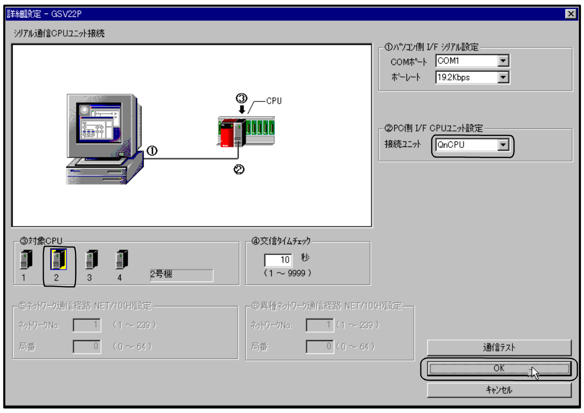
Write the servo setting data and SFC program to the Q172CPU.

1) Set the Q motion CPU to STOP.

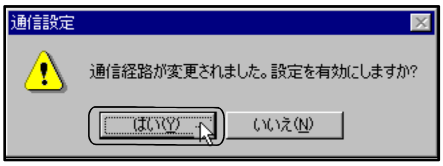
2) Click on the [Communication] and then the [Communication setting] menu on the PROGRAM EDIT screen.



3) The COMMUNICATION SETTING dialog box wil open, so check "RS-232C", select "1. Serial communication CPU connection" and click on the Detail setting button.



4) The DETAIL SETTING dialog box will open, set the "PC side I/F CPU setting" to 'QnCPU', and the "Object CPU" to '#2 machine'.
After setting, click on the OK button.
The COMMUNICATION SETTING dialog box wil open again, so click on the OK button.

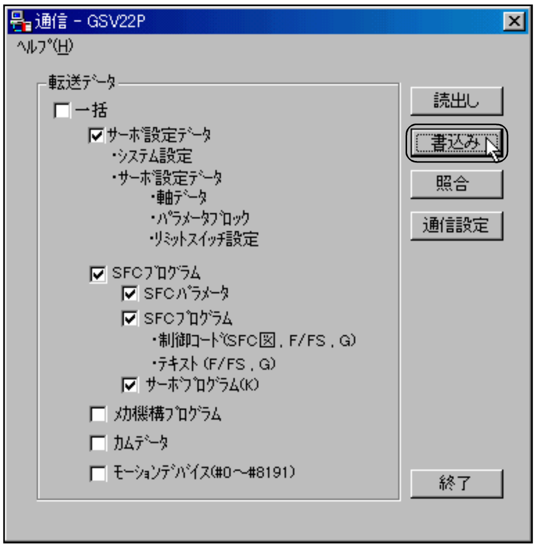


5) When the message shown at left appears, click on the OK button.

(Continued on next page)

(Continued from previous page)

6) Click on the [Communication] and then the [Transfer] menu in PROGRAM EDIT screen.



7) The COMMUNICATION dialog box will open, so check "Servo setting data" and "SFC program", and then click on the Write button.

8) Since the dialog box to confirm the motion CPU type and writing execution, click on the Yes button.

9) When the message "Completed normally." is displayed, click on the OK button.

10) Reset the Q-PLC CPU.

11) Set the mode changeover switch (arranged on the operation panel of practice machine) to "Real", and run the Q-PLC CPU and Q motion CPU.

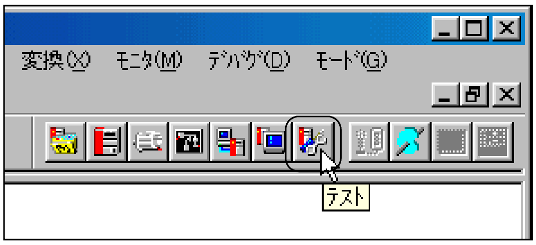
The state is normal if "RUN" of Q02HCPU and "RUN/M.RUN" of Q172CPU turn ON.

9.8 Test operation

PLC READY (M2000) must be turned OFF before starting test operation.
"Stop" the Q motion CPU.

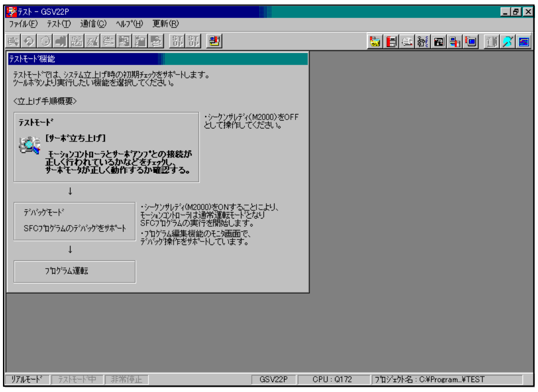
9.8.1 JOG operation

Perform the JOG operation in the test mode to level the disc attached to the servomotor.



1) Click on the Test tool button in the PROGRAM EDIT window.

After the TEST window opens, close the PROGRAM EDIT window.
(Click on × at the upper right of the window.)



2) The test start request will be executed automatically when the TEST window opens.
Each tool function's tool button is validated when the motion CPU is set to the test mode.

3) Click on the JOG operation tool button.

4) The JOG OPERATION dialog box will open, so set the "Axis No. designation" to '1'.
The JOG operation will start when the Forward or Reverse button is pressed.
This operation will level the axis 1 disc.
(When the "JOG speed" is set to small value, the JOG operation speed will decrease, allowing the delicate movements.)

(Continued on next page)

(Continued from previous page)

5) Set "Axis No. setting" to '2' so that the disc is levelled in the same manner as axis 1.

6) Alter the disc is levelled with JOG operation, click on the END button to close the JOG OPERATION dialog box.
This completes JOG operation.

9.8.2 Running the servo program

Run the zero point return and positioning servo program set with the program operation in the test mode.

1) Click on the Program operation tool button.

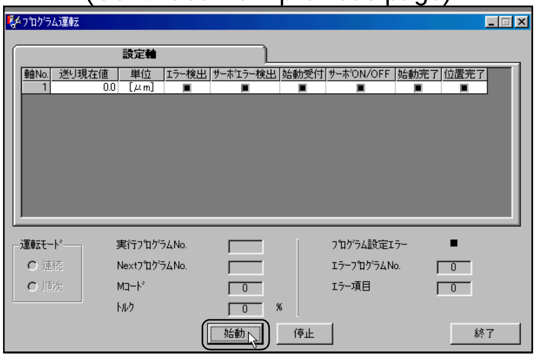
2) A dialog box for selecting the type of program operation will open, so click on the Independent operation button.

3) The PROGRAM OPERATION (INDEPENDENT) dialog box will open, so set the spin box to '1', and click on the Program No. setting button.

4) Click on the Setting completed button in the PROGRAM OPERATION (INDEPENDENT) dialog box.

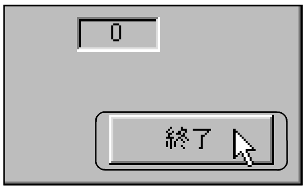
(Continued on next page)

(Continued from previous page)



5) The PROGRAM OPERATION dialog box will open, so click on the START button.
(The axis 1 is returned to zero point.)
Note: The motor will not rotate during data-set type zero point return. (The current feed value is set to '–30000.0µm'.)

6) The message "Program operation completed." will appear, so click on the OK button.

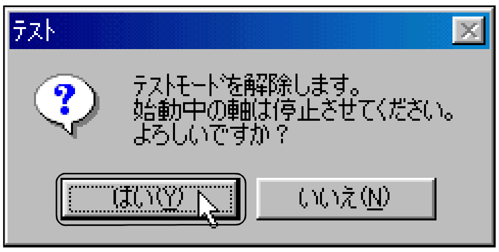


7) Click on the END button in the PROGRAM OPERATION dialog box.

8) Start operation of the servo program No.2 in the same manner.
(The axis 2 will return to the zero point.)

9) Start operation of the servo program No.11 in the same manner.

10) Click on [Test] and then the [Test mode reset] menu in the TEST window.



11) A dialog for confirming cancellation of the test mode will open, so click on the YES button.
This completes operation of the program using the test function.

9.9 Program for operation

This operation sequence/SFC program has been prepared for the SW6RN-GSV22P (for Q172).
The explanatory drawing of the practice machine's operation panel, is shown in section 9.2.

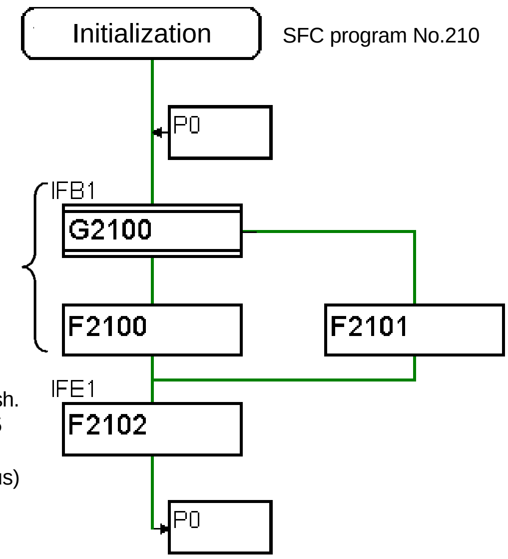
9.9.1 Initialization

When the RUN/STOP switch of Q motion CPU is set to RUN, PLC READY (M2000) will turn ON for the Q-PLC CPU.
Example of SFC program for when the all-axis servo is started upon reception of the PCPU READY flag (M9074).

(1) Program example

M2042 (all-axis servo start command flag) is turned ON in the following cases.
- M9074 (PCPU READY flag): ON
- M9075 (Test mode flag): OFF
- M9076 (Emergency stop input flag): ON

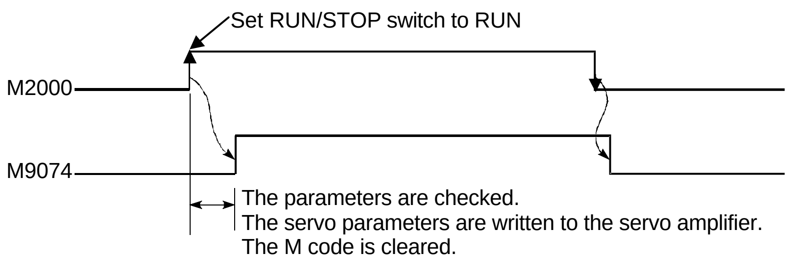
The following signals are output to the M200, M201, M205 and M206 set by automatic refresh.
- M2415 (Axis 1 servo ready flag) and M2435 (Axis 2 servo ready flag)
- M2044 (Real/virtual mode changeover status)
- M2001, M2002 (Axis 1, Axis 2 start accept flag)



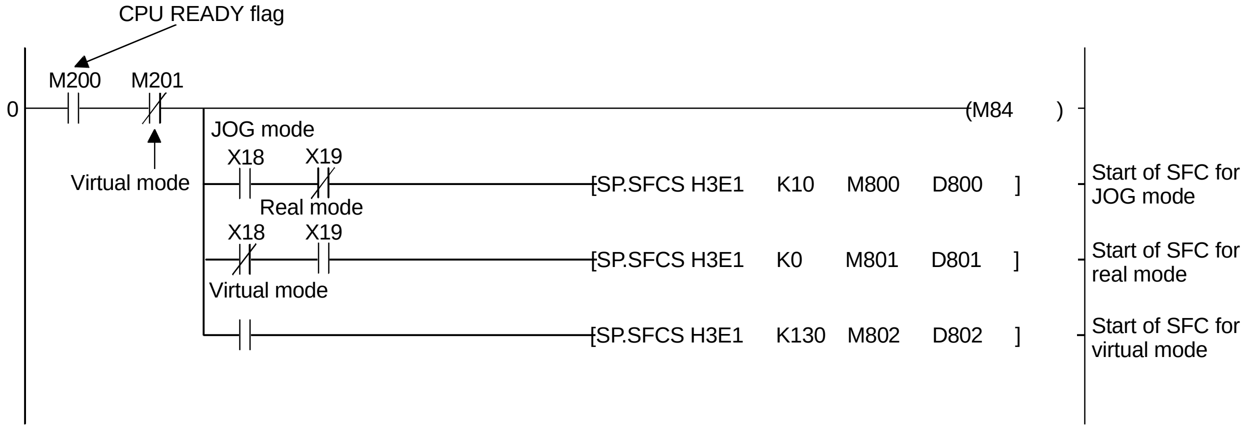
[Transition]	G2100 M9074*!M9075*M9076	
[Operation control step]	F2100 SET M2042	F2101 RST M2042
	F2102 SET M200 = M2415*M2435 SET M201 = M2044 SET M205 = M2001 SET M206 = M2002 RST M200 = !M2415*!M2435 RST M201 = !M2044 RST M205 = !M2001 RST M206 = !M2002	

M200	: M2415 and M2435 ON/OFF state	M201	: M2044 ON/OFF state
M205	: M2001 ON/OFF state	M206	: M2002 ON/OFF state
M2001	: Axis 1 start accept flag	M2002	: Axis 2 start accept flag
M2042	: All-axis servo start command flag	M2044	: Real/virtual mode changeover status
M2049	: All-axis servo ON accept flag	M9074	: PCPU READY flag
M2415	: Axis 1 servo ready flag	M2435	: Axis 2 servo ready flag
M9075	: Test mode flag	M9076	: Emergency stop input flag

[Timing chart]



(2) Q02HCPU sequence program



9.9.2 JOG operation

When the forward JOG start signal (M3202/axis 1) or reverse JOG start signal (M3203/axis 1) is turned ON, the axis will move at the speed stored in the JOG operation speed register (following table) and the details (acceleration/deceleration time) of the parameter block set in the JOG data.
The axis will stop when the JOG start signal is turned OFF.

(1) JOG operation speed setting registers

No.	JOG operation speed setting register		Speed setting range							
			mm		inch		degree		PULSE	
	High-order	Low-order	Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
1	D640	D641	1 to 600000000	$\times 10^{-2}$ mm / min	1 to 600000000	$\times 10^{-3}$ inch / min	1 to 2147483647	$\times 10^{-3}$ / min	1 to 10000000	pulse / s
2	D642	D643								
3	D644	D645								
4	D646	D647								
5	D648	D649								
6	D650	D651								
7	D652	D653								
8	D654	D655								

(2) Forward JOG start signal and reverse JOG start signal

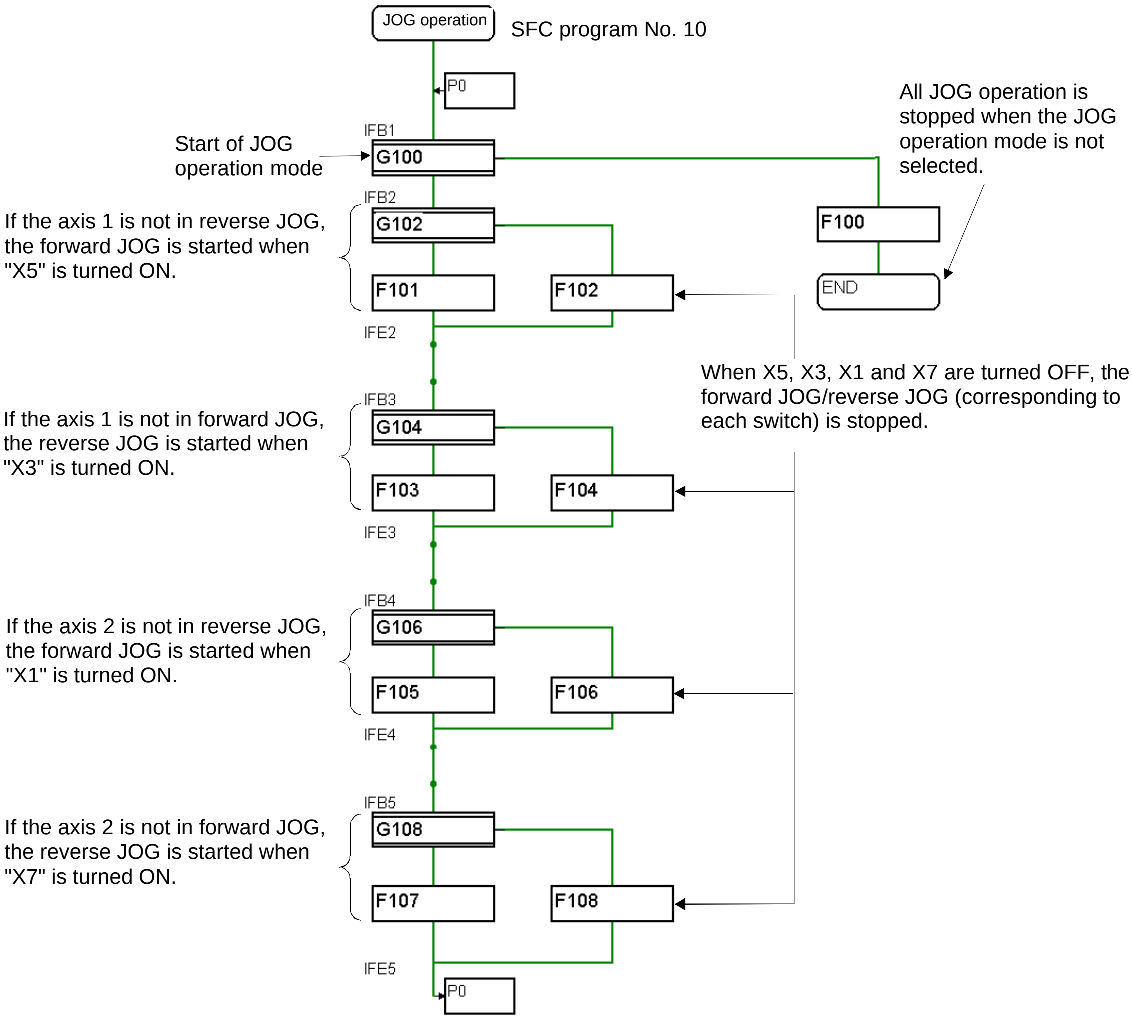
Control axis	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Forward	M3202	M3222	M3242	M3262	M3282	M3302	M3322	M3342
Reverse	M3203	M3223	M3243	M3263	M3283	M3303	M3323	M3343

(3) Program example

1) JOG operation conditions

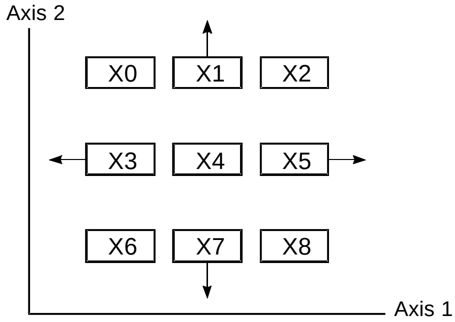
Item	Condition	
Control axis JOG operation speed	Axis 1	Axis 2
JOG operation speed	1000mm/min	1000mm/min
JOG operation command input	Forward (X5)	Forward (X1)
	Reverse (X3)	Reverse (X7)

2) Example of SFC program for JOG operation of axis 1 and axis 2 with independent start

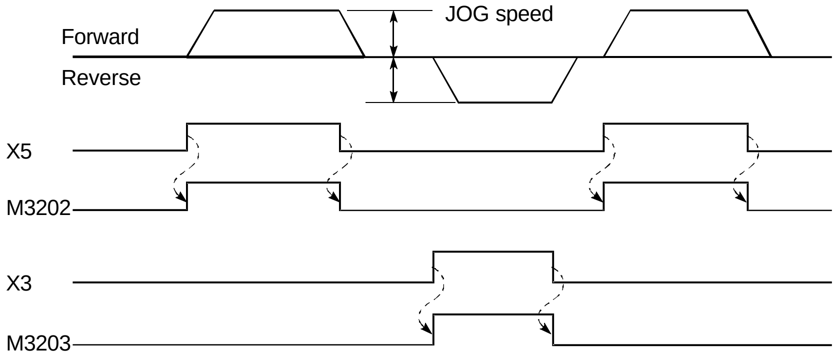


[Transition]	G100 PX18	G102 PX5 * !M3203
	G104 PX3 * !M3202	G106 PX1 * !M3223
	G108 PX7 * !M3222	
[Operation control step]	F100 RST M3202 RST M3203 RST M3222 RST M3223	F101 D640L = K100000 SET M3202
	F102 RST M3202	F103 D640L = K100000 SET M3203
	F104 RST M3203	F105 D642L = K100000 SET M3222
	F106 RST M3222	F107 D642L = K100000 SET M3223
	F108 RST M3223	

PX5 : Axis 1 forward JOG command
PX3 : Axis 1 reverse JOG command
PX1 : Axis 2 forward JOG command
PX7 : Axis 2 reverse JOG command
D641, D640 : Axis 1 JOG speed setting register
D643, D642 : Axis 2 JOG speed setting register



[Timing chart]



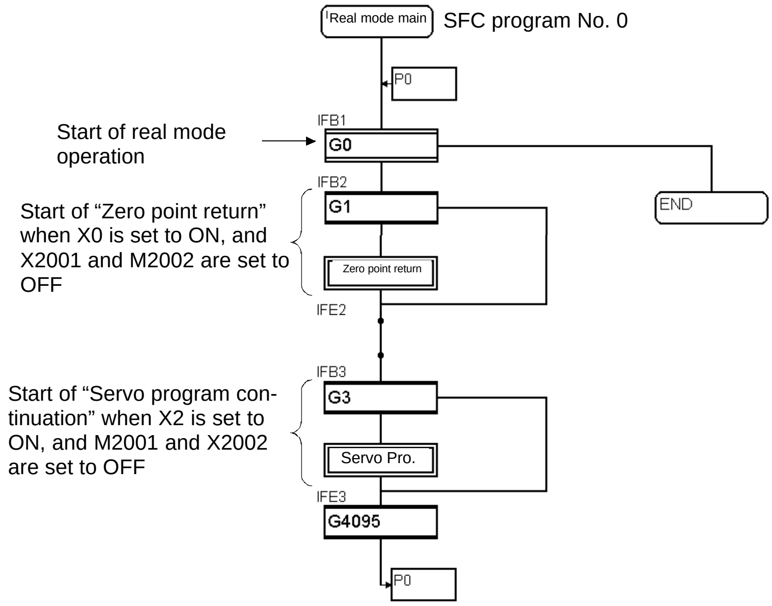
9.9.3 Main routine SFC program (real mode operation)

This SFC program is executed in the main routine when the real mode is selected.
It is used to start other SFC programs (from this main routine SFC program) to execute various operations in the real mode.

(1) SFC program started from main routine SFC program

SFC program No.	Program name	Description section
20	Zero point return	9.9.4
80	Servo program continuation	9.9.5

(2) Program example



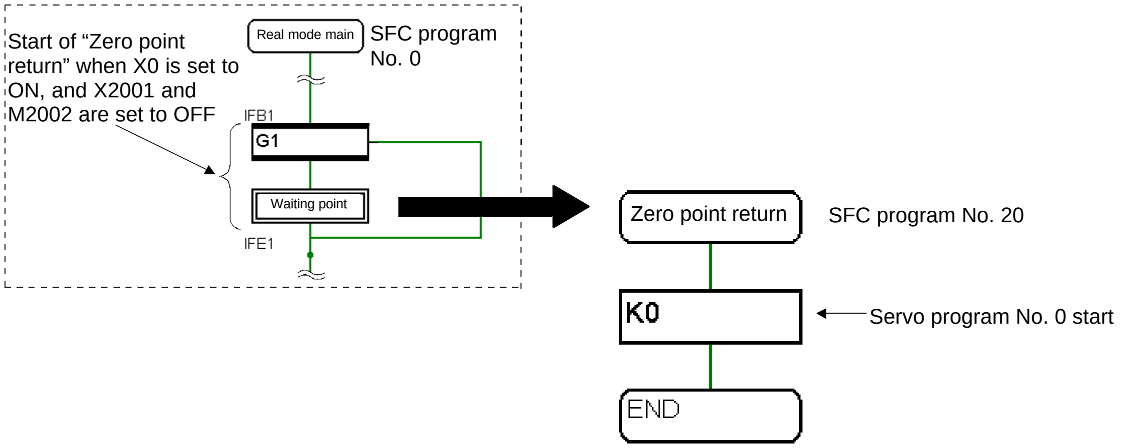
[Transition]	G0 PX19	G1 PX0*!M2001*!M2002
	G3 PX2*!M2001*!M2002	G4095 NOP

M2001 : Axis 1 start accept flag
M2002 : Axis 2 start accept flag
PX0 : Zero point return command
PX2 : Positioning start command

9.9.4 Zero point return

When the servo program is executed at the motion control step of the SFC program, the operation is executed according to the contents of the data and parameter block for the executed servo program.

[SFC program]



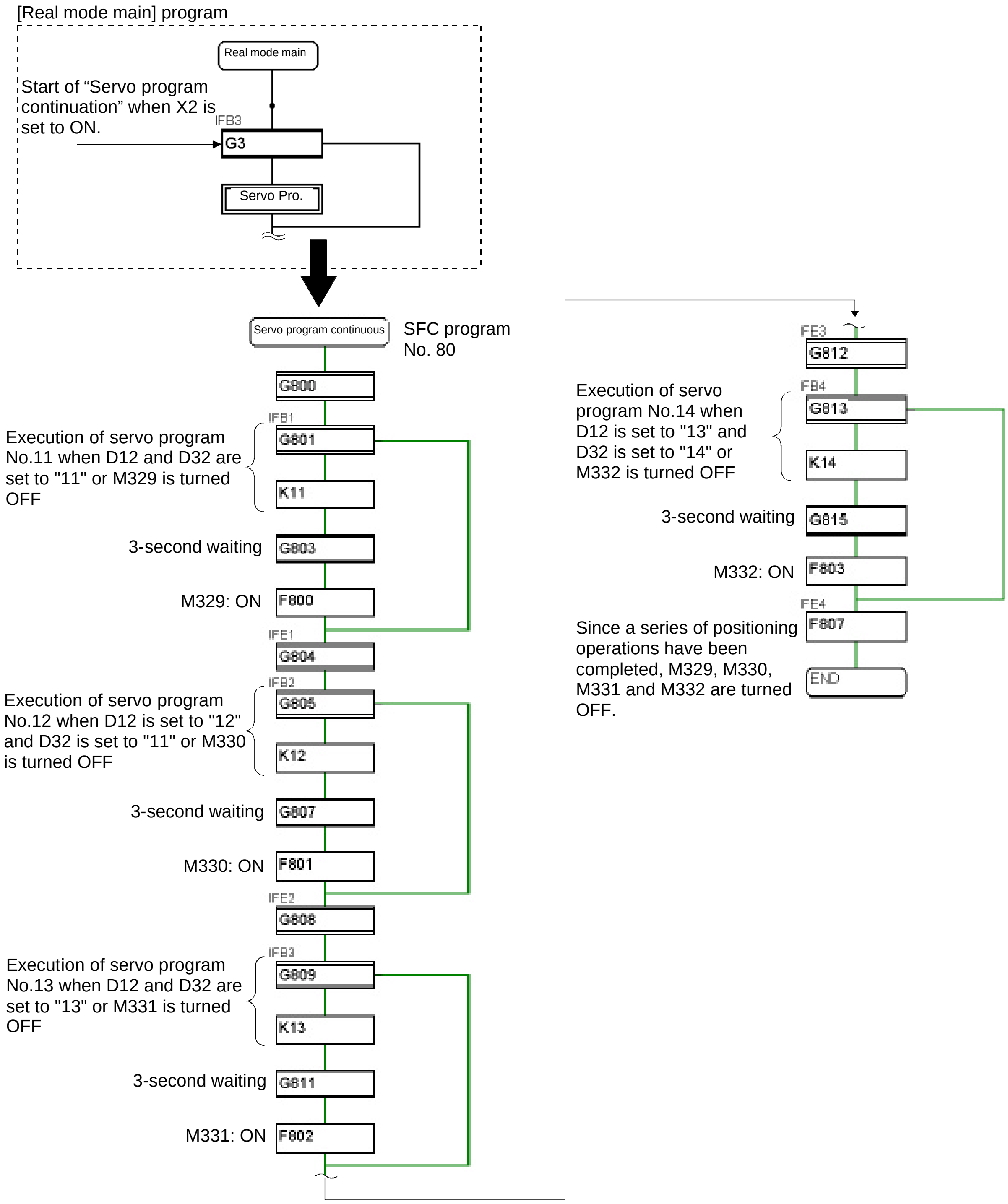
[Motion control step]	K0: REAL 1 START PROGRAM NO. K 1 PROGRAM NO. K 2 PROGRAM NO.	K1: Real 1 ZERO AXIS 1
	K2: Real 1 ZERO AXIS 2	

PX0: Command for zero point return start

9.9.5 Continuous positioning

To execute the servo program in the sequence of 11 → 12 → 13 → 14, the 'WAIT' type transition is used after the motion control step (servo program), so that the execution is shifted to the succeeding motion control step (servo program) after completion of servo program being executed.

When the program is interrupted during continuous execution, the operation is executed continuously from the interrupted servo program to re-start the operation.



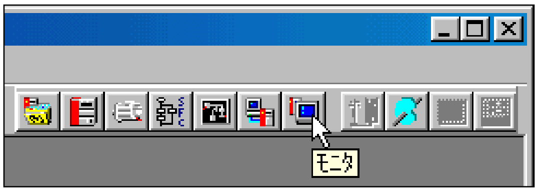
[Transition]	G800 PX19	G801 ((D12 == K11)*(D32 == K11))+ !M329
	G803 TIME K3000	G804 PX19
	G805 ((D12 == K12)*(D32 == K11))+ !M330	G807 TIME K3000
	G808 PX19	G809 ((D12 == K13)*(D32 == K13))+ !M331
	G811 TIME K3000	G812 PX19
	G813 ((D12 == K13)*(D32 == K14))+ !M332	G815 TIME K3000
[Operation control step]	F800 SET M329	F801 SET M330
	F802 SET M331	F807 RST M329 RST M330 RST M331 RST M332
[Motion control step]	K11: REAL 1 ABS-2 AXIS 1, 2000.0µM AXIS 2, 2000.0µM COMPOSITE SPEED 5000.00MM/MIN	K12: REAL 1 ABS-1 AXIS 1, 300000.0µM SPEED 4000.00MM/MIN M CODE 3
	K13: REAL 1 ABS-2 AXIS 1, 20000.0µM AXIS 2, 200000.0µM COMPOSITE SPEED 7000.00 MM/MIN M CODE 3	K14: REAL 1 ABS-1 AXIS 2, 20000.0µM SPEED 4000.00 MM/MIN M CODE 3

D12 : Program No. executed by axis 1
D32 : Program No. executed by axis 2
D329 : Completion of servo program No.11
D330 : Completion of servo program No.12
D331 : Completion of servo program No.13
D332 : Completion of servo program No.14

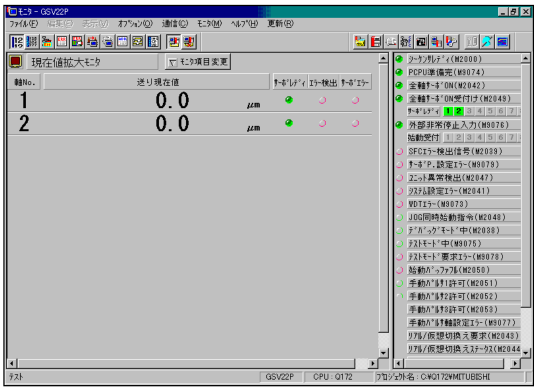
9.10 Operating the practice machine

9.10.1 Operation

The servomotor movement is monitored with the servo monitor using the SW6RN-GSV22P.



1) Click on the monitor tool button on the TEST window.

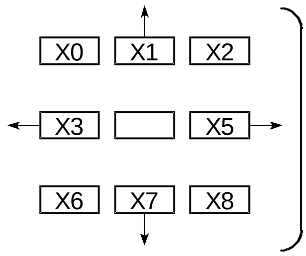


2) The MONITOR window will open and the enlarged current value monitor will appear.
(For details on the monitor operation, refer to section 9.10.2.)

(Continued on next page)

(Continued from previous page)

[Execute JOG operation]



The axis will move with JOG while the switch is ON.

Items to check when axis does not move:

- Is the servo ON?
- Is the Q-PLC/Q motion CPU set to RUN?
- Is the personal computer in the test mode.
  (Cancel the mode if in the test mode.)

[Execute zero point return]

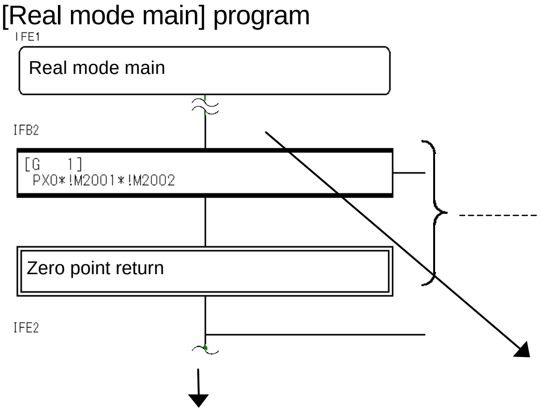
Set the mode selector switch to [REAL] X19.

- Press X0. The X axis and Y axis will return to the zero point.

Note: The motor will not rotate during data-set type zero point return.

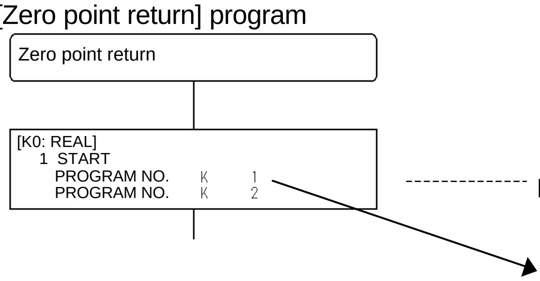
[Real mode main] program

(SFC program No. 0)
(SFC program No. 20)



The SFC program "Zero point return" is started when X0 is turned ON, and M2001 and M2002 are turned OFF.

[Zero point return] program



Execution of servo program No. 0 (simultaneous start)

Axis 1 zero point return

```
[K1: REAL]
  1 ZERO
    AXIS            1
```

The axis 1 current value will change to –30000.0μm.

Axis 2 zero point return

```
[K2: REAL]
  1 ZERO
    AXIS            2
```

The axis 2 current value will change to –30000.0μm.

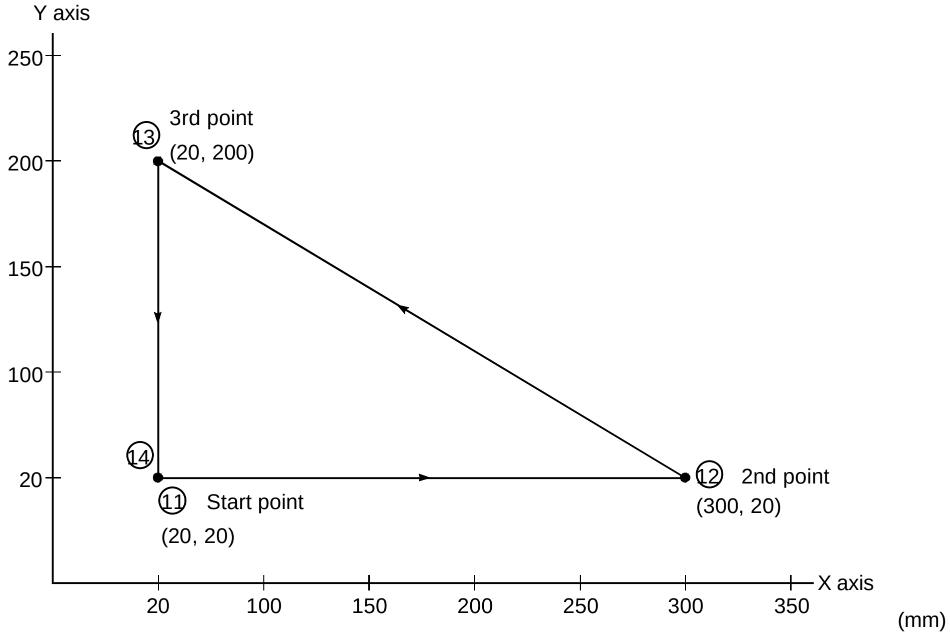
(Continued on next page)

(Continued from previous page)

[Starting continuous positioning]

• Positioning will be carried out with the following path when X2 turns ON.

The circle number indicates the servo program No.



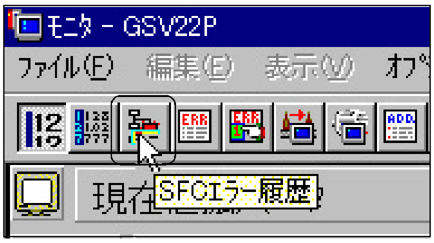
End of operations

9.10.2 Monitor operation

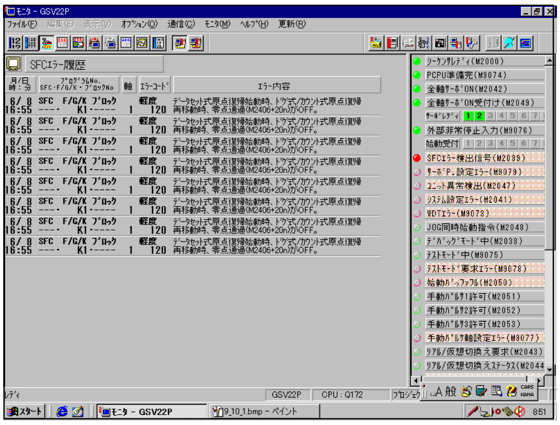
The current value, cause of error occurrence and motion SFC operation status, etc., can be checked with each monitor.

(1) SFC error history

The history of errors that occurred in the SFC program after turning ON or resetting the motion CPU power supply is displayed.

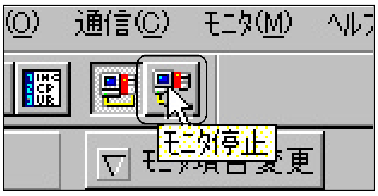


1) Click on the SFC error history button in the MONITOR window.

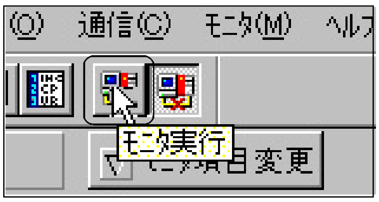


2) The SFC error history monitor will appear.

(2) Starting and stopping the monitor



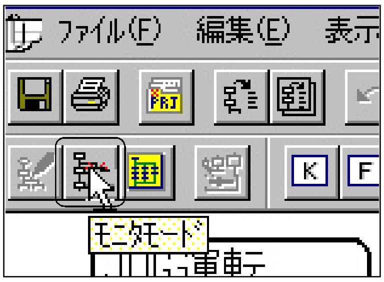
1) To stop the monitor, click on the monitor stop button in the MONITOR window.



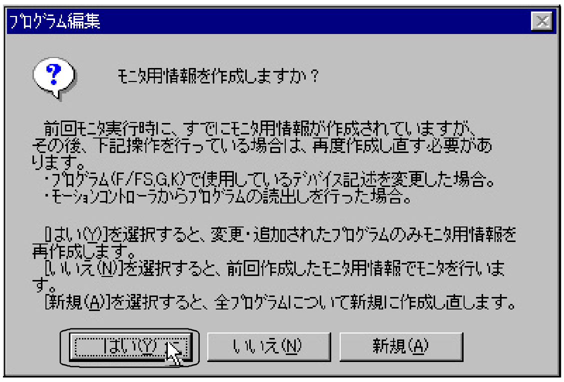
2) To start the monitor again, click on the monitor start button in the MONITOR window.

(3) Motion SFC monitor

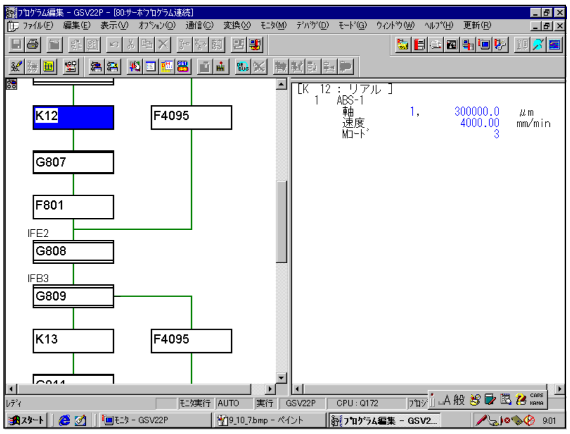
The motion CPU program monitor is displayed.



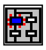
1) Click on the monitor mode button on PROGRAM EDIT screen.



2) When the message shown at left appears, click on the YES button.



3) The motion SFC program monitor will appear.

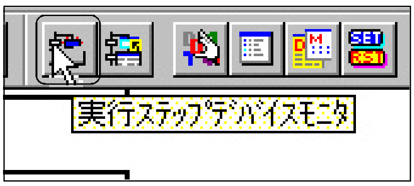
 : Running

 : Stopped

 : In break

(Blue) : Active

(Red) : Waiting for parallel connection

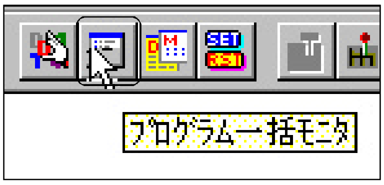


4) Click on the execution step device monitor button.

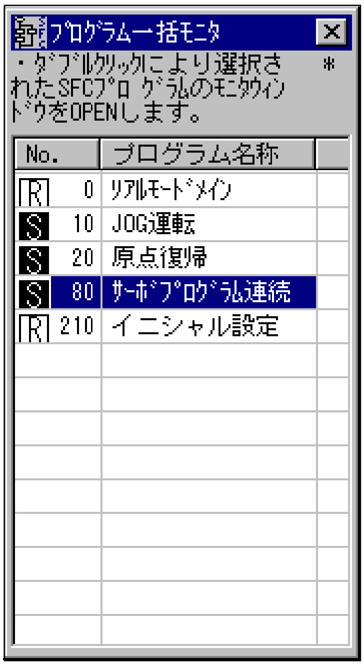
(Continued on next page)

(Continued from previous page)

5) The EXECUTION STEP DEVICE MONITOR window will open. The state of the active step's device can be confirmed.



6) To change the displayed program, click on the PROGRAM BATCH MONITOR button.



7) The PROGRAM BATCH MONITOR EXECUTION STEP window will open.
When the program to monitor is double-clicked on, the window for the selected SFC program will open.

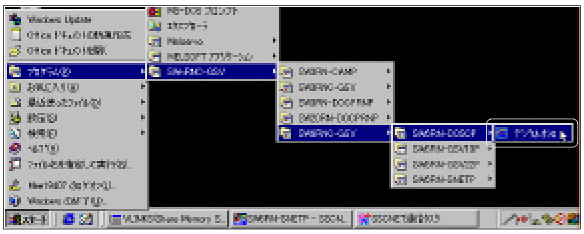
9.10.3 Monitor trace graph

The position command, position droop, motor speed, motor current, and speed command, etc., can be traced with the SW6RN-DOSCP's digital oscilloscope.

Each method differs as shown in the following table according to the communication connection method. The method using an RS-232C connection is practiced in this section.

Items	PC real-time read method	Motion buffering method	
	SSCNET communication	SSCNET communication	RS-232C/USB communication
Communication method	High-speed refresh communication	Transient communication	
Number of sampling points	8,192/data (max.) (default value: 8192/data)	8,192/data (max.) (default value: 2048/data)	
Sampling cycle	3.5 × Nmsec (1 ≤ N ≤ 65535) 3.5msec (min.), 233.0sec (max.)	0.888 × Nmsec (1 ≤ N ≤ 65535) 0.888msec (min.), 58.253sec (max.)	
Display of waveform during RUN	○	×	
Temporary stop of screen during RUN	○	×	
Zoom-enlargement of screen during RUN	○	×	
Waveform display method	Scroll method to display updated data	Data is read in a batch after STOP and displayed in a waveform.	
Trigger function (filter designation)	Filter time value: 3.5 × Nmsec (N: 0 to 65535)	Filter time value: 0.888 × Nmsec (N: 0 to 65535)	
Buffering method for sampling data	Data sent from motion CPU is buffered every 3,555ms at PC side.	Data is buffered at motion CPU side until STOP is requested or trigger is established.	
Judgment of trigger condition for establishment	Judgment at PC side	Judgment at motion CPU side	
Stop/interrupt during buffering	When the STOP is clicked, the buffering is stopped. (It is unnecessary to interrupt the buffering.)	STOP (stop) • The buffering stop request is sent, and the buffered data is read in a batch. INTERRUPT • While buffering is executing The buffering stop request is sent, but the data is not read. • While the data is reading in a batch The data reading is interrupted, and the waveform is not displayed. • The file is not saved. (Automatic saving by establishment of trigger)	
IRQ setting at SNETP	Required	Required	Not required

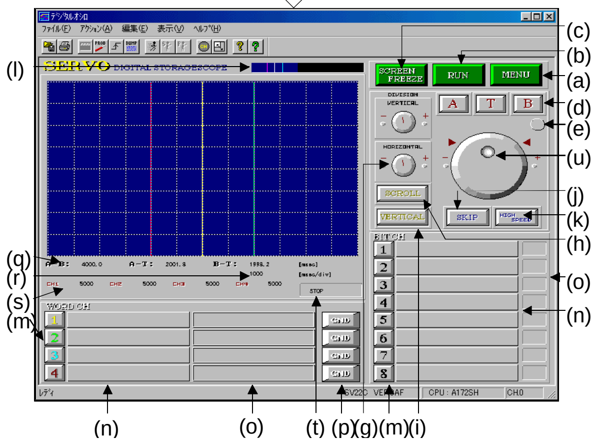
(1) Communication setting



(Continued on next page)

1) Click on [Start], [Program], [SWnRNC-GSV], [SW6RN-GSV], [SW6RN-DOSCP] and then [Digital oscilloscope] in Windows.

(Continued from previous page)



2) The DIGITAL OSCILLOSCOPE window will open.

No.	Name	Function
(a)	MENU button	Displays the shift buttons (PROB, TRIGGER, PRINTER, DUMP, FILE) to each screen, CH No. selection, and application screen changeover button.
(b)	RUN button	Starts/stops the monitor.
(c)	SCREEN FREEZE button	Stops the screen temporarily.
(d)	Monitor cursor selection button	Selects the monitor cursor to move from monitor cursors (A, T, B3). When held down for more than three sec., the monitor cursor of the selected button will move to the center of the graph.
(e)	Monitor cursor display button	Shows or hides the monitor cursor.
(f)	VERTICAL DIVISON dial	Adjusts the vertical axis range for the selected channel.
(g)	HORIZONTAL DIVISION dial	Adjusts the time axis range.
(h)	SCROOL button	Enables the horizontal scroll.
(i)	VERTICAL button	Enables the vertical scroll for the selected channel.
(j)	SKIP button	Validates/invalidates the movement of monitor cursor to waveform change point.
(k)	HIGH SPEED button	Changes the monitor cursor movement unit/screen scroll unit.
(l)	Display area MAP	Displays the monitor cursor position within sampling area.
(m)	Channel selection button	Designates the channel for adjusting the GND movement/vertical scroll/vertical axis range.
(n)	Monitor data display column	Displays the monitor data item name. (For monitor data corresponding to axis, the axis No. is displayed before monitor data item name.) Changes the display/non-display of waveform at ON/OFF of display column.
(o)	Sampling value display column	Displays the sampling data value for monitor cursor position designated last.
(p)	GND button	Validates/invalidates the GND display. "GND" is a reference position on display. When the oscilloscope is started, the GND is set initially to "0". If the GND button held down for over three seconds, the waveform sampled is displayed so that the GND position is at the center of vibration. The vibration is also adjusted automatically to a readily visible size.
(q)	Monitor cursor time interval display column	Displays the time width of each monitor cursor (A/T/B).
(r)	Time range display column	Displays the time range.
(s)	Vertical axis display column	Displays the vertical axis range.
(t)	Operation status display column	Displays the current operation status (STOP/MONITOR/TRIGGER/FREEZE).

(Continued on next page)

(Continued from previous page)

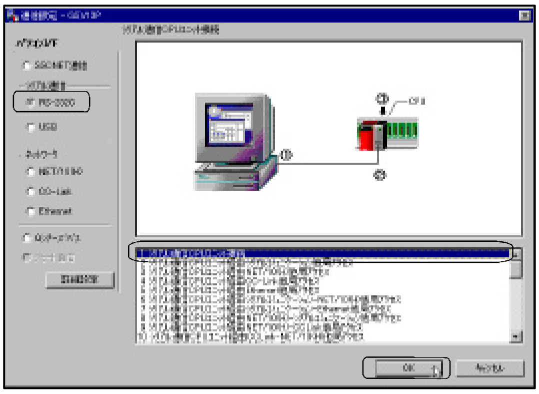
<table>
<tr><th>No.</th><th>Name</th><th>Function</th></tr>
<tr><td>(u)</td><td>JOG dial</td><td>Adjusts the monitor cursor position/waveform display position when the monitor cursor is moved or the waveform display is scrolled.
The function may change depending on the combination of following push-buttons pressed.
<table>
<tr><td rowspan="4">HIGH SPEED button ON</td><td colspan="2">SCROLL button ON</td><td>Scrolls the screen every 6 grids horizontally.</td></tr>
<tr><td>SCROLL button OFF</td><td>VERTICAL button ON</td><td>Scrolls the designated channel every grid vertically.</td></tr>
<tr><td rowspan="2">Monitor cursor selection button ON</td><td>SKIP button ON</td><td>Moves the monitor cursor to the waveform change point each time when the SKIP button is pressed 10 times.</td></tr>
<tr><td>SKIP button OFF</td><td>Moves the monitor cursor every grid to the waveform change point.</td></tr>
<tr><td rowspan="4">HIGH SPEED button OFF</td><td colspan="2">SCROLL button ON</td><td>Scrolls the screen every grid horizontally.</td></tr>
<tr><td>SCROLL button OFF</td><td>VERTICAL button ON</td><td>Scrolls the designated channel every sampling data vertically.</td></tr>
<tr><td rowspan="2">Monitor cursor selection button ON</td><td>SKIP button ON</td><td>Moves the monitor cursor to the waveform change point.</td></tr>
<tr><td>SKIP button OFF</td><td>Moves the monitor cursor every sampling data.</td></tr>
</table>
</td></tr>
</table>

3) Click on [EDIT] and then the [CHANNEL] menu on DIGITAL OSCILLOSCOPE window to set the communication items.

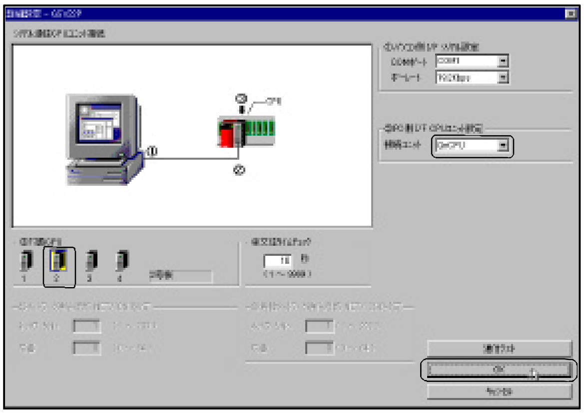
4) The DIGITAL OSCILLOSCOPE COMMUNICATION SETTING dialog box will open, so set the "Sampling method" to the motion buffering method.
After setting, click on the $\boxed{\text{Connection destination designation}}$ button.

(Continued on next page)

(Continued from previous page)



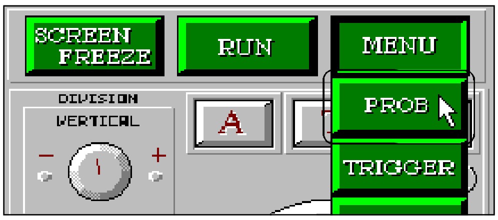
5) The COMMUNICATION SETTING dialog box will open, so check "RS-232C", select "1. Serial communication CPU connection" and click on the Detail setting button.



6) The DETAIL SETTING dialog box will open, so set "PC side I/F CPU setting" to 'QnCPU', and "Object CPU" to '#2 machine'.
After setting, click on the OK button.
Since the communication setting dialog box is resumed in this case, click on the OK button.

7) When the message shown at left is displayed, click on the OK button.

(2) Waveform measurement

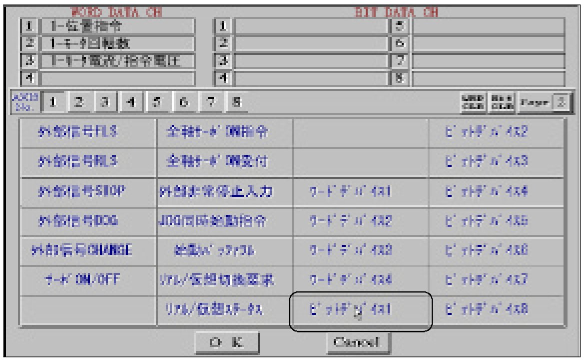


1) Select the item to trace.
Click on the MENU button, and then the PROB button.

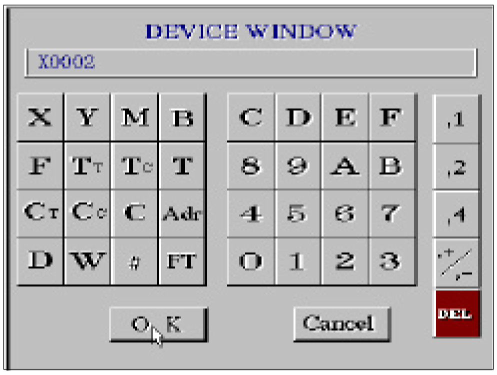
(Continued on next page)

(Continued from previous page)

2) Click on 'Motor current/command voltage', 'Motor speed' and 'Position command', and then click on the Page1 button.



3) Click on 'Bit device 1'.

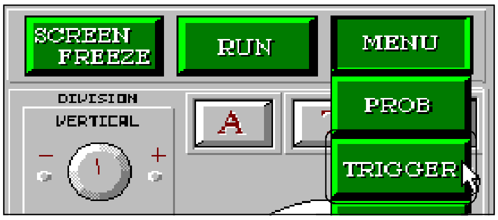


4) Click on 'X2' using the alphanumeric button arranged on the device window, and click on the OK button.

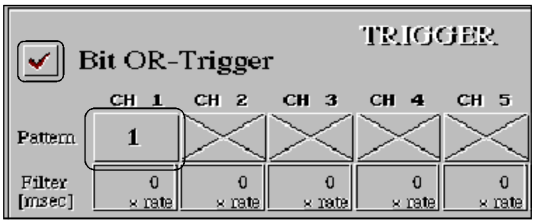
5) Click on the OK button.

(Continued on next page)

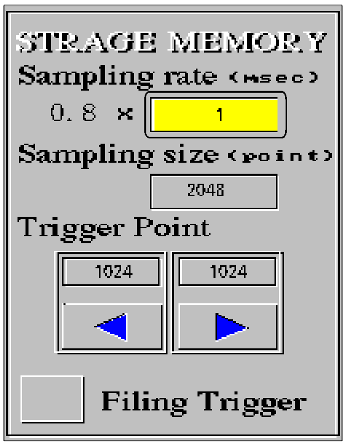
(Continued from previous page)



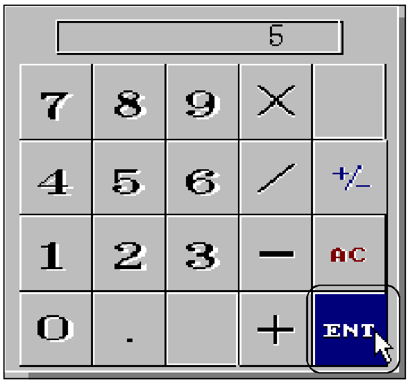
6) Set the trace conditions.
Click on the MENU button, and then the TRIGGER button.



7) Check "Bit OR-Trigger".
Click on "Pattern" of CH 1 twice.
('1' (ON status) will appear.)



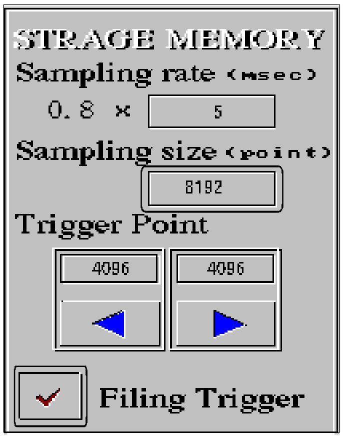
8) Set the sampling unit to 4.0msec.
Click on and select the setting value '1' for "Sampling rate" on "STRAGE MEMORY".
(When selected, the display color will change to yellow.)



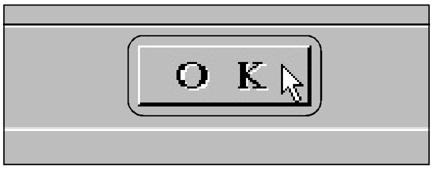
9) Click on the '5' and then 'ENT' using the value input button. (When the AC button is clicked on, the value will be cleared.)
With the above step, the "Sampling rate" will be set to '5'.

(Continued on next page)

(Continued from previous page)



10) Set the sampling size to 8,192 times.
Set "Sampling size (point)" to '8192' in the same manner as steps 8) to 10), and the check "Filing Trigger".



11) Click on the OK button.

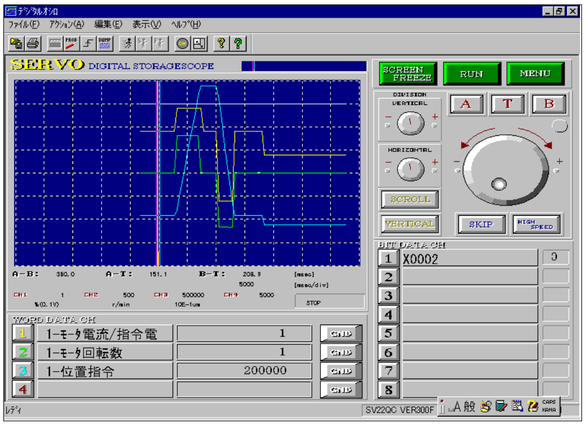
12) Click on the RUN button.
The trace monitor will be executed.

13) The trigger waiting status will occur, causing "Sampling before trigger" to appear at the display area MAP.

14) Turn ON X2 of the practice machine.

(Continued from previous page)

15) When buffering is completed after establishment of the trigger, the Buffering data reading progress bar will appear.
The waveform will appear when the buffering data is read.

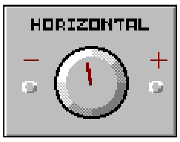


16) The displayed graph can be enlarged or reduced and confirmed with the steps given on the next page.

(Continued on next page)

[Enlargement/reduction of graph in horizontal direction]

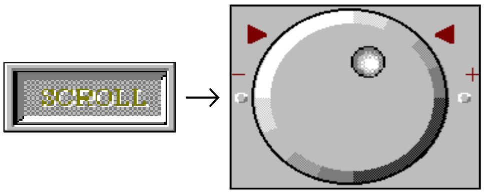
Adjust the time axis range.



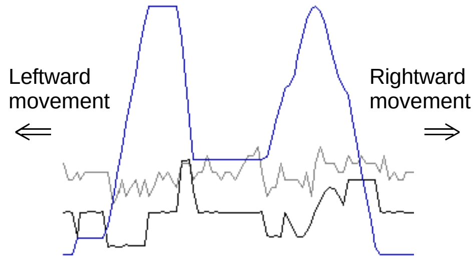
HORIZONTAL

[Movement of graph in horizontal direction]

Allows the graph to be scrolled horizontally.

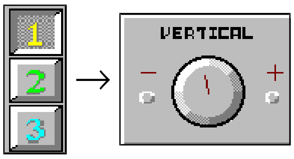


SCROLL

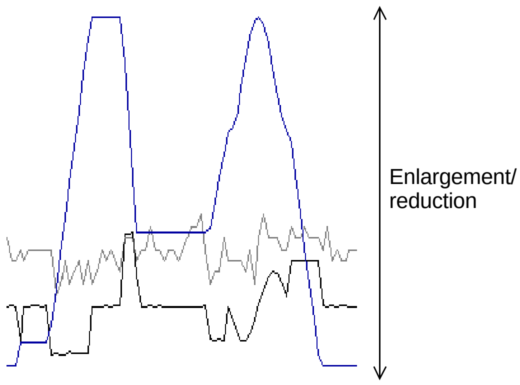


[Enlargement/reduction of graph in vertical direction]

Adjust the vertical axis range of selected graph.



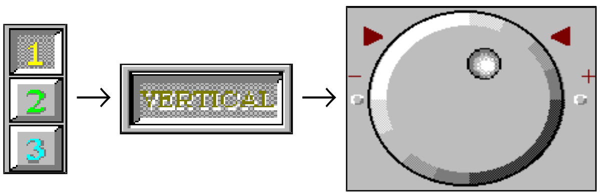
VERTICAL



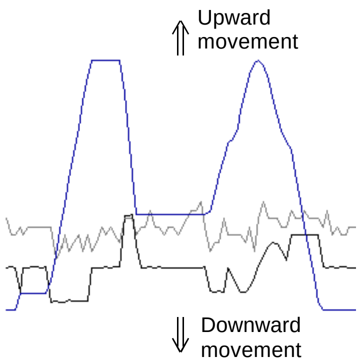
Note: Enlarge/reduce only the graph for the designated data No. in the vertical direction.

[Movement of graph in vertical direction]

Allows the selected graph to be scrolled vertically.



VERTICAL

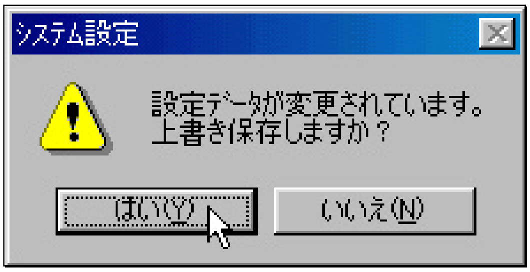


Note: Only the graph for the designated data No. is moved in the vertical direction.

9.11 Ending the operations

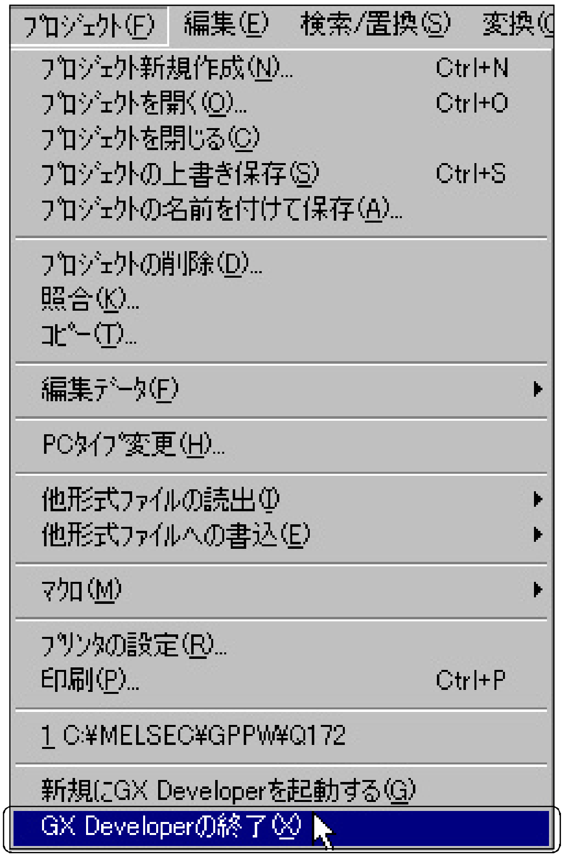
9.11.1 Ending the SW6RN-GSV22P operations

1) Click on [File], and then the [GSV22P End] menu on the TEST window.
(The [File] menu and [GSV22P Exit] menu are located on the window of each function.)



2) If the setting data is not saved, the message to confirm overwriting of the data will appear. Click on the YES button.
The message for system setting is shown on the left.

9.11.2 GX Developer END operation



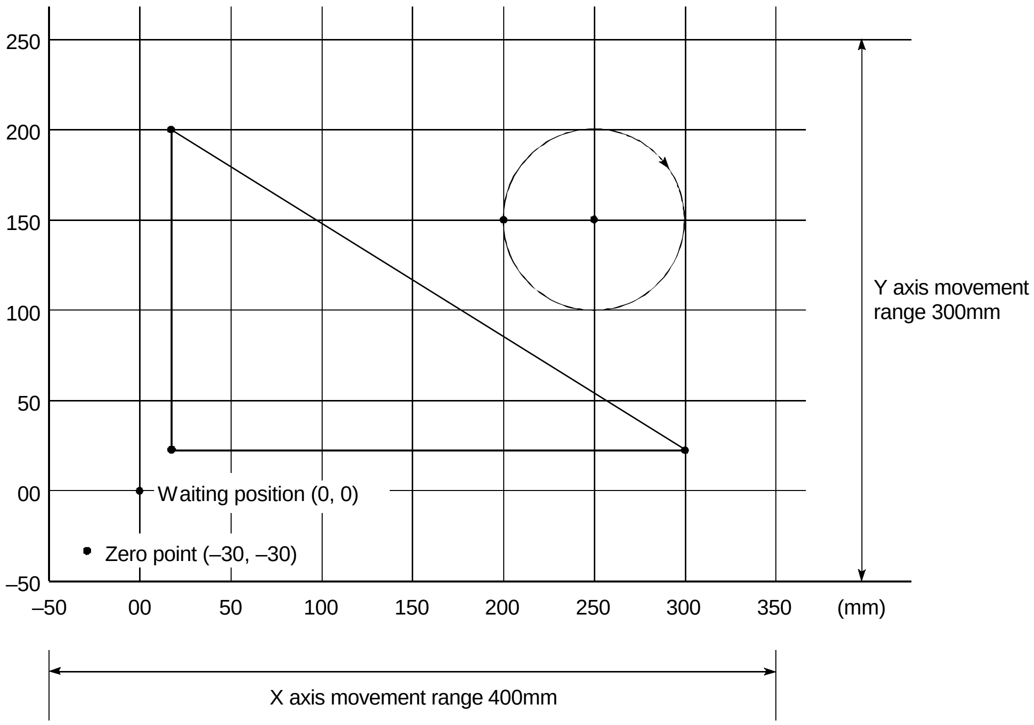
1) Click on [Project] and then the [GX Developer Exit] menu in GX Developer.

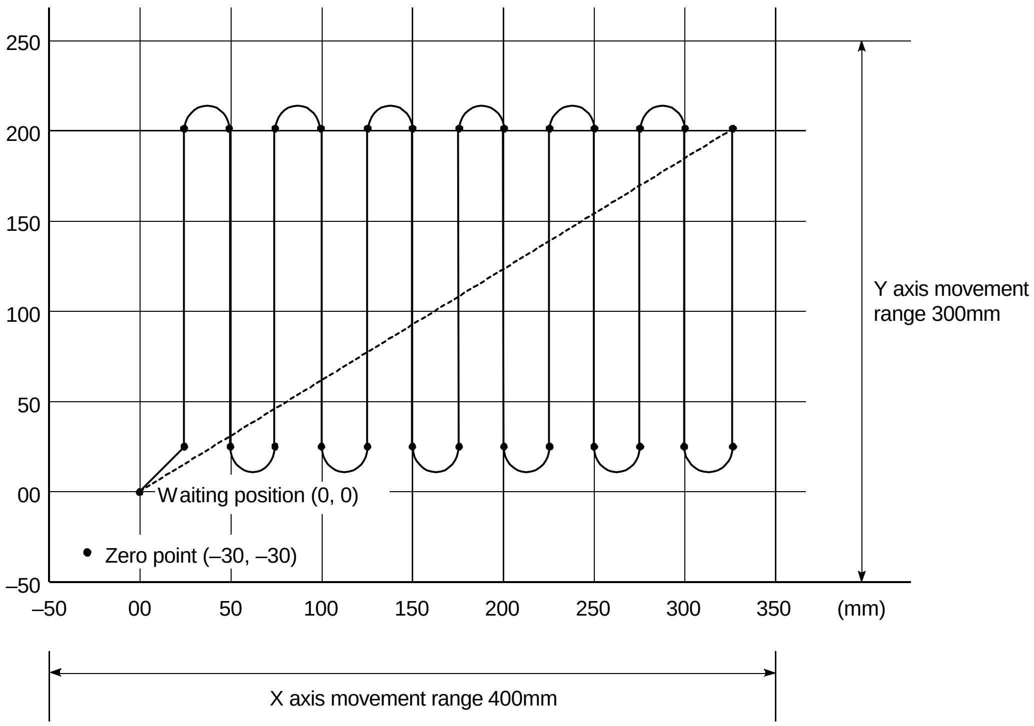
2) If the setting data is not saved, the message to confirm saving will appear. Click on the YES button.

Chapter 10 Applied Practice with SV22 Real Mode

10.1 Details of practice

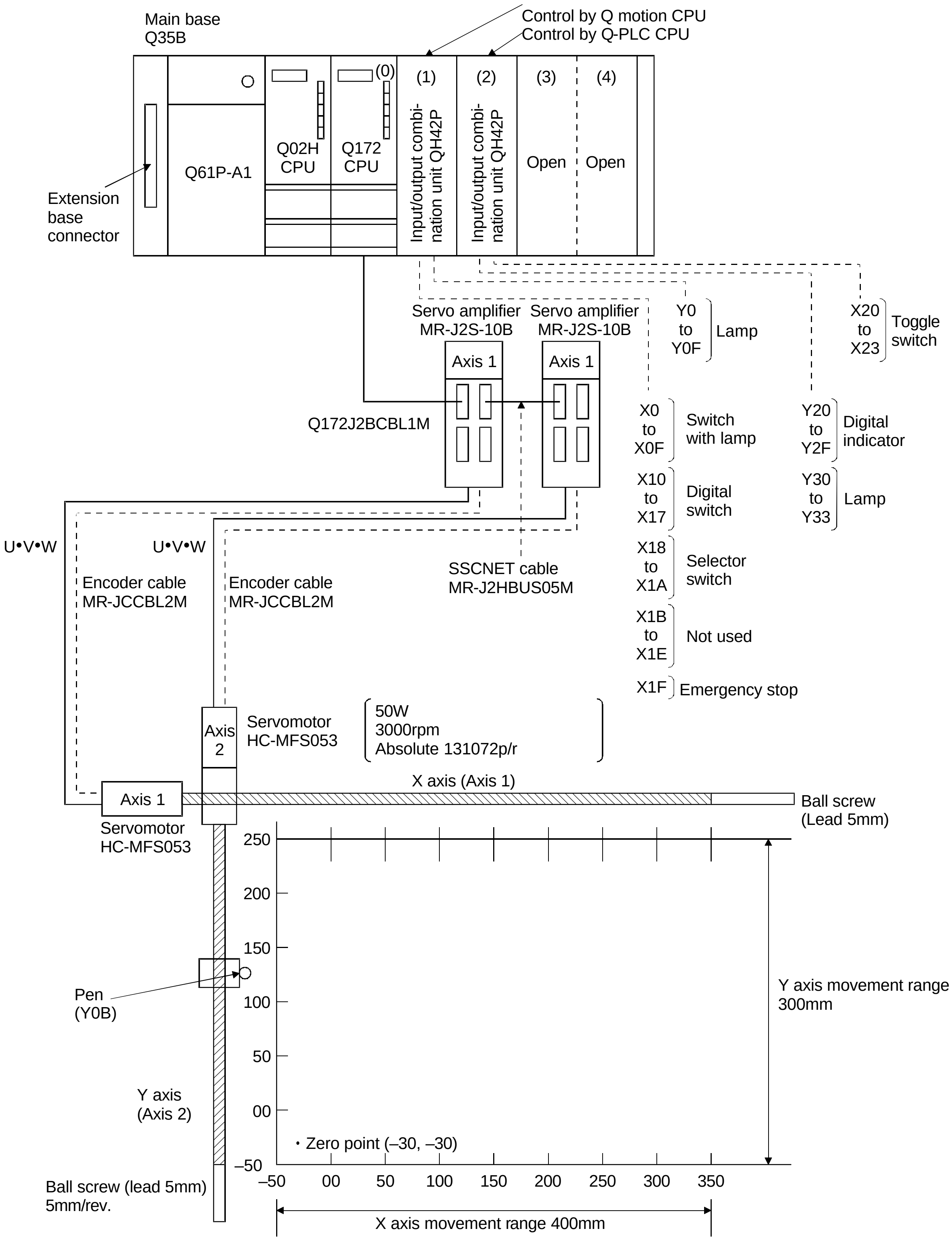
Practice drawing triangles and circles as positioning paths on the X-Y table, and practice uniform speed control and speed control.
Since SV13 is the same as real mode applied to the SV22, this practice is used in common.





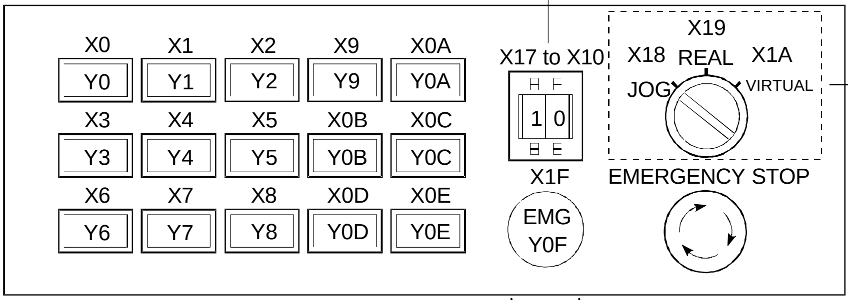
10.2 Q172CPU practice machine system configuration

Since the external signals (limit, DOC) are not used for this practice, the Q1272LX unit is omitted.



Practice machine operation panel

The function selector switches are wired to X10 to X17.



The digital switches are wired to X18 to X1A.

The input signal switches are wired to X0 to X0E.
The lamps are wired to Y0 to Y0E.

The basic settings for the Q motion CPU are set.
The lamp is wired to Y0F.

X0	X1	X2	X9	X0A
STANDBY POINT POSITION-ING	SET NO. POSITION-ING	CONTINUE POSITION-ING		VIRTUAL MODE PRSENT VALUE
X3	**X4**	**X5**	**X0B**	**X0C**
SPEED CHANGE 2000	SPEED CHANGE 1000	SPEED CHANGE 300	PEN UP/DOWN	ERROR RESET
X6	**X7**	**X8**	**X0D**	**X0E**
CAM DATA CHANGE REQUEST		CLUTCH OFF	STOP	SUDDEN STOP

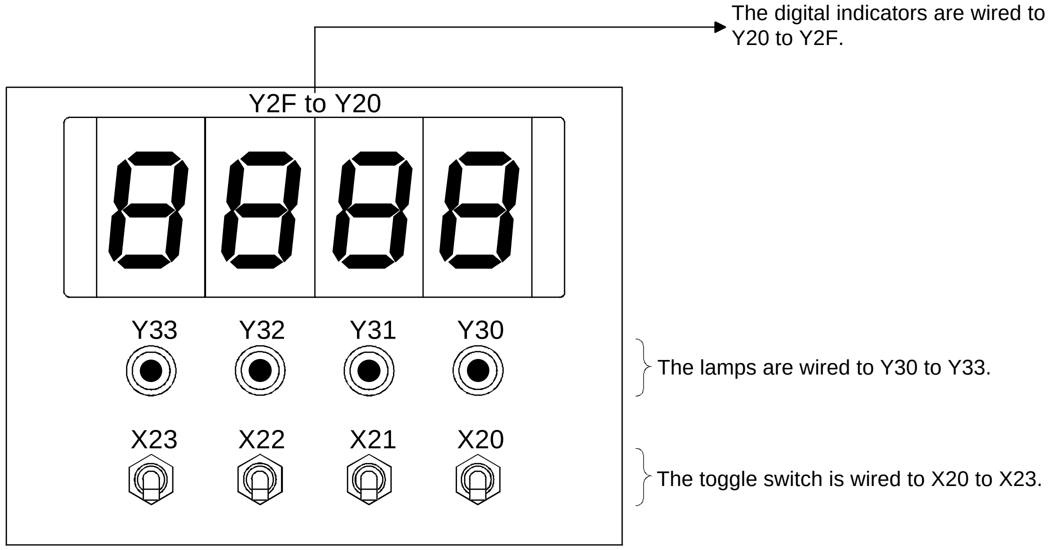
X10 to X17 — DIGITAL switch 2 digit

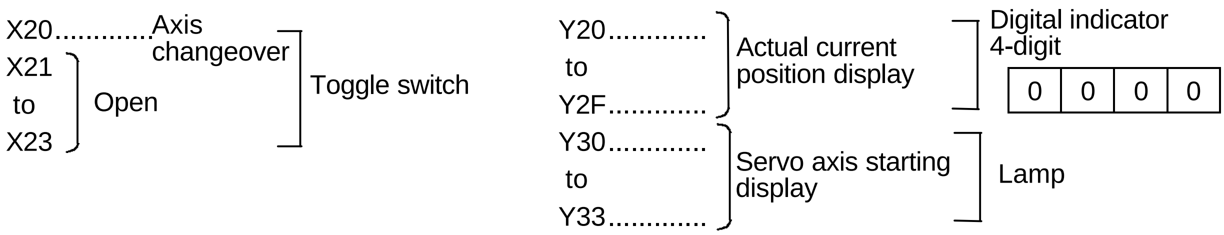
0	0

X18............JOG
X19............Real
X1A............Virtual — Operation selector switch
X1F............Emergency stop

X0 ....
X1 ....
X2 ....
X3 ....
X4 ....
X5 ....
X6 ....
X7 ....
X8 .... — JOG operation Positioning start Speed change command
X9 ....Current value change
X0A............Open
X0B............Pen UP/DOWN
X0C............Error reset
X0D............Stop
X0E............Sudden stop

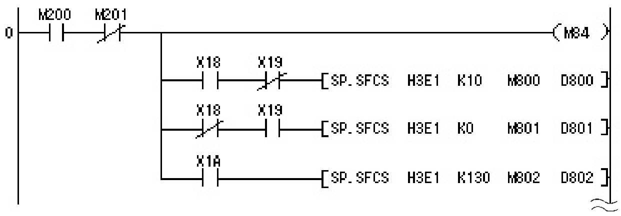
(X0 to X0E: Push-button switch)


The digital indicators are wired to Y20 to Y2F.
Y2F to Y20
Y33
Y32
Y31
Y30
The lamps are wired to Y30 to Y33.
X23
X22
X21
X20
The toggle switch is wired to X20 to X23.


X20............Axis changeover
X21
to
X23
Open
Toggle switch
Y20............
to
Y2F............
Actual current position display
Digital indicator 4-digit
0
0
0
0
Y30............
to
Y33............
Servo axis starting display
Lamp

10.3 Practice SFC programs

The sequence programs and SFC programs used for practice are listed below. Refer to the following explanations for details on each program.

Normal execution	Start by sequence program	Start by SFC program
• Sequence program  • [Start/sudden stop] SFC program No. 40 • [Speed change] SFC program No. 60 • [Actual current value read] SFC program No. 70 • [M code read] SFC program No. 90 • [Error detection_Reset_EMG] SFC program No. 110 • [Push-button] SFC program No. 120 • [Initial setting] SFC program No. 210	• [JOG operation] SFC program No.10 • [Real mode main] SFC program No.0	Automatic start • [Waiting point positioning] SFC program No. 20 • [Servo program execution] SFC program No. 30 • [Current value change] SFC program No. 60 • [Servo program continuation] SFC program No. 80 • [Address indirect designation] SFC program No. 100 • Virtual mode for practice

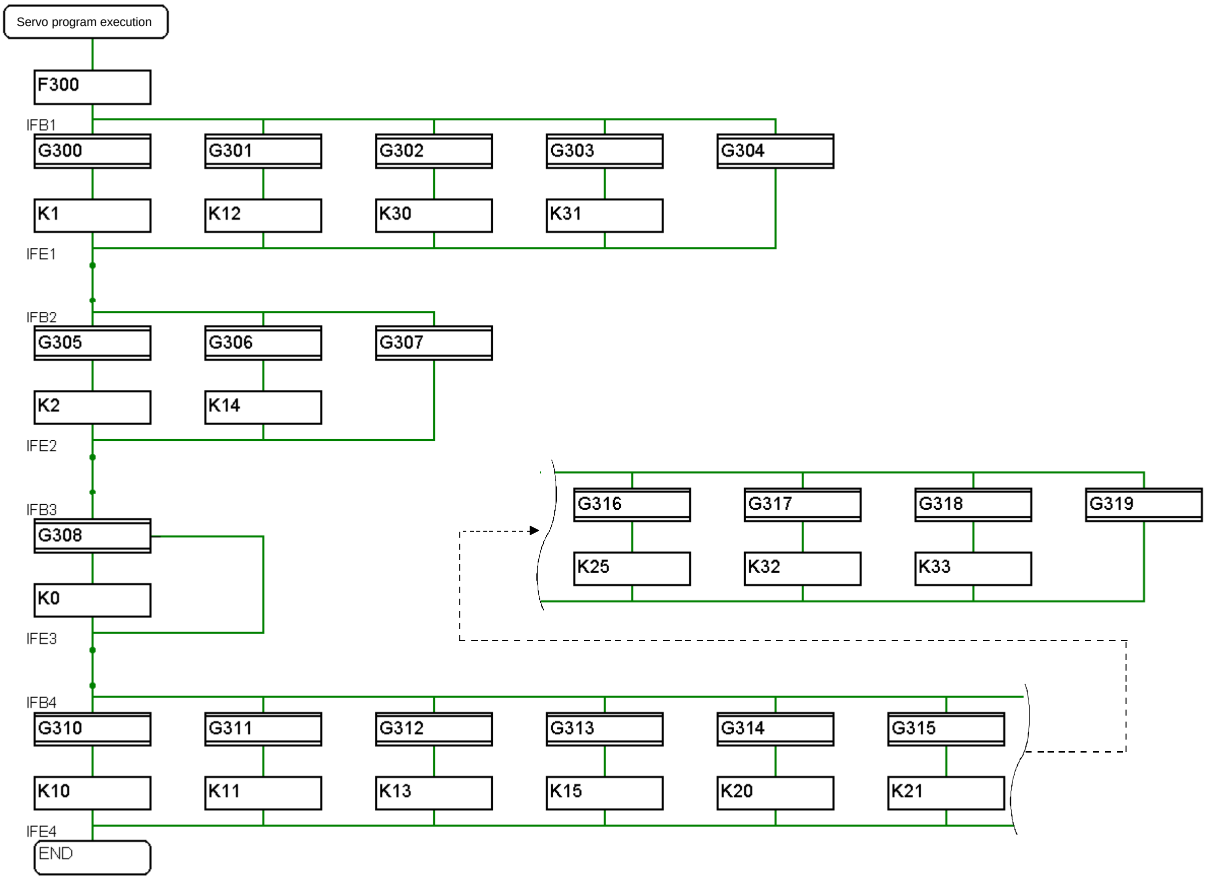
SFC program parameters

No.	Program name	AUTO start	END operation	Number of shifts	Execution timing
0	Real mode main	No			Normal
10	JOG operation	No			Normal
20	Waiting point positioning	No			Normal
30	Servo program execution	No			Normal
40	Stop/sudden stop	Yes			Normal
50	Current value change	No			Normal
60	Speed change	Yes			Normal
70	Actual current value read	Yes			Normal
80	Servo program continuation	No			Normal
90	M code read	Yes			Normal
100	Address indirect designation	No			Normal
110	Error detection_Reset_EMG	Yes			Normal
120	Push-button	Yes			Normal
210	Initial setting	Yes			Normal

• **Program started by SFC program (1)**

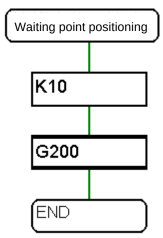
[Servo program execution] program No. 30

Started by No. 0



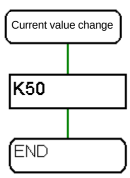
[Waiting point positioning] program No. 20

Start by No. 0



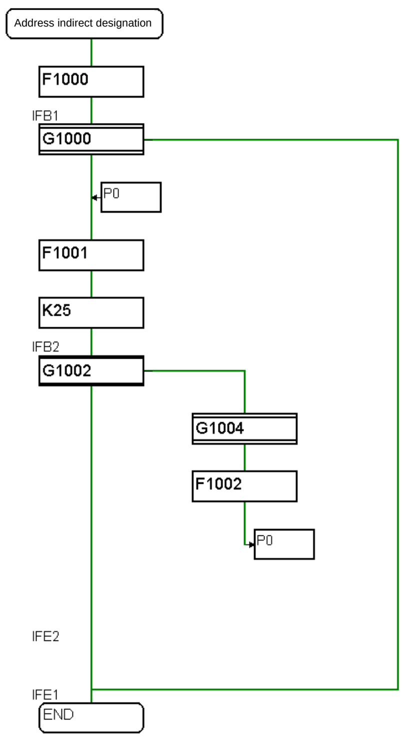
[Current value change] program No. 50

Start by No. 0



[Address indirect designation] program No. 100

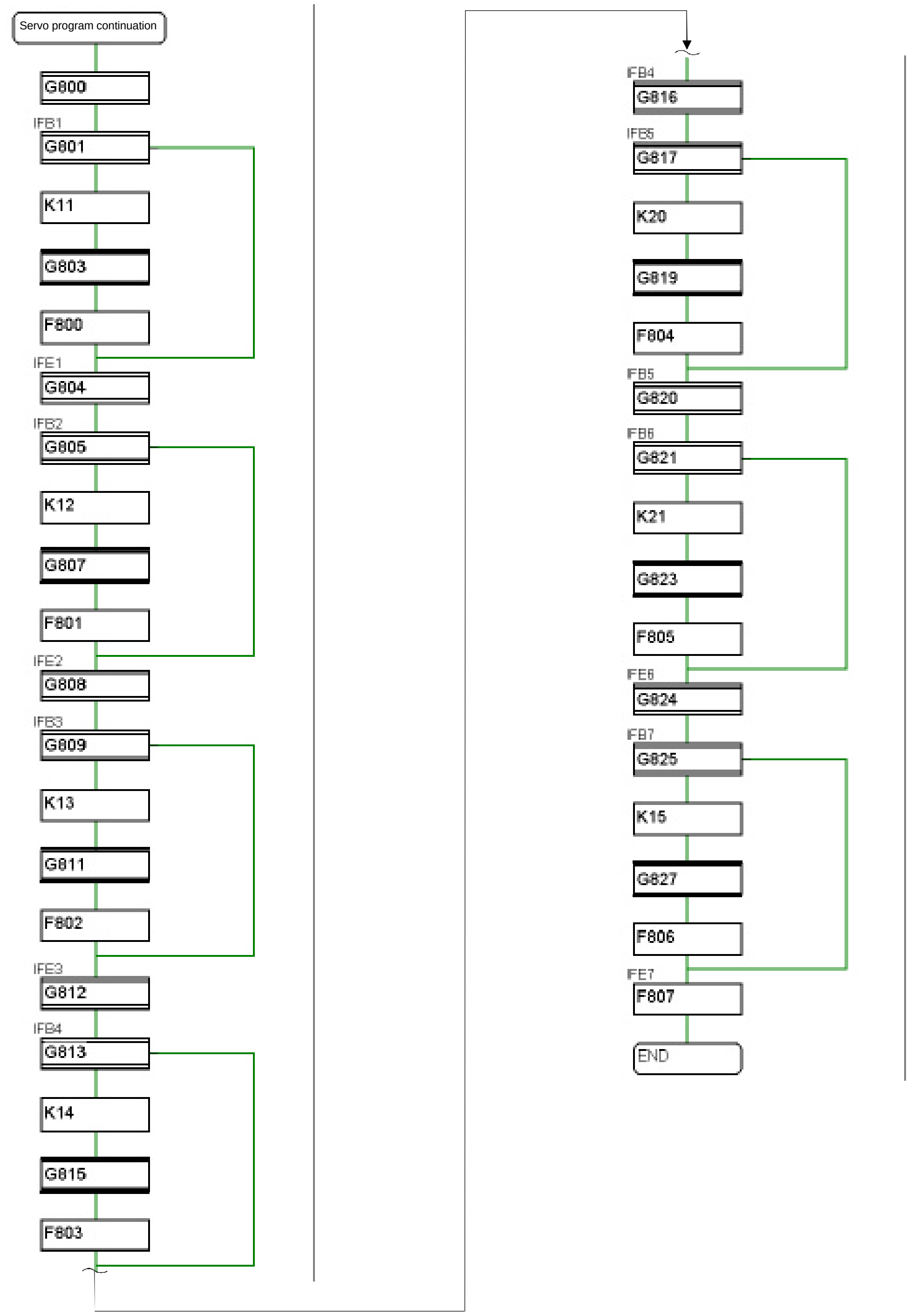
Start No. 0



• **Program started by SFC program (2)**

[Servo program continuation] program No. 80

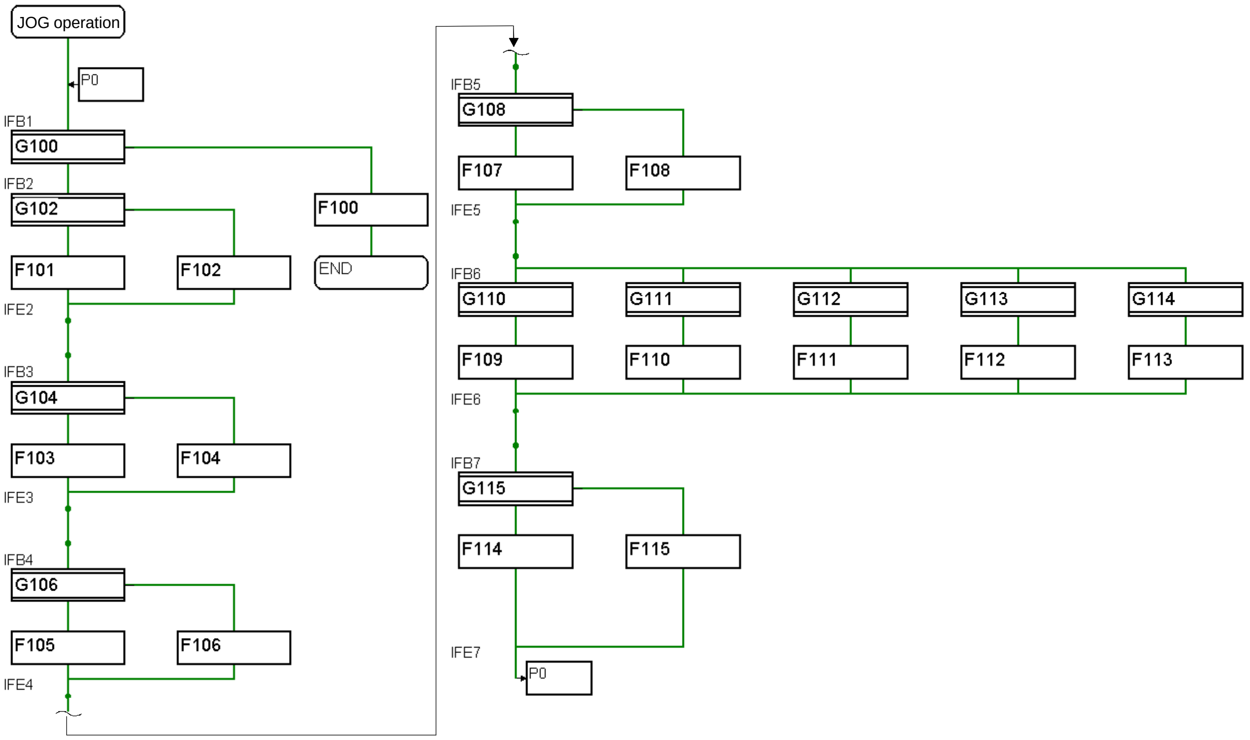
Started by No. 0



• **Program started by sequence program**

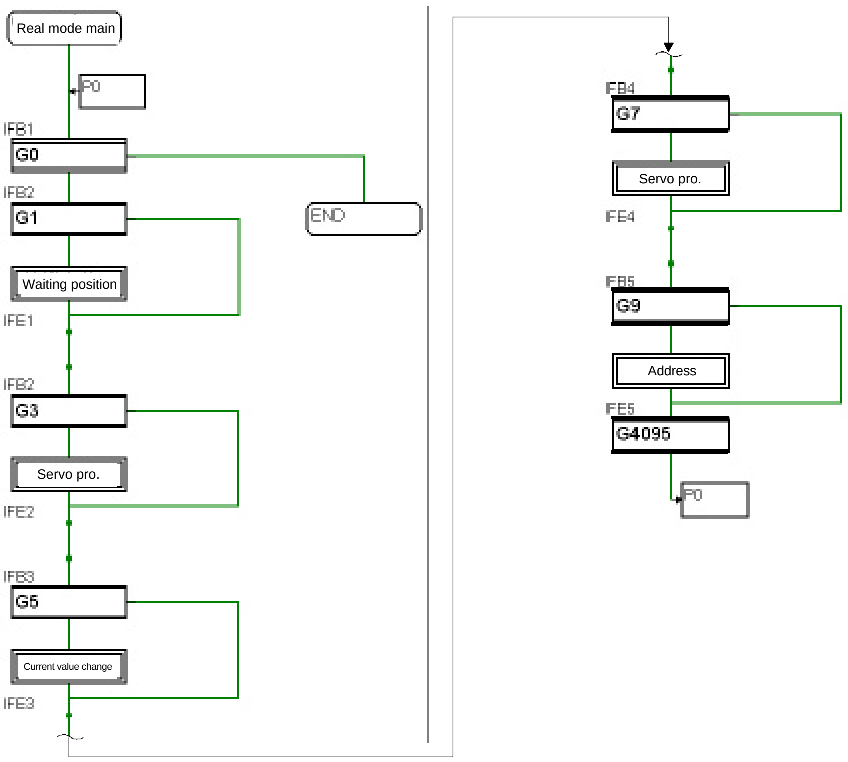
[JOG operation] program No. 10

Started by sequence program



[Real mode main] program No. 0

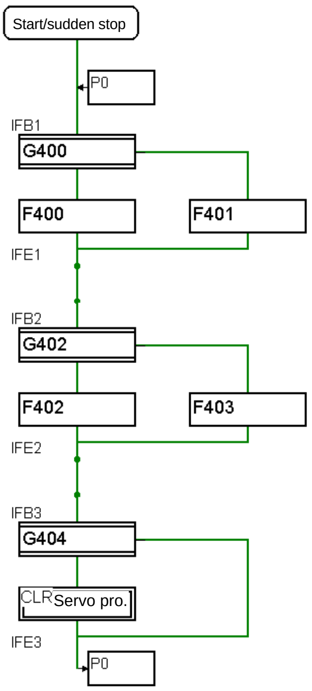
Started by sequence program



- **Normal execution program**

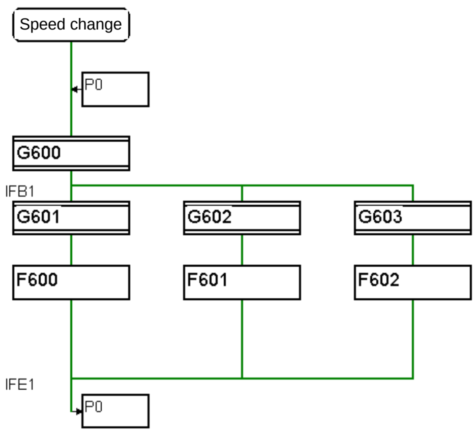
[Start/sudden stop] program No.40

Started automatically.



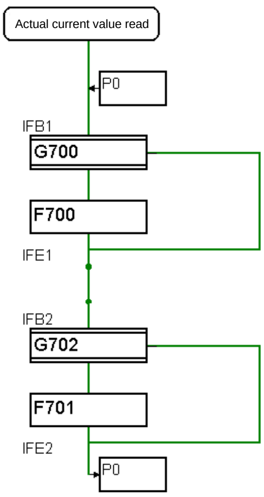
[Speed change] program No. 60

Started automatically.



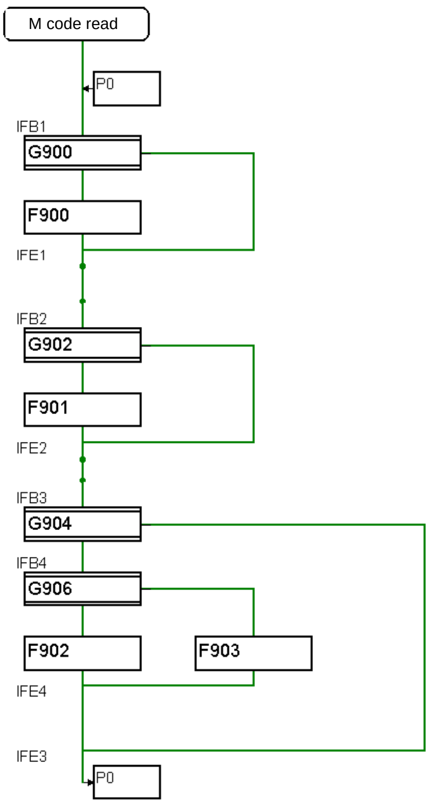
[Actual current value read] program No. 70

Started automatically.



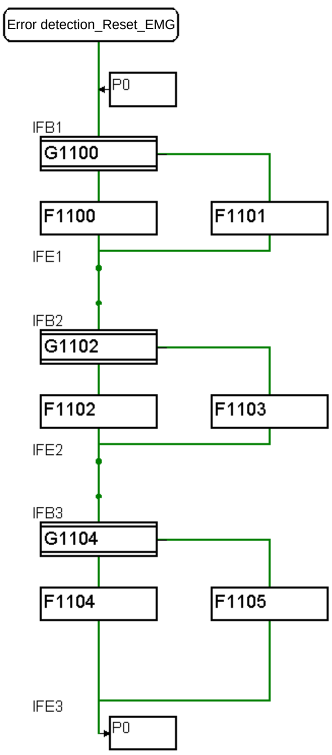
[M code read] program No. 90

Started automatically.



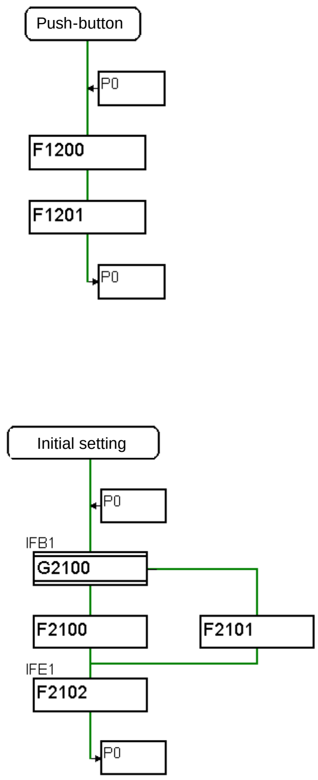
[Error detection_Reset_EMG] program No. 110

Started automatically.

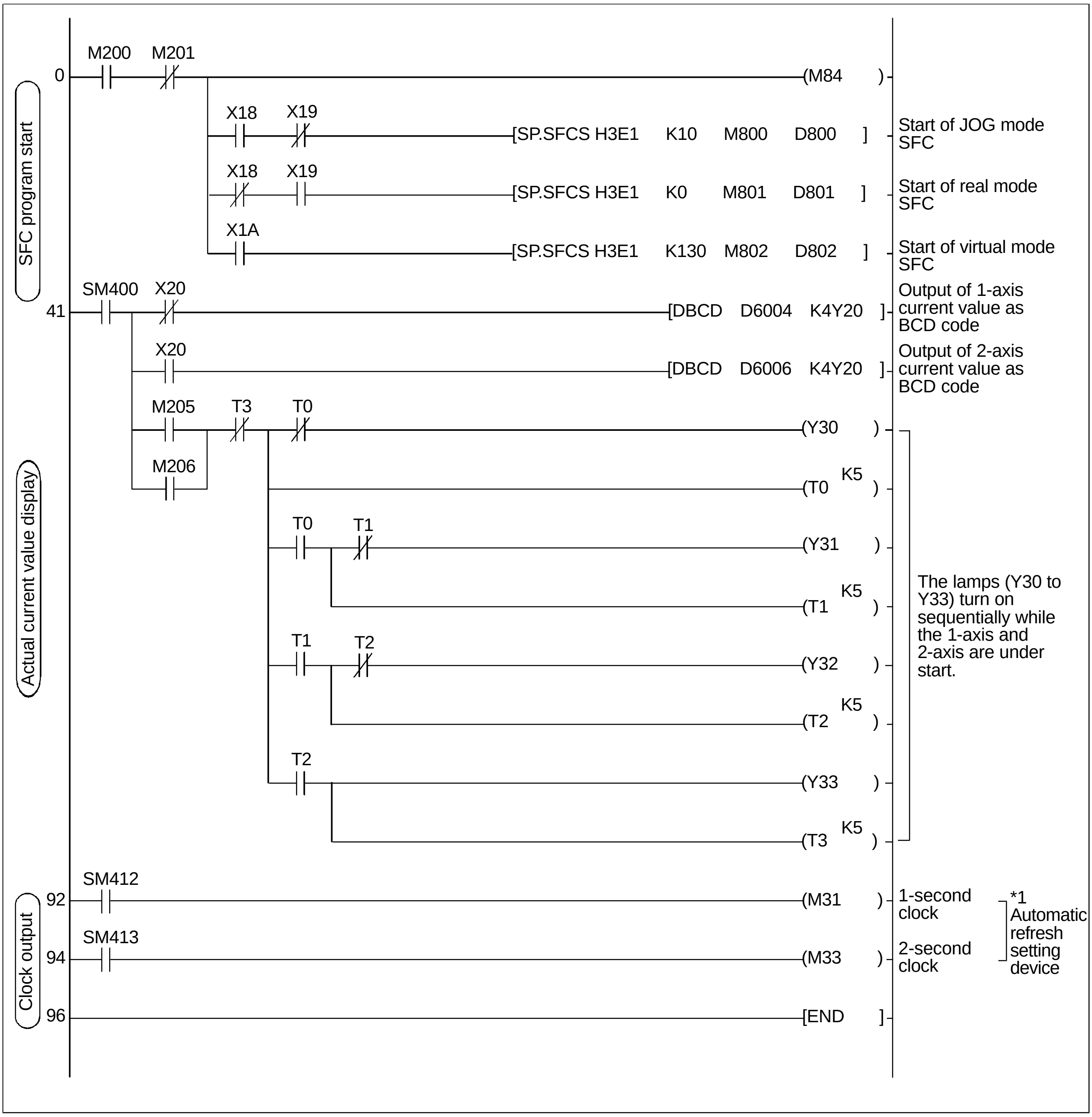


[Push-button] program No. 120

Started automatically.



• **Q02H sequence program**

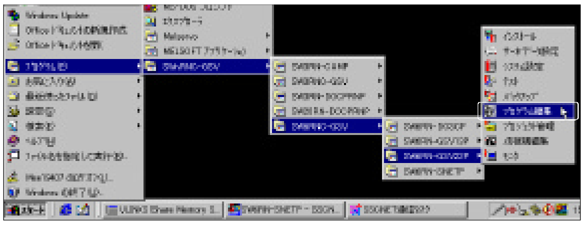


*1 When the clock relay is refreshed for automatic refresh operation, it may not be refreshed at correct timing depending on automatic refresh timing.

10.4 Writing to the motion CPU

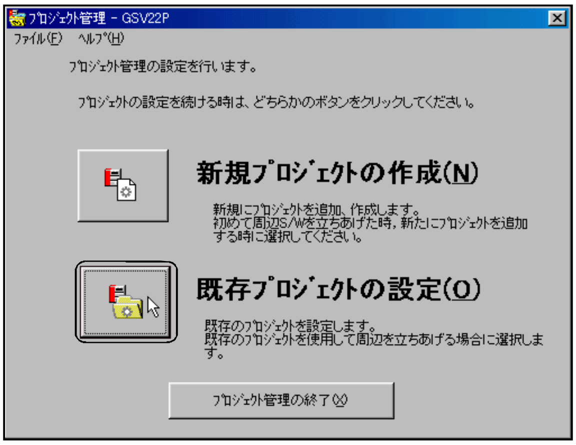
Write the servo setting data and SFC program to the Q172CPU.
Read the existing program from the folder destination path C:\Q172, project name SFC.

(1) Reading from existing program file



1) Click on the [Start], [Program], [SWnRNC-GSV], [SW6RNC-GSV], [SW6RN-GSV22P] and then [Program editing].

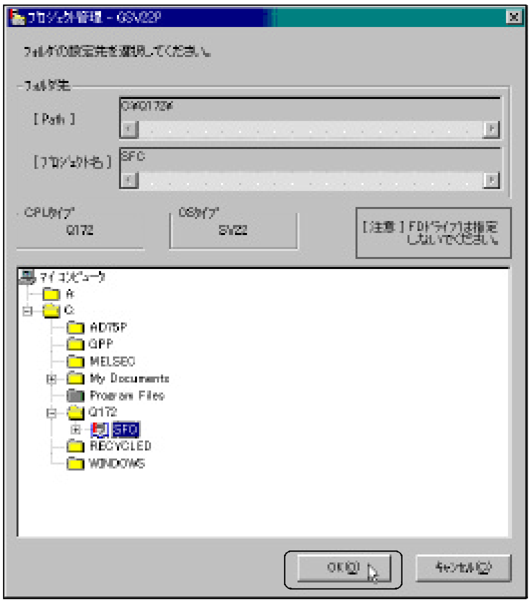
An existing program can be read even when another menu is selected.



2) Click on the Existing project setting button in the PROJECT CONTROL dialog box.

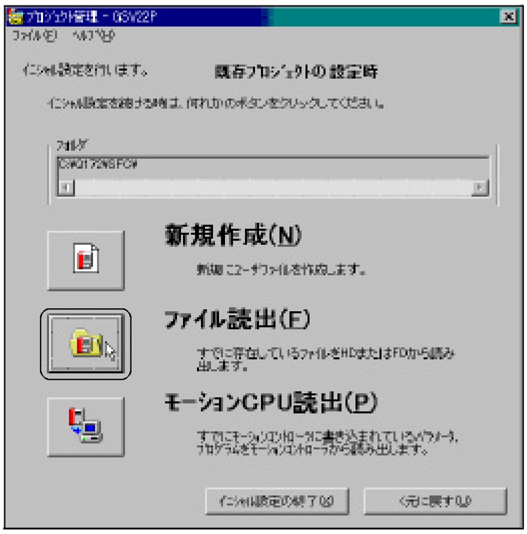
(Continued on next page)

(Continued from previous page)



3) Check that the [Path] under "Folder destination" is set to 'C:\Q172', and that the [Project name] is 'SFC'. Then click on the OK button.

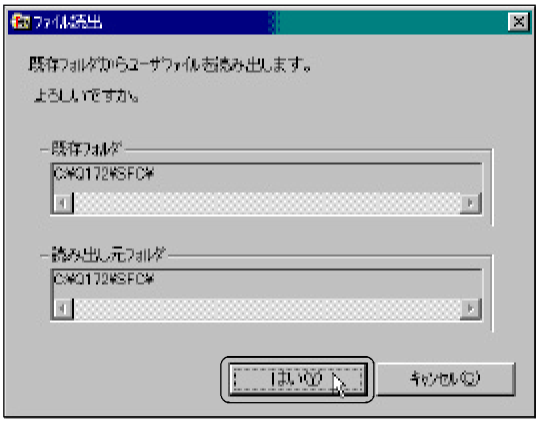
4) Click on the OK button in the PROJECT CONTROL dialog box.



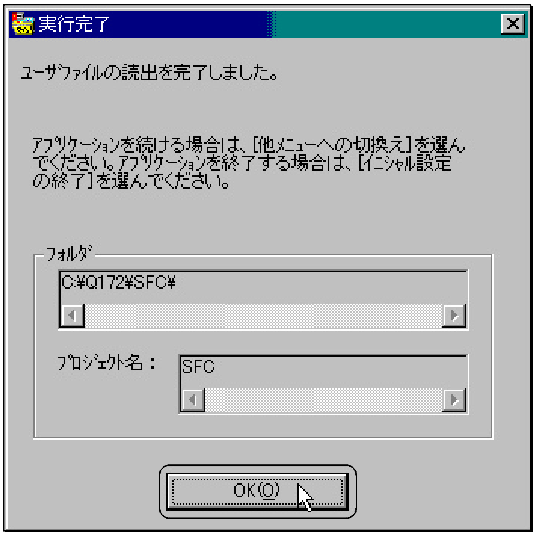
5) Click on the File read button after the screen changes.

(Continued on next page)

(Continued from previous page)



6) The FILE READ dialog box will open, so click on the YES button.

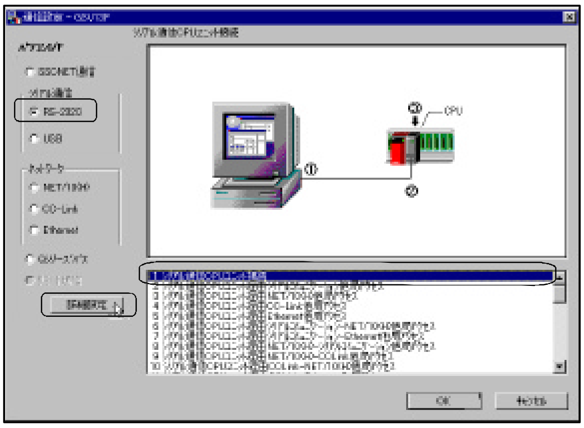


7) The EXECUTION COMPLETED dialog box will open, so click on the OK button.

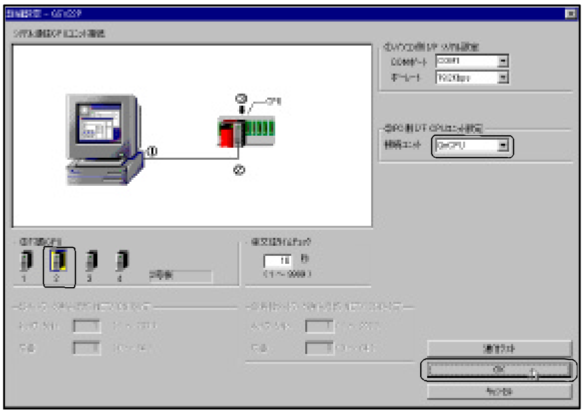
(2) Writing to Q motion CPU

1) Set the Q motion to STOP.

2) Click on the [Communication] and then the [Communication setting] menu on the PROGRAM EDIT window.



3) The COMMUNICATION SETTING dialog box will open, so check "RS-232C", select "1. Serial communication CPU connection" and click on the Detail setting button.



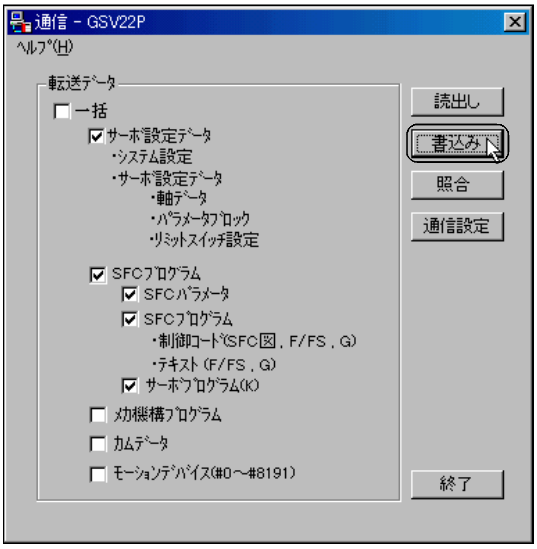
4) The DETAIL SETTING dialog box will open, so set the "PC side I/F CPU" to 'QnCPU', and the "Object CPU" to '#2 machine'.
After setting, click on the OK button.
The COMMUNICATION SETTING dialog box will open again, so click on the OK button.

5) When the message shown at left appears, click on the OK button.

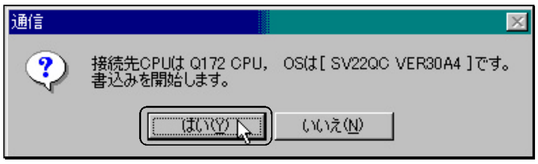
6) Click on the [Communication] and then the [Transfer] menu in PROGRAM EDIT screen.

(Continued on next page)

(Continued from previous page)



7) The COMMUNICATION dialog box will open, so check "Servo setting data" and "SFC program", and click on the Write button.



8) Since the dialog box to confirm the motion CPU type and writing execution, click on the Yes button.

9) When the message "Completed normally." is displayed, click on the OK button.

10) Reset the Q-PLC CPU.

11) Set the mode changeover switch (arranged on the operation panel of practice machine) selected to "Real", run the Q-PLC CPU and Q motion CPU.

The state is normal if "RUN" of Q02HCPU and "RUN/M.RUN" of Q172CPU turn ON.

10.5 Program for operation

This operation sequence/SFC program has been prepared for the SW6RN-GSV22P (for Q172).
The explanatory drawing of the practice machine's operation panel, is shown in section 9.2.
For initial setting program and independent JOG operation start, refer to the section 9.9.

10.5.1 JOG operation

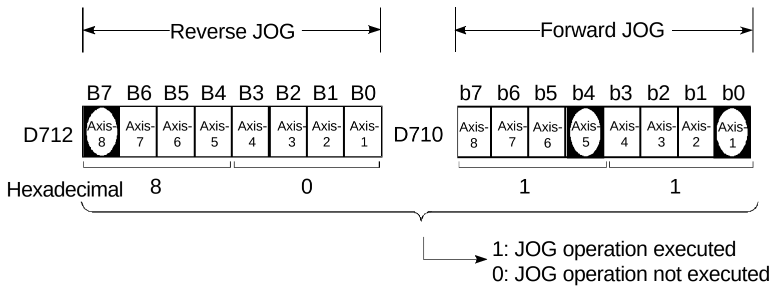
The JOG operation can be executed with independent start or simultaneous start.
For independent start, refer to section 9.9.2.

Simultaneous start

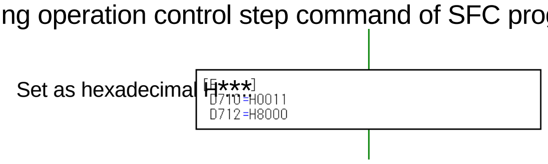
JOG operation of multiple designated axes is started simultaneously.

(1) Simultaneous JOG start axis setting area: D710 to D713

Set the axis for JOG operation in D710 to D713.



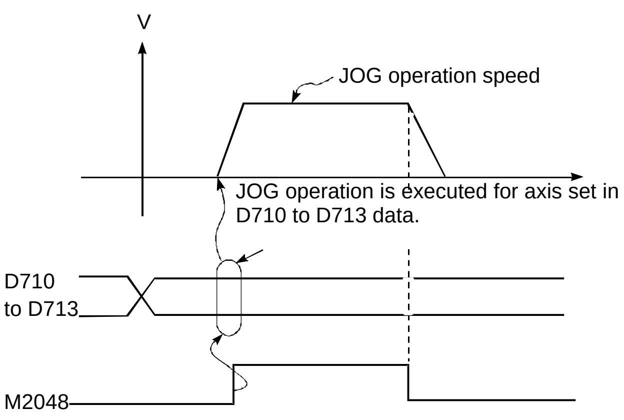
Set as follows to set the axis 1/axis 5 to "Forward JOG" and axis 8 to "Reverse JOG" using operation control step command of SFC program.



(2) JOG simultaneous start command flag (M2048)

JOG operation is executed at the JOG operation speed register value set for each axis while the JOG simultaneous start command flag (M2048) is ON, and is decelerated to a stop when the M2048 is OFF.

The acceleration/deceleration is controlled according to the contents of the parameter block set by the JOG operation data.

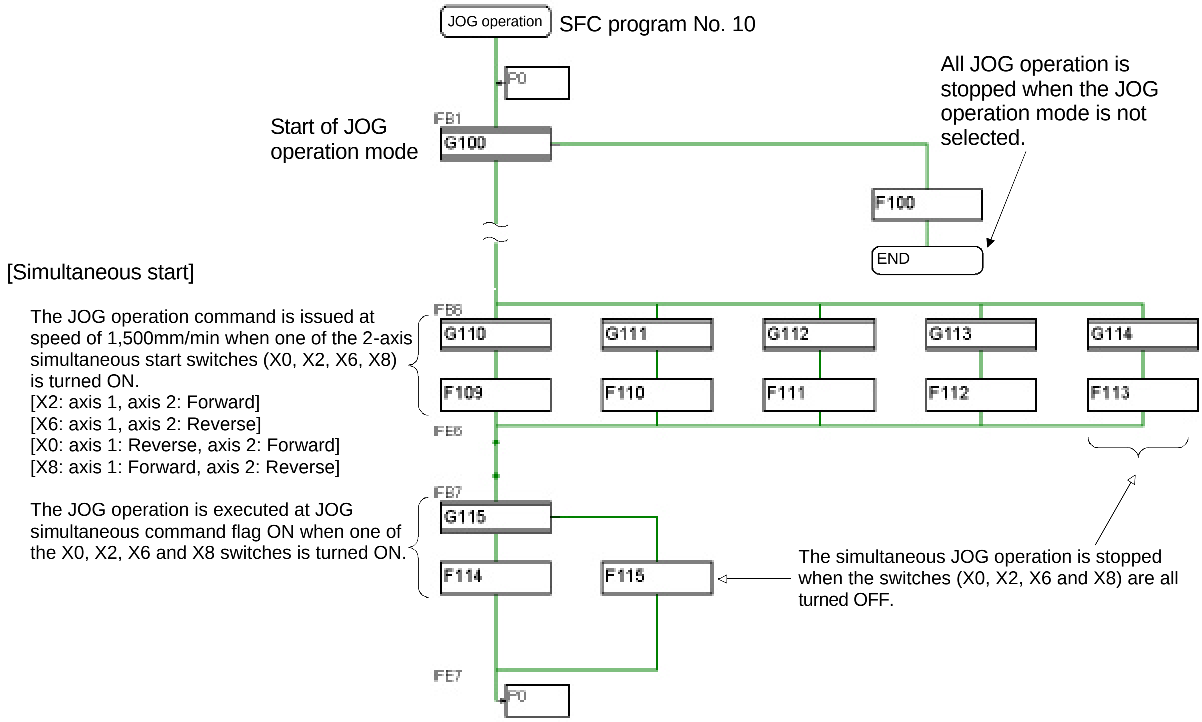


(3) Program example

1) JOG operation conditions

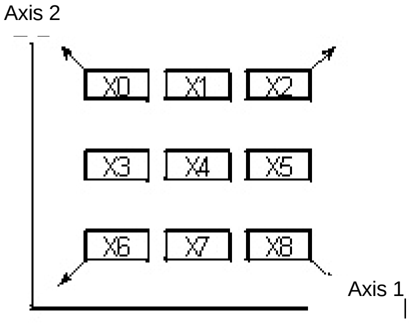
Item	Condition	
Axis used	Axis 1	Axis 2
JOG operation speed	1500mm/min	1500mm/min

2) SFC program example when axis 1 and axis 2 are started simultaneously

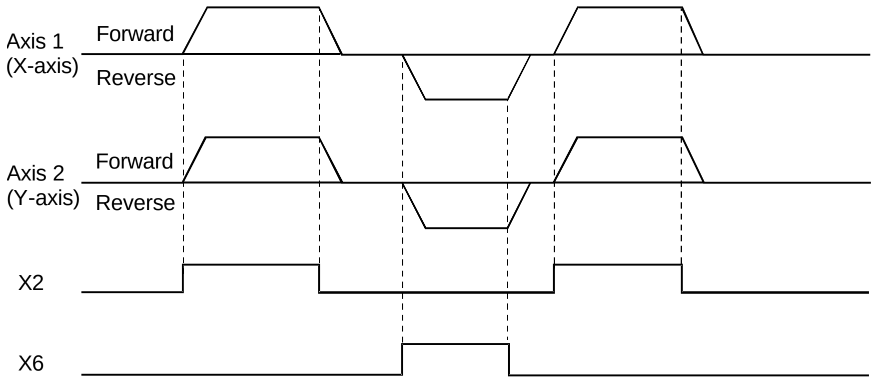


[Transition]	G110 PX2 * !PX6 * !PX0 * !PX8	G111 PX6 * !PX2 * !PX0 * !PX8
	G112 PX0 * !PX2 * !PX6 * !PX8	G113 PX8 * !PX2 * !PX6 * !PX0
	G114 !(PX2 + PX6 + PX0 + PX8)	G115 PX2 + PX6 + PX0 + PX8
[Operation control step]	F100 RST M3202 RST M3203 RST M3222 RST M3233 RST M2048	F109 D710 = H0003 D712 = H0000 D640L = K150000 D642L = K150000
	F110 D710 = H0000 D712 = H0003 D640L = K150000 D642L = K150000	F111 D710 = H0002 D712 = H0001 D640L = K150000 D642L = K150000
	F112 D710 = H0001 D712 = H0002 D640L = K150000 D642L = K150000	F113 RST M2048
	F114 SET M2048	F115 RST M2048

M2048	:	Simultaneous JOG start command flag
D710 to D713	:	Simultaneous JOG operation start axis setting area
PX2	:	Forward JOG command for axis 1, axis 2
PX6	:	Reverse JOG command for axis 1, axis 2
PX0	:	Reverse JOG command for axis 1, forward JOG command for axis 2
PX8	:	Forward JOG command for axis 1, reverse JOG command for axis 2
D641, D640	:	JOG speed setting register for axis 1
D643, D642	:	JOG speed setting register for axis 2



[Timing chart]



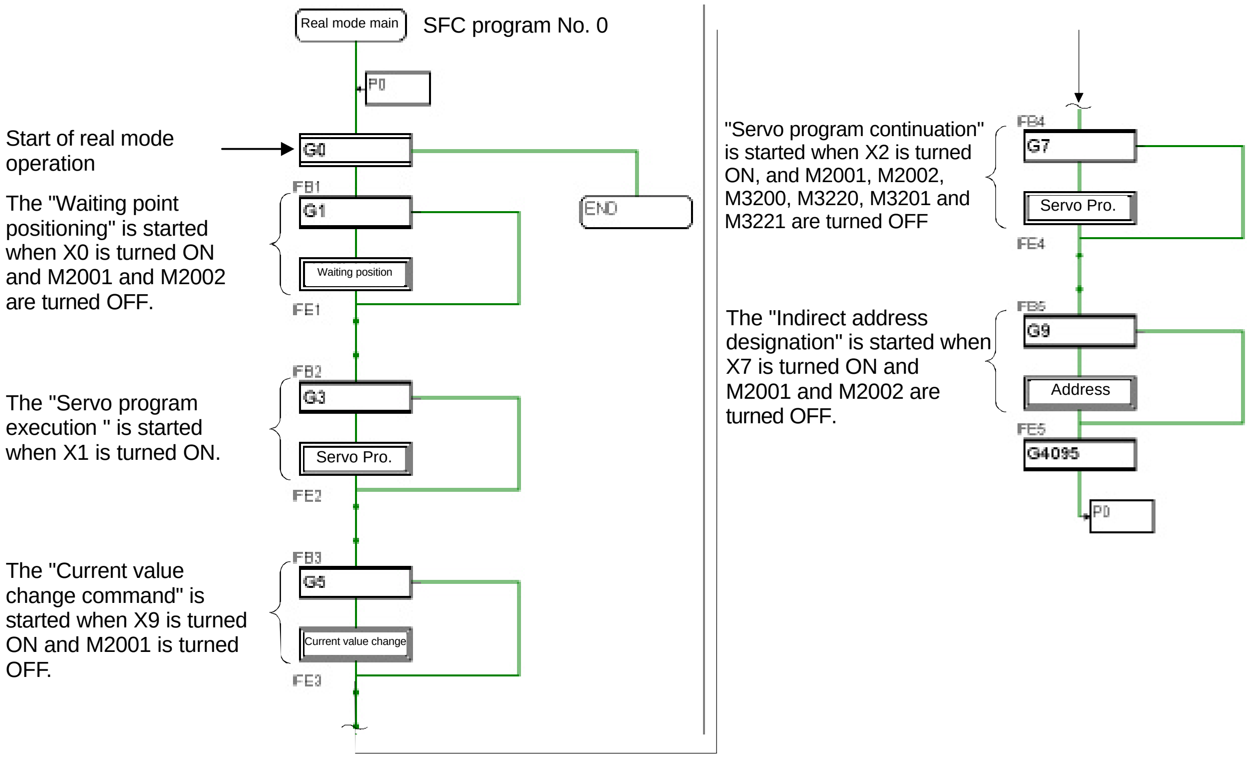
10.5.2 Main routine SFC program (real mode operation)

This a SFC program executed in the main routine when the real mode is selected.
It is used to start other SFC programs (from this main routine SFC program) to execute various operations in the real mode.

(1) SFC program started from main routine SFC program

SFC program No.	Program name	Description section
20	Waiting point positioning	10.5.3
30	Servo program execution	10.5.3
50	Current value change	10.5.6
80	Servo program continuation	10.5.9
110	Indirect address designation	10.5.10

(2) Program example



[Transition]	G0 PX19	G1 PX0*!M2001*!M2002
	G3 PX1	G5 PX09*!M2001
	G7 PX2*!M2001*!M2002*!M3200*!M3220*!M3201 *!M3221	G9 PX7 * !M2001 * !M2002
	G4095 NOP	

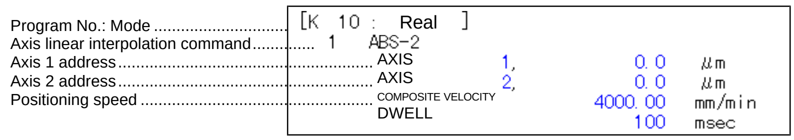
M2001 : Axis 1 start accept flag
M2002 : Axis 2 start accept flag
M3200 : Axis 1 stop command
M3201 : Axis 1 sudden stop command
M3220 : Axis 2 stop command
M3221 : Axis 2 sudden stop command
PX0 : Positioning start command
PX1 : Servo program execution command
PX2 : Positioning start command
PX7 : Indirect setting address receiving command
PX9 : Current value change command

10.5.3 Execution of servo program (motion control step)

When the servo program is executed at the motion control step of the SFC program, the operation is executed according to the contents of the data and parameter block for executed servo program.

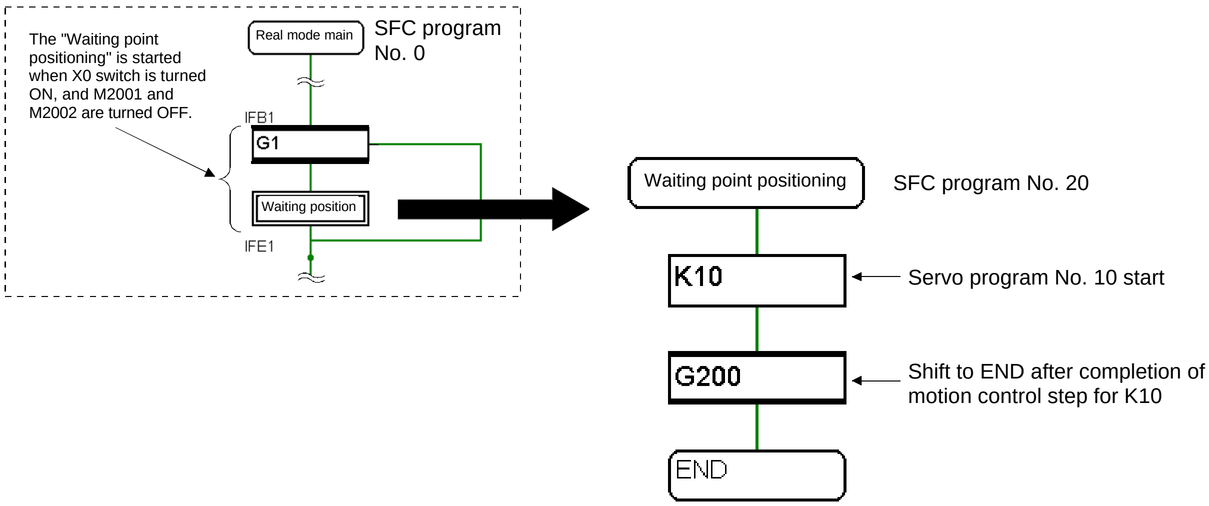
Example 1 Example of SFC program used to execute the servo program No.10 (to execute the linear interpolation of axis 1 and axis 2)

[Servo program]



[SFC program]

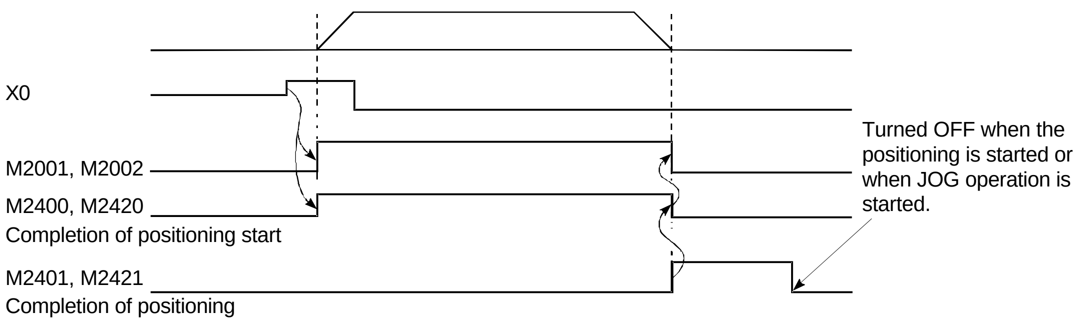
[Real mode main] program



[Transition]	G200 NOP
[Motion control step]	K10: REAL 1 ABS-2 AXIS 1, 0.0 µm AXIS 2, 0.0 µm COMPOSITE VELOCITY 4000.00 mm/min DWELL 100 msec

PX0 : Positioning start command

[Timing chart]

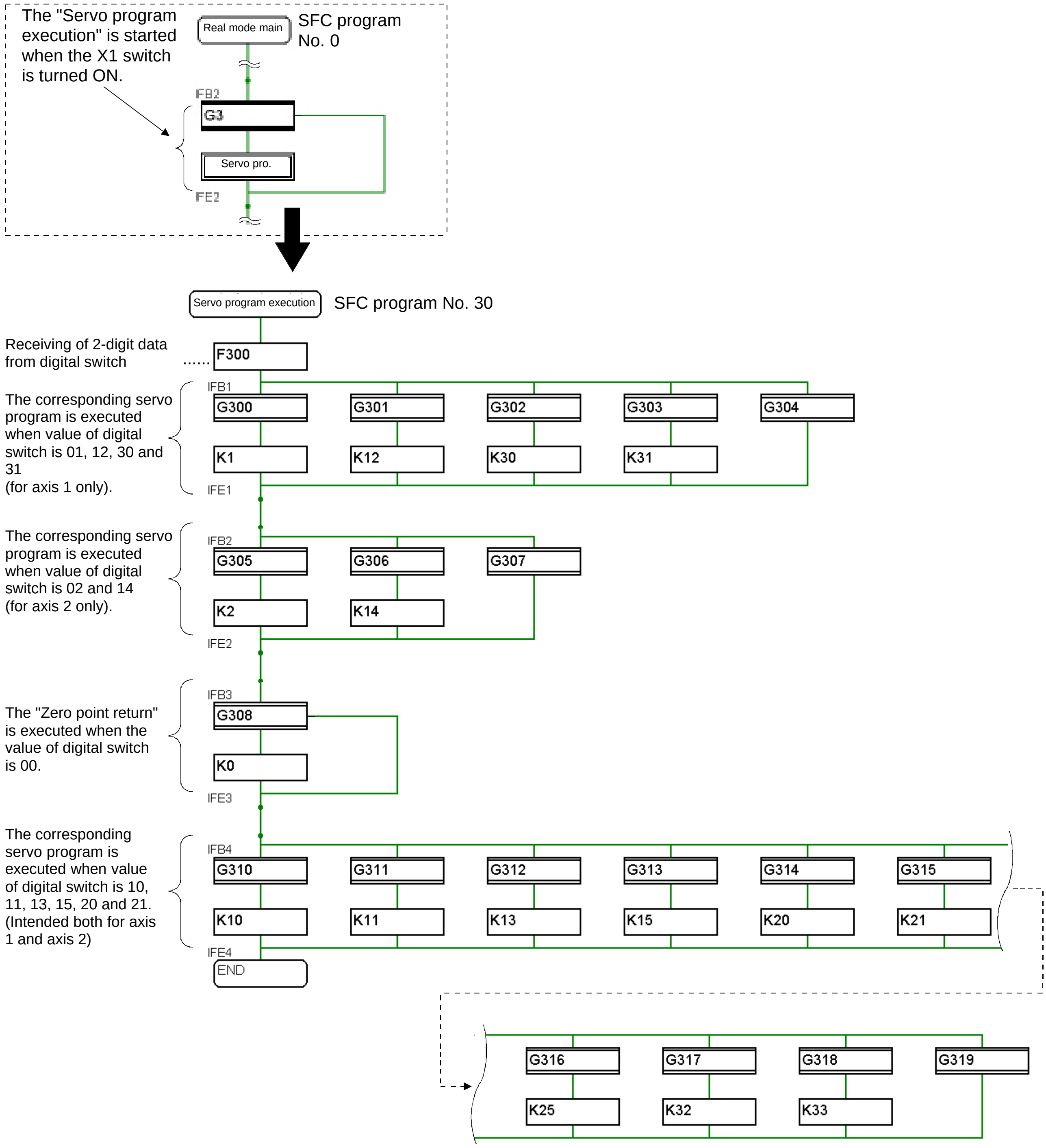


Example 2 Example of SFC program used to execute the servo program No. (designated by two digits of digital switch (X10 to X17)) with indirect settings.

When the servo program No. to be started is prepared as shown below

Axis to control	Servo program No.
Axis 1	1, 12, 30, 31
Axis 2	2, 14
Axis 1, axis 2	0 and other than above

[Real mode main] program



[Transition]	G300 (D4000==K1) * !M2410 * !M2001	G301 (D4000==K12) * !M2001
	G302 (D4000==K30) * !M2001	G303 (D4000==K31) * !M2001
	G304 !((D4000==K1)+(D4000==K12)+(D4000==K30) +(D4000==K31))	G305 (D4000==K2) * !M2430 * !M2002
	G306 (D4000==K14) * !M2002	G307 !((D4000 == K2)+(D4000 == K14))
	G308 (D4000==K0)*!(M2410+M2430+M2001+M2002)	G310 (D4000==K10)*!(M2001+M2002)
	G311 (D4000==K11)*!(M2001+M2002)	G312 (D4000==K13)*!(M2001+M2002)
	G313 (D4000==K15)*!(M2001+M2002)	G314 (D4000==K20)*!(M2001+M2002)
	G315 (D4000==K21)*!(M2001+M2002)	G316 (D4000==K25)*!(M2001+M2002)
	G317 (D4000==K32)*!(M2001+M2002)	G318 (D4000==K33)*!(M2001+M2002)
	G319 !((D4000==K10)+(D4000==K11)+(D4000==K13) +(D4000==K15)+(D4000==K20)+(D4000==K21) +(D4000==K25)+(D4000==K33))	
[Operation control step]	F300 DIN D4000,PX10 D4000=D4000%H100 D4000=BIN(D4000)	
[Motion control step]	K0: REAL 1 START PROGRAM NO. K 1 PROGRAM NO. K 2 PROGRAM NO.	K1: REAL 1 ZERO AXIS 1
	K2: REAL 1 ZERO AXIS 2	K10: REAL 1 ABS-2 AXIS 1, 0.0µm AXIS 2, 0.0µm COMPOSITE VELOCITY 4000.00mm/min DWELL 100msec
	K11: REAL 1 ABS-2 AXIS 1, 2000.0µm AXIS 2, 2000.0µm COMPOSITE VELOCITY 5000.00mm/min	K12: REAL 1 ABS-1 AXIS 1, 300000.0µm VELOCITY 4000.00mm/min M CODE 3
	K13: REAL 1 ABS-2 AXIS 1, 20000.0µm AXIS 2, 200000.0µm COMPOSITE VELOCITY 7000.00mm/min M CODE 3	K14: REAL 1 ABS-1 AXIS 2, 20000.0µm VELOCITY 4000.00mm/min M CODE 3
	K15: REAL 1 ABS-2 AXIS 1, 0.0µm AXIS 2, 0.0µm COMPOSITE VELOCITY 5000.00mm/min	K20: REAL 1 ABS-2 AXIS 1, 200000.0µm AXIS 2, 150000.0µm COMPOSITE VELOCITY 5000.00mm/min
	K21: REAL 1 ABS↷ AXIS 1, 200000.0µm AXIS 2, 150000.0µm VELOCITY 6000.00mm/min CENTER POINT 1, 250000.0µm CENTER POINT 2, 150000.0µm M CODE 4	K25: REAL 1 ABS-2 AXIS 1, D 4006µm AXIS 2, D 4008µm COMPOSITE VELOCITY 5000.00mm/min
	K30: REAL 1 VF AXIS 1 VELOCITY 2500.00 mm/min	K31: REAL 1 VR AXIS 1 VELOCITY 2500.00mm/min

[Motion control step]	K32: REAL 1 VPF AXIS 1, 100000.0µm VELOCITY 5000.00mm/min	K33: REAL 8 CPSTART2 AXIS 1 AXIS 2 VELOCITY 4000.00mm/min 1 ABS-2 AXIS 1, 25000.0µm AXIS 2, 25000.0µm * FOR-TIMES SET VALUE K 6 2 INC-2 AXIS 1, 0.0µm AXIS 2, 175000.0µm 3 INC⌒ AXIS 1, 25000.0µm AXIS 2, 0.0µm AUXILIARY P 1, 125000.0µm AUXILIARY P 2, 125000.0µm 4 INC-2 AXIS 1, 0.0µm AXIS 2, –175000.0µm 5 INC⌒ AXIS 1, 25000.0µm AXIS 2, 0.0µm AUXILIARY P 1, 12500.0µm AUXILIARY P 2, –12500.0µm * NEXT 6 INC-2 AXIS 1, 0.0µm AXIS 2, 175000.0µm 7 ABS-2 AXIS 1, 0.0µm AXIS 2, 0.0µm M CODE 0 8 CPEND

M2410 : Axis 1 zero point return completed
M2430 : Axis 2 zero point return completed
M2001 : Axis 1 start accept flag
M2002 : Axis 2 start accept flag
PX1 : Servo program execution command
D4000 : Name of device designated indirectly

10.5.4 Stopping

It is possible to stop the operation either by "Deceleration stop" or "Sudden stop".

(1) Deceleration stop signal and sudden stop signal

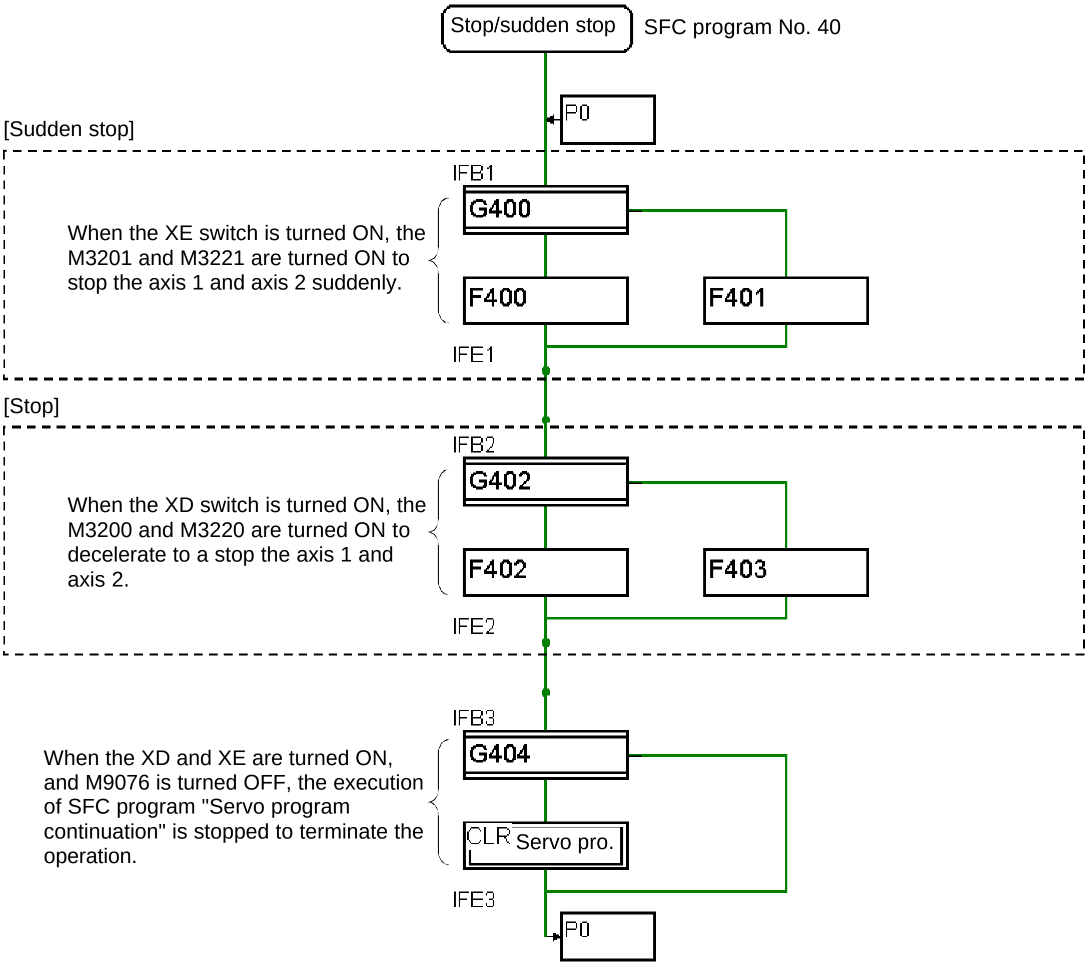
Control axis	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Deceleration stop	M3200	M3220	M3240	M3260	M3280	M3300	M3320	M3340
Sudden stop	M3201	M3221	M3241	M3261	M3281	M3301	M3321	M3341

[Example 1] SFC program used to stop the axis 1 and axis 2 suddenly

The axis 1 and axis 2 are stopped at the sudden stop deceleration time designated by parameter block No. 1

[Example 2] SFC program used to decelerate the axis 1 and axis 2 to a stop

The axis 1 and axis 2 are stopped at deceleration time designated by parameter block No. 1.



[Transition]	G400 PX0E	G402 PX0D*!M3201*!M3221
	G404 PXD + PXE + !M9076	
[Operation control step]	F400 SET M3201 SET M3221	F401 RST M3201 RST M3221
	F402 SET M3200 SET M3220	F403 RST M3200 RST M3220

PX0D : Deceleration stop command
PX0E : Sudden stop command
M9076 : Emergency stop input flag

10.5.5 Error reset

When an error occurs, the error detection signal (M2407/axis 1) is turned ON, causing the minor error code or major error code to be stored in the monitor data register.
When a servo error occurs, on the other hand, the servo error detection signal (M2408/ axis 1) is turned ON, causing the servo error code to be stored in the monitor data register.
When the error is reset after correcting the contents of error, it is possible to clear the error detection signal and object monitor data register.

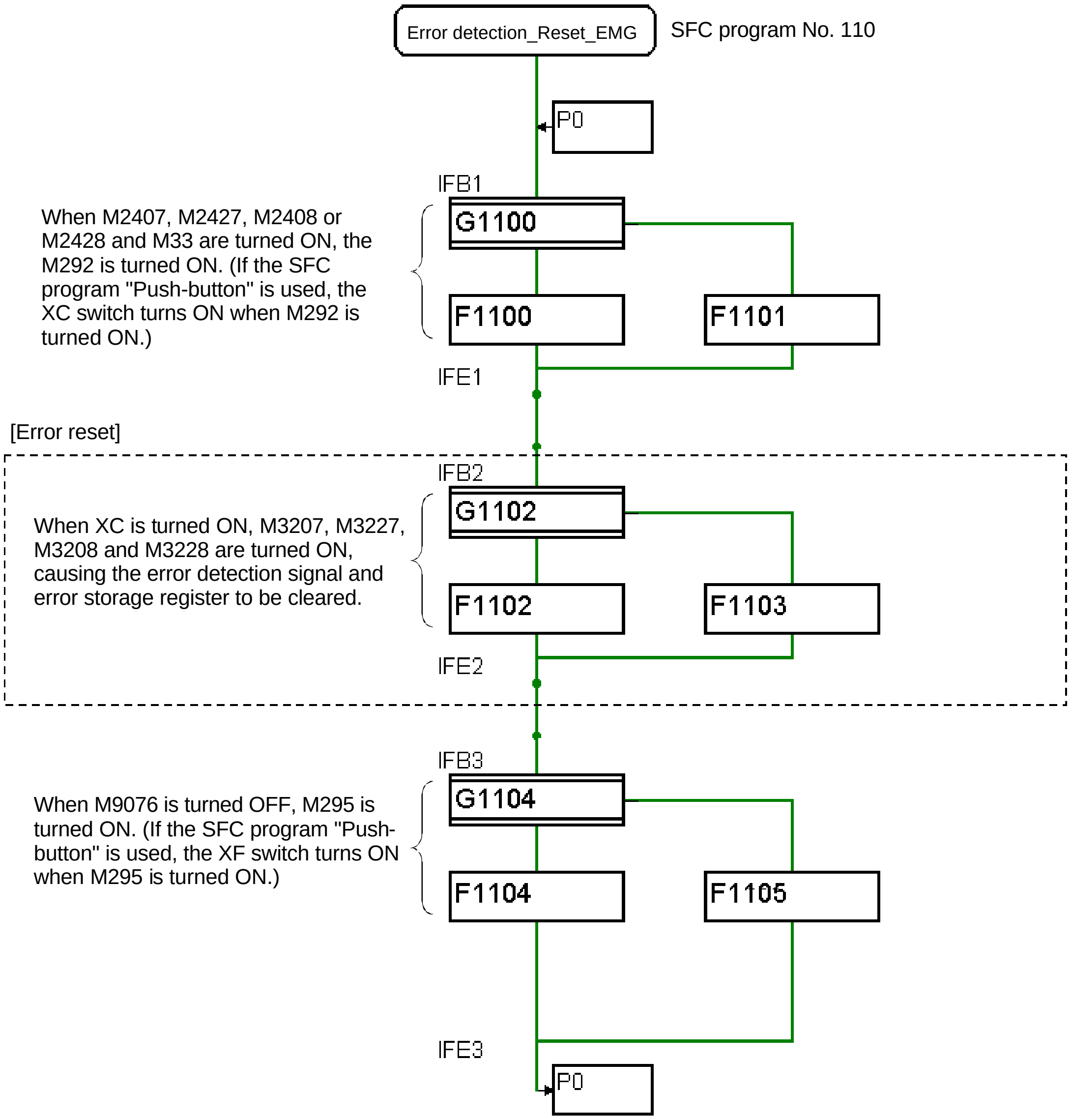
(1) Error detection signal and error storage register

Control axis	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Error detection	M2407	M2427	M2447	M2467	M2487	M2507	M2527	M2547
Minor error	D6	D26	D46	D66	D86	D106	D126	D146
Major error	D7	D27	D47	D67	D87	D107	D127	D147
Servo error	M2408	M2428	M2448	M2468	M2488	M2508	M2528	M2548
Servo error	D8	D28	D48	D68	D88	D108	D128	D148

(2) Error reset signal

Control axis	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Error reset	M3207	M3227	M3247	M3267	M3287	M3307	M3327	M3347
Servo error	M3208	M3228	M3248	M3268	M3288	M3308	M3328	M3348

(3) Example of SFC program to reset axis 1/axis 2 error



[Transition]	G1100 (M2407+M2427+M2408+M2428)*M33	G1102 PXC
	G1104 !M9076	
[Operation control step]	F1100 SET M292	F1101 RST M292
	F1102 SET M3207 SET M3227 SET M3208 SET M3228	F1103 RST M3207 RST M3227 RST M3208 RST M3228
	F1104 SET M295	F1105 RST M295

PXC : Error reset switch

PXF : EMG switch

10.5.6 Current value change

Change the position at which the axis designated by CHGA command (change address) of servo program is stopped to the set address.

CAUTION
Changing the current value during start may cause a minor error (code 300) to occur, causing it to be unable to be executed.

(1) Current value change setting range

Current value change setting range							
mm		inch		degree		pulse	
Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
-2^{31} to $2^{31}-1$	$\times 10^{-4}$ mm	-2^{31} to $2^{31}-1$	$\times 10^{-5}$ inch	0 to 36999999	$\times 10^{-5}$ deg	-2^{31} to $2^{31}-1$	pulse

CAUTION
• Even if the set value is out of stroke range, it will not result in an error. • $-2^{31} = -2147483648$ • $2^{31}-1 = +2147483647$

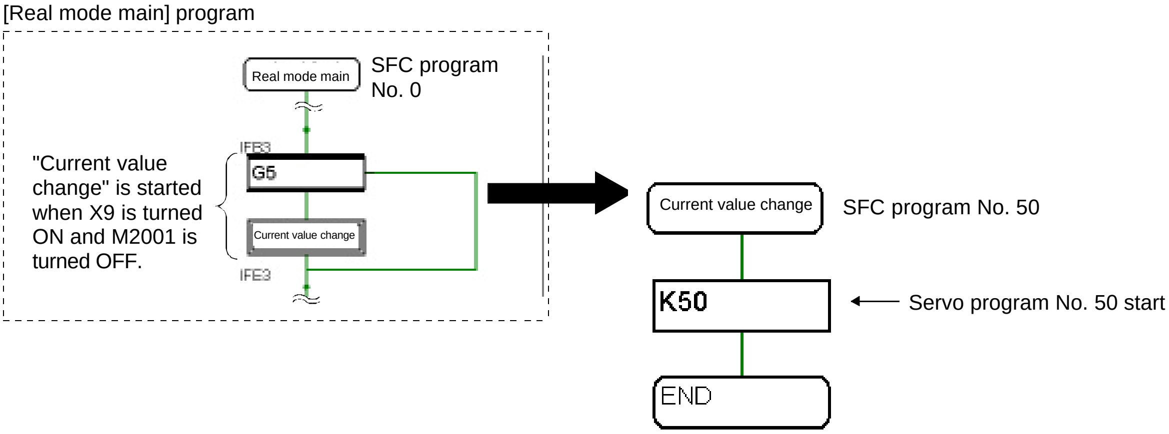
(2) Program example

1) Current value change condition

Control axis	Axis 1	Axis 2
Address after change	111.222mm	

2) Example of SFC program to execute the servo program No.50 (to change the current value of the stopped axis 1 forcibly to 111.222mm)

[Real mode main] program



[Motion control step]	K50: REAL 1 CHGA AXIS 1, 111222.0μm

M2001 : Axis 1 start accept flag

PX09 : Current value change command

[Servo program]

```
[K 50 : REAL ]
 1   CHGA                          → Current value change command
     AXIS      1,    111222. 0  μm → Address after change of axis 1
                                     (Set in 0.1μm units)
```

10.5.7 Speed change (CHGV)

The motion dedicated function CHGV command (change velocity) is used to forcibly change the speed set during positioning control (excluding circular interpolation) and JOG operation.

(1) CHGV speed change request command

This item describes the number of the axis for which the speed is to be changed and the new speed.

CHGV (K1, K30000)

K [Speed after change]
D0 to D8191
W0 to 1FFFF
R0 to R8191

→ Axis No. (1 to 8) for which speed is to be changed

(2) Speed change setting range

Speed change setting range							
mm		inch		degree		pulse	
Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
–600000000 to 600000000	$\times 10^{-2}$ mm/min	–600000000 to 600000000	$\times 10^{-3}$ inch/min	–2147483648 to 2147483648	$\times 10^{-3}$ degree/min	–10000000 to 10000000	pulse/s

POINT

When setting the speed using the CHGV command, set the value given by multiplying the actual speed by 100 (mm)/1,000 (inch, degree).

(3) Program example

1) Speed change condition

Items		Conditions	
Control axis		Axis 1	Axis 2
Speed change command input	X3	Speed after change: 2000mm/min	———
	X4	Speed after change: 1000mm/min	———
	X5	Speed after change: 300mm/min	———

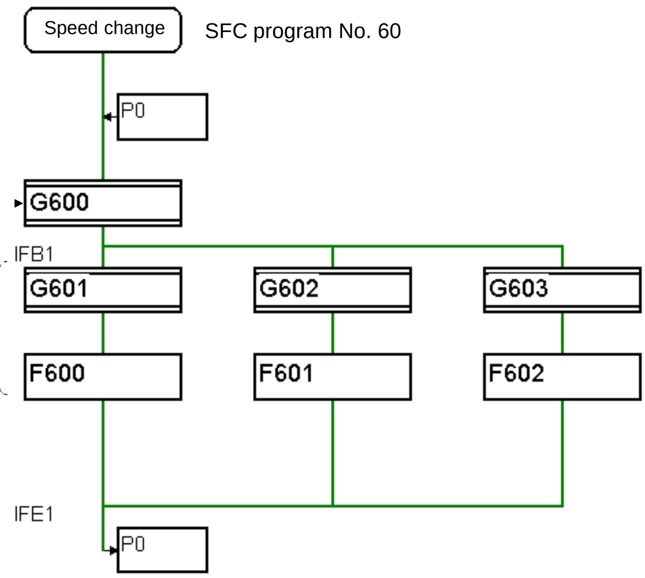
Example

To change the speed to 10,000.00mm/min, set '1000000'.

2) Speed change program example

Changed when the real mode is selected.

If M2001 is turned ON and M2002 is turned OFF, the speed is changed to 2,000mm/min when X3 is turned ON.
The speed is changed to 1,000mm/min when X4 is turned ON.
The speed is changed to 300mm/min when X5 is turned ON.



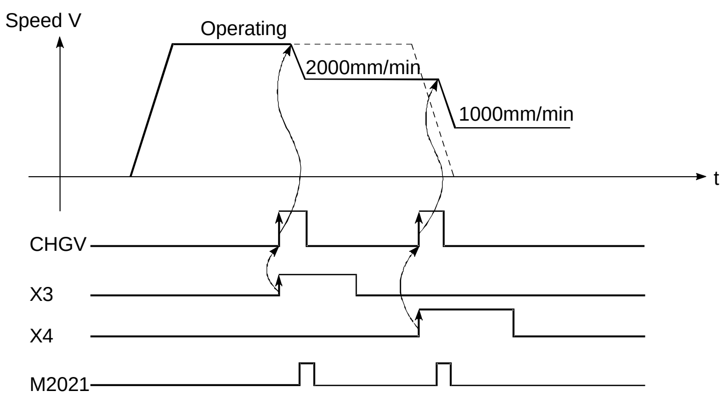
[Transition]	G600 PX19	G601 PX3 * M2001 * !M2061
	G602 PX4 * M2001 * !M2061	G603 PX5 * M2001 * !M2061
[Operation control step]	F600 CHGV(K1,K200000)	F601 CHGV(K1,K100000)
	F602 CHGV(K1,K30000)	

M2001 : Axis 1 start accept flag
M2061 : Axis 1 speed change flag
PX3 : Change speed to 2000mm/min command
PX4 : Change speed to 1000mm/min command
PX5 : Change speed to 300mm/min command

```
[F 600]
    1  CHGV(K1, K200000)
```

Speed change command (Axis 1, speed after change 'Setting in units of 0.01mm/min')

[Timing chart]



CAUTION
• The speed cannot be changed when stopped. • The speed cannot be changed during zero point return, circular interpolation or deceleration. • The speed can be changed within range of 1 to speed limit value.

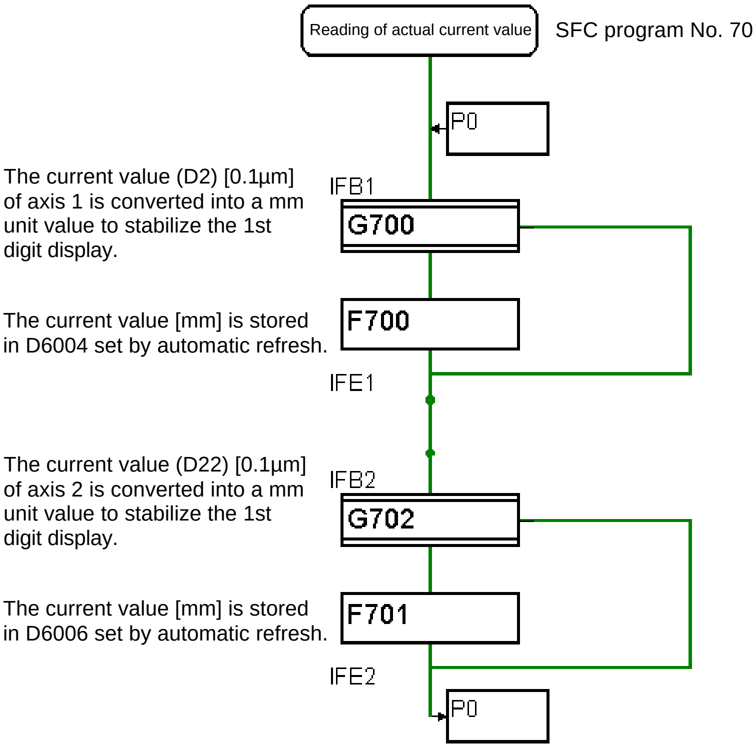
10.5.8 Reading actual current value

The monitor data includes D0 to D159 stored in the actual current value storage register (shown below). Consequently, a read program does not need to be created.

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Actual current value	D2	D22	D42	D62	D82	D102	D122	D142
	D3	D23	D43	D63	D83	D103	D123	D143

Example SFC program and sequence program used to output the actual current value of axis 1 and axis 2 to the digital display unit (controlled by Q-PLC CPU) in [mm] units

(1) SFC program



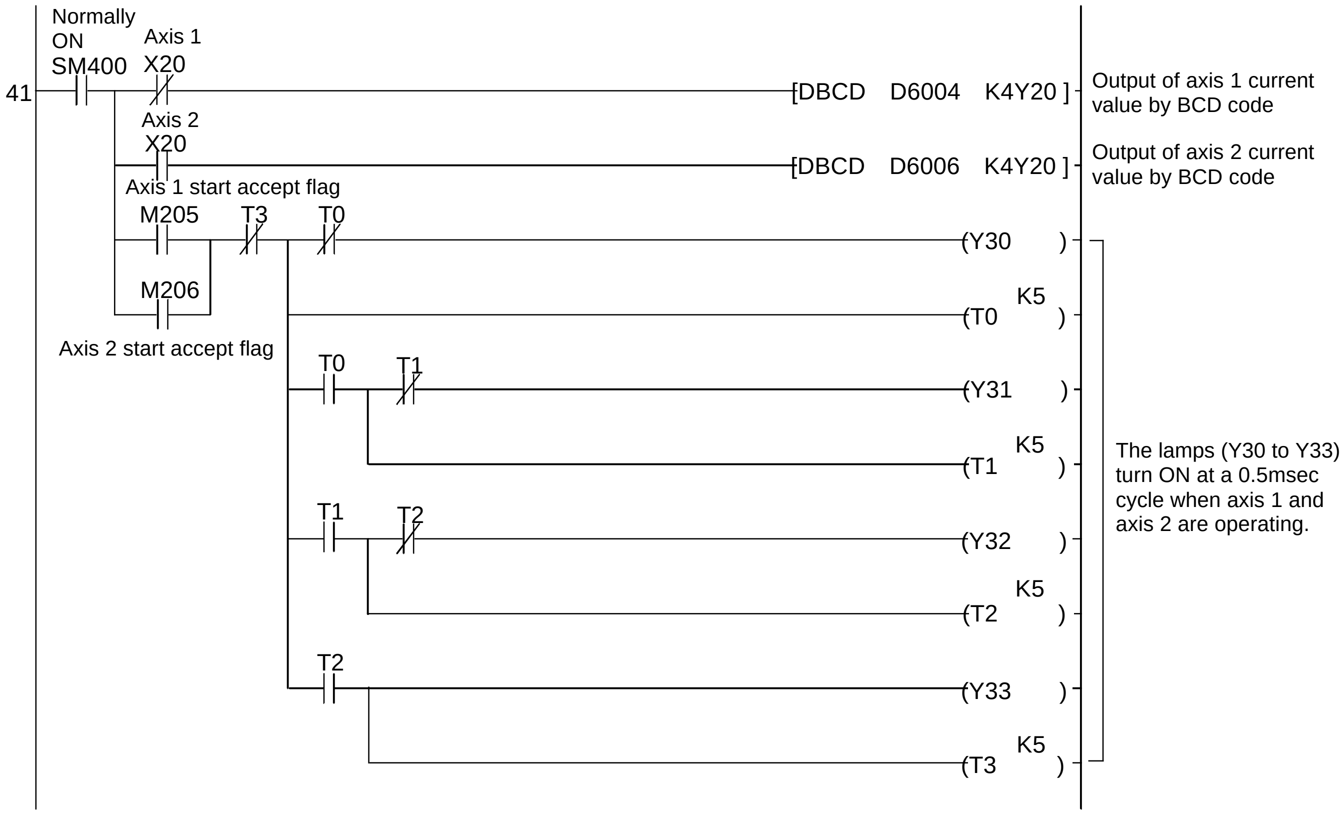
[Transition]	G700 D4016L=D2L/K10000 D4014L=D4016L%K10 (K0==D4014L)*(K0<=D4016L)	G702 D4020L=D22L/K10000 D4018L=D4020L%K10 (K0==D4018L)*(K0<=D4020L)
[Operation control step]	F700 D6004L=D4016L	F701 D6006L=D4020L

D3, D2 : Axis 1 actual current value storage register
D23, D22 : Axis 2 actual current value storage register
D6004, D6005 : Axis 1 actual current value (Automatic refresh setting device)
D6006, D6007 : Axis 2 actual current value (Automatic refresh setting device)

REFERENCE

Since the actual current value is stored as a 32-bit data in [0.1µm] unit, it is divided by 10,000 to convert into a [mm] unit.

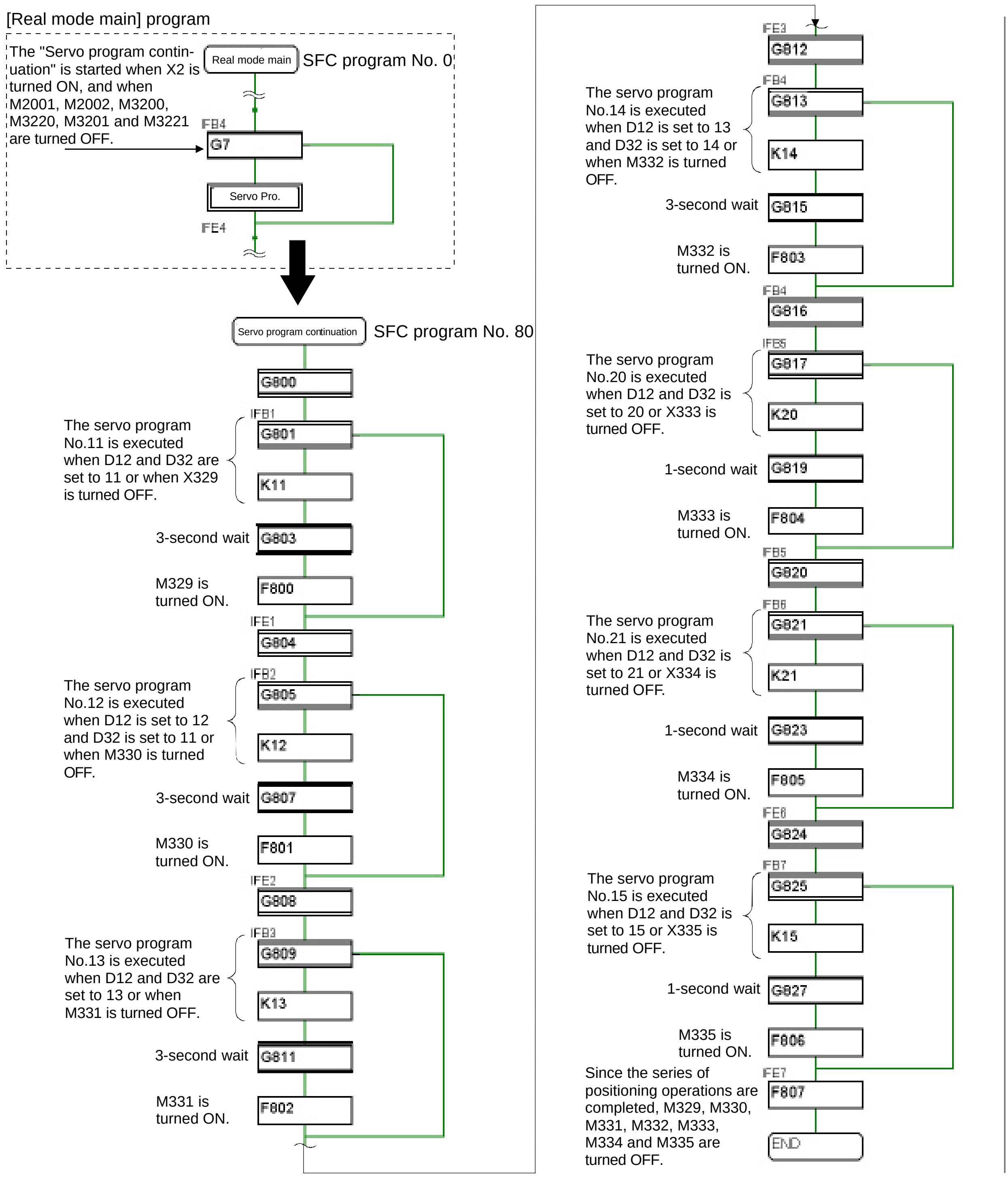
(2) Q02HCPU sequence program



10.5.9 Continuous positioning

To execute the servo program in the sequence of 11, 12, 13, 14, 20, 21 and 15, use the transition of 'WAIT' type after the motion control step (servo program) to shift to the succeeding motion control step (servo program) after completion of servo program under execution.

When the servo program is interrupted during execution, re-execute the program continuously from the interrupted servo program.



[Transition]	G800 PX19	G801 ((D12 == K11)*(D32 == K11))+ !M329
	G803 TIME K3000	G804 PX19
	G805 ((D12 == K12)*(D32 == K11))+ !M330	G807 TIME K3000
	G808 PX19	G809 ((D12 == K13)*(D32 == K13))+ !M331
	G811 TIME K3000	G812 PX19
	G813 ((D12 == K13)*(D32 == K14))+ !M332	G815 TIME K3000
	G816 PX19	G817 ((D12 == K20)*(D32 == K20))+ !M333
	G819 TIME K1	G820 PX19
	G821 ((D12 == K21)*(D32 == K21))+ !M334	G823 TIME K1
	G824 PX19	G825 ((D12 == K15)*(D32 == K15))+ !M335
	G827 TIME K1000	
[Operation control step]	F800 SET M329	F801 SET M330
	F802 SET M331	F803 SET M332
	F804 SET M333	F805 SET M334
	F806 SET M335	F807 RST M329 RST M330 RST M331 RST M332 RST M333 RST M334 RST M335
[Motion control step]	K11: REAL 1 ABS-2 AXIS 1, 2000.0µm AXIS 2, 2000.0µm COMPOSITE VELOCITY 5000.00mm/min	K12: REAL 1 ABS-1 AXIS 1, 300000.0µm VELOCITY 4000.00mm/min M CODE 3
	K13: REAL 1 ABS-2 AXIS 1, 20000.0µm AXIS 2, 200000.0µm COMPOSITE VELOCITY 7000.00mm/min M CODE 3	K14: REAL 1 ABS-1 AXIS 2, 20000.0µm VELOCITY 4000.00mm/min M CODE 3
	K15: REAL 1 ABS-2 AXIS 1, 0.0µm AXIS 2, 0.0µm COMPOSITE VELOCITY 5000.00mm/min	K20: REAL 1 ABS-2 AXIS 1, 200000.0µm AXIS 2, 150000.0µm COMPOSITE VELOCITY 5000.00mm/min
	K21: REAL 1 ABS↷ AXIS 1, 200000.0µm AXIS 2, 150000.0µm VELOCITY 6000.00mm/min CENTER POINT 1, 250000.0µm CENTER POINT 2, 150000.0µm M CODE 4	

D12 : Program No. executed by axis 1
D13 : Program No. executed by axis 2
M329 : Completion of servo program No.11
M330 : Completion of servo program No.12
M331 : Completion of servo program No.13
M332 : Completion of servo program No.14
M333 : Completion of servo program No.20
M334 : Completion of servo program No.21
M335 : Completion of servo program No.15

10.5.10 M code function

The M code No. ranges from 0 to 255, and is added to the servo program. When this servo program is executed, the M code is set in the M code monitor register.
Since the M code is known when it is checked using the compare command in the sequence program, pre-determined work can be executed.

(1) Example of servo program (with M code added)

```
[K  12 :  REAL ]
  1   ABS-1
       AXIS         1,        300000. 0    µm
       VELOCITY                4000. 00    mm/min
       M CODE                       3
```

Addition of M code "3"

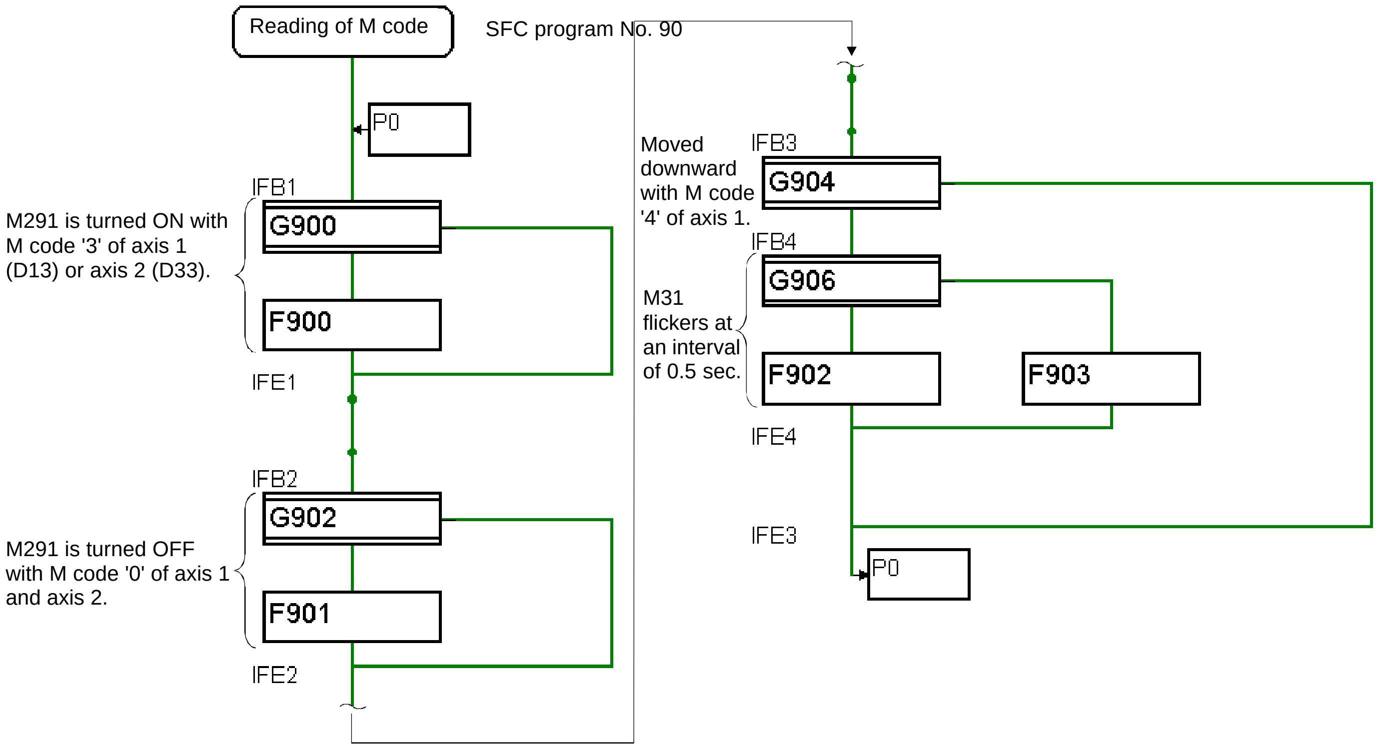
(2) Practice conditions

- When the M code '3' is detected, the pen is lowered.
- When the M code '0' is detected, the pen is raised.
- When the M code '4' is detected, the pen is raised/lowered repeatedly.
- The solenoid to move the pen is wired to the Y0B. (The pen is lowered when the solenoid is excited.)

(3) SFC program

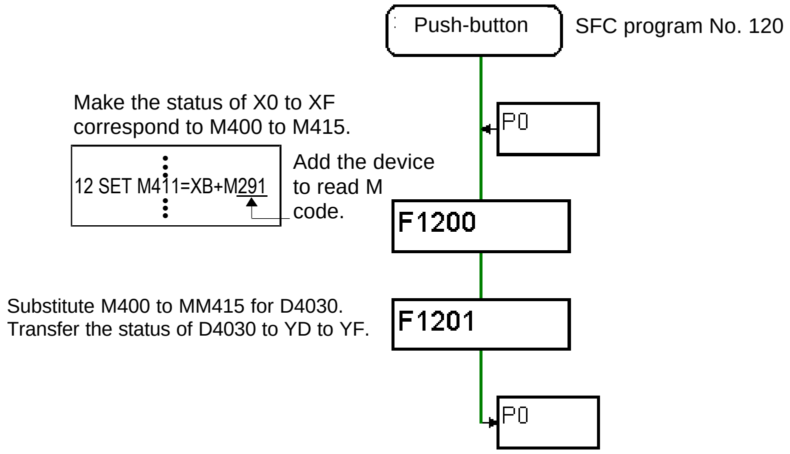
It is necessary to create a program to read the M code and to raise/lower the pen.

1) Read the M code.



[Transition]	G900 (D13==K3)+(D33==K3)	G902 (D13==K0)*(D33==K0)
	G904 (D13==K4)*M2001*M2002	G906 M31
[Operation control step]	F900 SET M291	F901 RST M291
	F902 SET M291	F903 RST M291

2) Substitute the M code read out to the other device to raise/lower the pen.



	F1200	F1201
[Operation control step]	SET M400=PX0 SET M401=PX1 SET M402=PX2 SET M403=PX3 SET M404=PX4+M84 SET M405=PX5 SET M406=PX6 SET M407=PX7 SET M408=(PX8*!M2044)+M322 SET M410=PXA+M2044 SET M411=PXB+M291 SET M412=PXC+M292+M316+M2045 SET M413=PXD SET M414=PXE SET M415=PXF+M295	RST M400=!PX0 RST M401=!PX1 RST M402=!PX2 RST M403=!PX3 RST M404=!PX4*!M84 RST M405=!PX5 RST M406=!PX6 RST M407=!PX7 RST M408=!PX8+!M2044 RST M408=!M322*M2044 RST M410=!PXA RST M411=!PXB*!M291 RST M412=!PXC*!M292*!M316*!M2045 RST M413=!PXD RST M414=!PXE RST M415=!PXF*!M295 DIN D4030,M400 DOUT PY0,D4030

When M291 is turned ON, M411 is turned ON.
M411 functions to turn ON Y0B with the DIN command and DOUT command to lower the pen.

D13 : M code No. of axis 1
D13 : M code No. of axis 2
M31 : 0.5-second timing block (automatic refresh setting device)
M2001 : Axis 1 start accept flag
M2002 : Axis 2 start accept flag

DIN command : Inputs the bit device (handled as a word) to the word device.

DOUT command : Outputs the word device to the bit device (handled as a word).

10.5.11 Indirect setting of servo program address

Indirect settings enable use of the even-number address of un-used data registers (D), link registers (W) and motion devices (#).
In addition to the address, the speed, dwell, M code and parameter block can be set indirectly.

(1) Practice conditions

Set the positioning address for axis 1/axis 2 indirectly in servo program No.25 using the ABS-2 command.

- Input the pre-determined position as a two-digit number using the digital switches (X10 to X17).
- The indirect setting device uses 32 bits of D4006, D4007, D4008 and D4009.

(2) Servo program

Indirectly set the address of axis 1 in servo program No.25.

```
[K  25 : REAL  ]
   1 ABS-2
     AXIS                    D 4006    μm
     AXIS                    D 4008    μm
     COMPOSITE VELOCITY     5000.00    mm/min
```

The axis 1 address uses 32 bits of D4006 and D4007.

The axis 2 address uses 32 bits of D4008 and D4009.

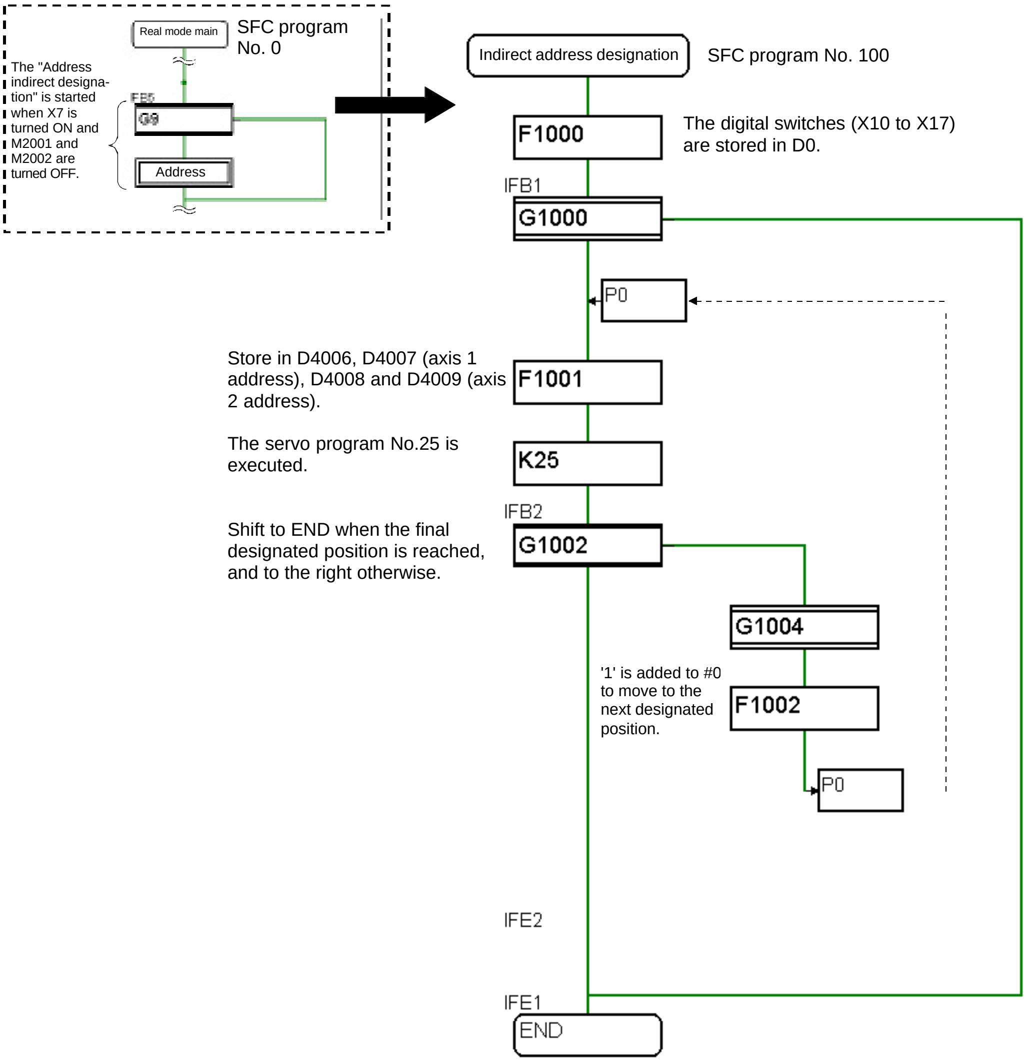
> In addition to the servo command and axis No. indirect settings can be made using the data register (D), link register (W) or motion device (#).

(3) SFC program

Calculate the axis 1 and axis 2 addresses from the digital switch value, and store in the D4006, D4007, D4008 and D4009.

Execute the servo program No.25 using the calculated addresses.

[Real mode main] program



[Transition]	G1000 (#0>=K0)*(#0<=K99)	G1002 (#0 == K99)+PXD+PXE
	G1004 TIME K1500	
[Operation control step]	F1000 DIN D4000L,PX10 D4000L=D4000L&HFF D4000L=BIN(D4000L) #0L=D4000L-K1	F1001 #8L=#0L/K10 #6L=#0L%K10 D4006L=K100000*#6L D4008L=K1800000-(#8L*K200000)
	F1002 #0=#0+K1	
[Motion control step]	K25: REAL 1 ABS-2 AXIS 1, D 4006µm AXIS 2, D 4008µm COMPOSITE VELOCITY 5000.00mm/min	

PX7 : Indirect address designation command PX10 to PX17: Digital switch input device

D4000 : Value of digital switch (designated point)

#0 : Storage of address calculation value (designated point-1)

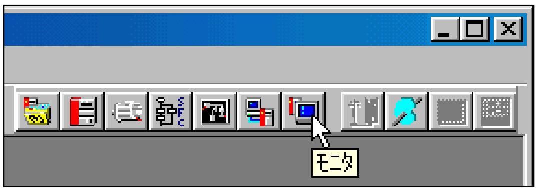
#8, #9 : For calculation of axis 1 address #6, #7 : For calculation of axis 2 address

D4006, D4007: Storage of axis 1 address D4008, D4008: Storage of axis 2 address

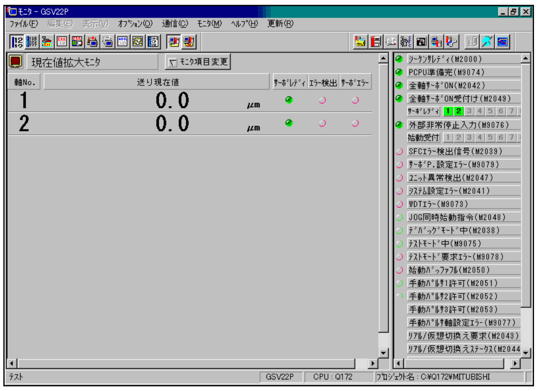
10.6 Operating the practice machine

10.6.1 Operation

The servomotor movement is monitored with the servo monitor using the SW6RN-GSV22P.



1) Click on the monitor tool button on the TEST window.

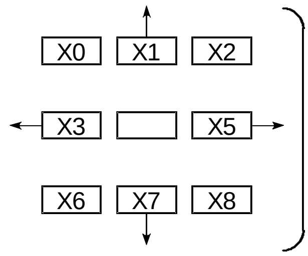


2) The MONITOR window will open and the enlarged current value monitor will appear.

(Continued on next page)

(Continued from previous page)

[Execute JOG operation]



The axis will move with JOG operation while the switch is ON.

Items to check when axis does not move:

- Is the servo ON?
- Is the Q-PLC/Q motion CPU set to RUN?
- Is the personal computer in the test mode.
  (Cancel the mode if in the test mode.)

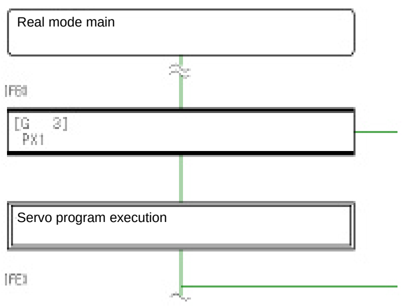
[Execute zero point return]

Set the mode selector switch to [REAL] X19.

- When [X1] is pressed with the digital switch set to [0 0], the X-axis and Y-axis are returned to the zero point.

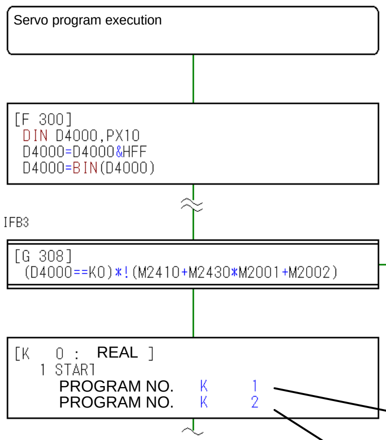
Note: The motor will not rotate during data-set type zero point return.

[Real mode main] program (SFC program No. 0)



The SFC program [Servo program execution] is started when X1 is turned ON.

[Servo program execution] program (SFC program No. 30)



Storage of digital switch value to D4000

Execution of K0 when D4000 is set to 0

Execution of servo program No. 0 (simultaneous start)

Axis 1 zero point return

```
[K  1 : REAL ]
 1  ZERO
    AXIS        1
```

The axis 1 current value will change to –30000.0μm.

Axis 2 zero point return

```
[K  2 : REAL ]
 1  ZERO
    AXIS        2
```

The axis 2 current value will change to –30000.0μm.

(Continued on next page)

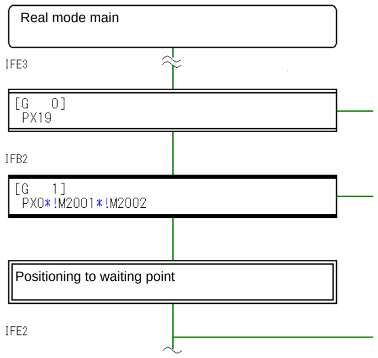
(Continued from previous page)

[Positioning to waiting point]

Set the mode selector switch to [REAL] X19.

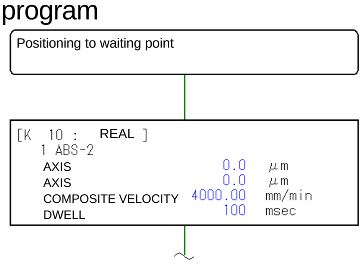
- When X0 is pressed, the axes are positioned to the waiting point (X-axis address '0', Y-axis address '0').

[Real mode main] program (SFC program No. 0)



The SFC program [Positioning of waiting point] is started when X0 is turned ON.

[Waiting point positioning] program (SFC program No. 20)



The servo program No.10 (Positioning "2-axis linear interpolation") is executed.

(Continued on next page)

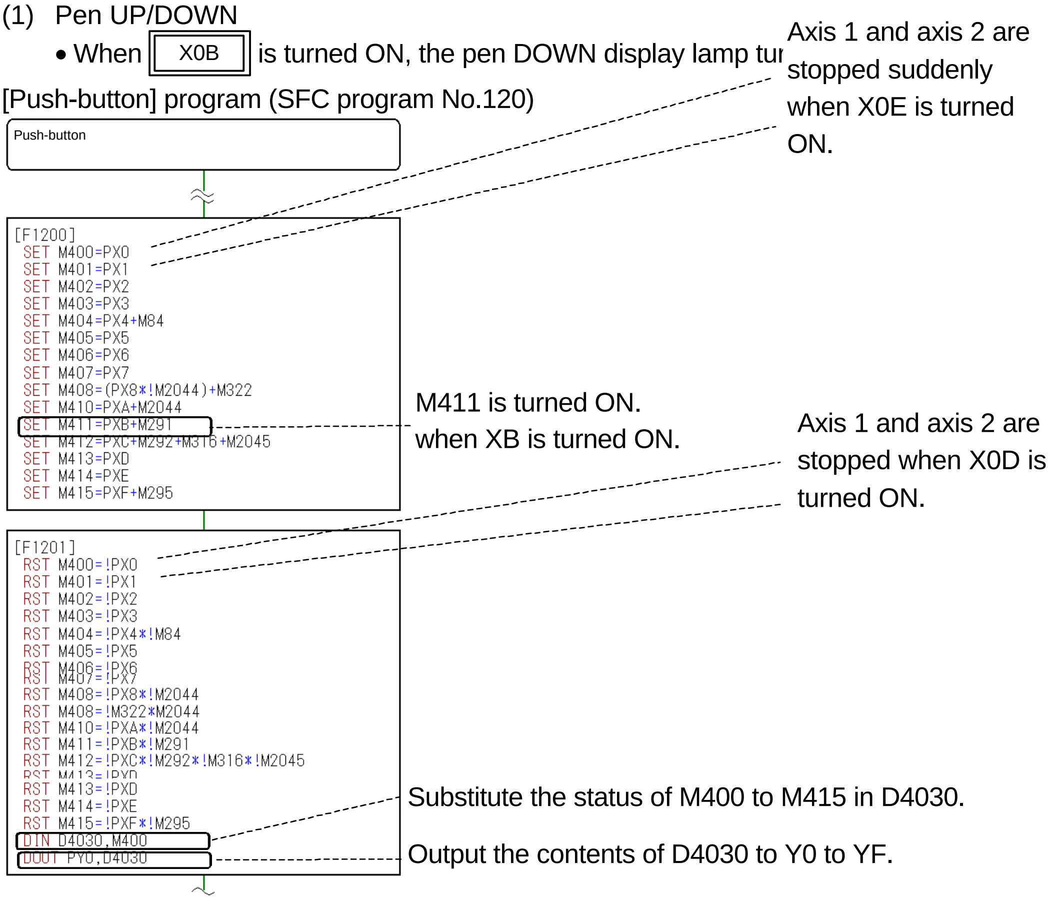
(Continued from previous page)

[Items to confirm during operation]

(1) Pen UP/DOWN

- When X0B is turned ON, the pen DOWN display lamp tur

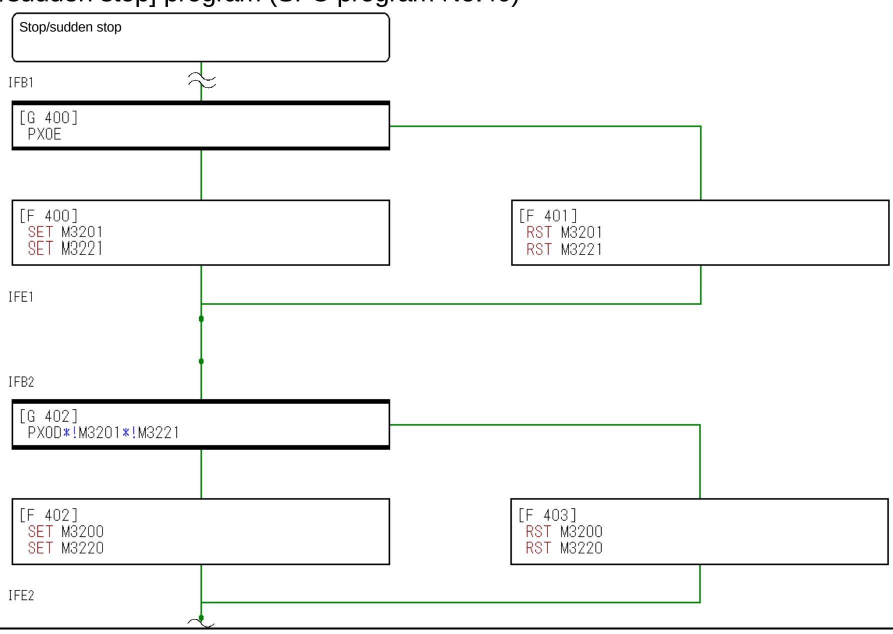
[Push-button] program (SFC program No.120)



(2) Stopping during operation

- The operation is stopped when X0D is turned ON.
- The operation is stopped suddenly when X0E is turned ON.

[Stop/sudden stop] program (SFC program No.40)



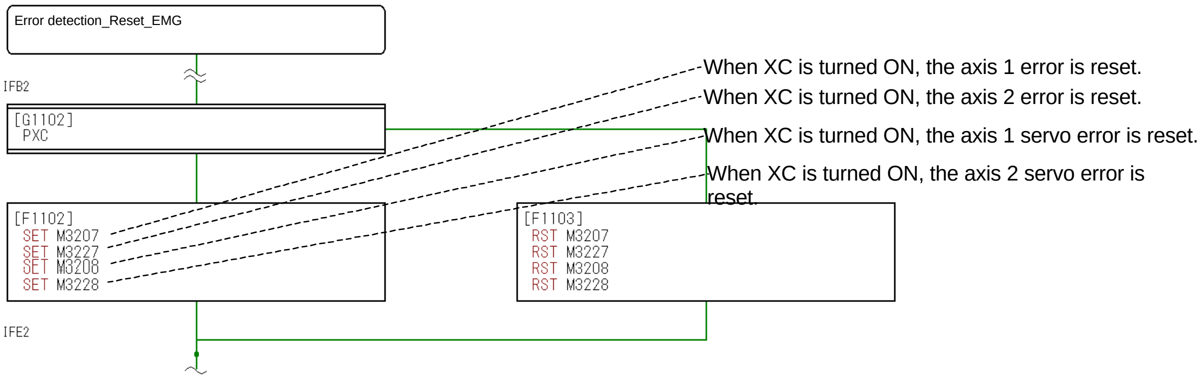
(Continued on next page)

(Continued from previous page)

(3) Error reset

- When [X0C] is turned ON, the occurring error can be reset.

[Error detection_Reset_EMG] program (SFC program No.110)

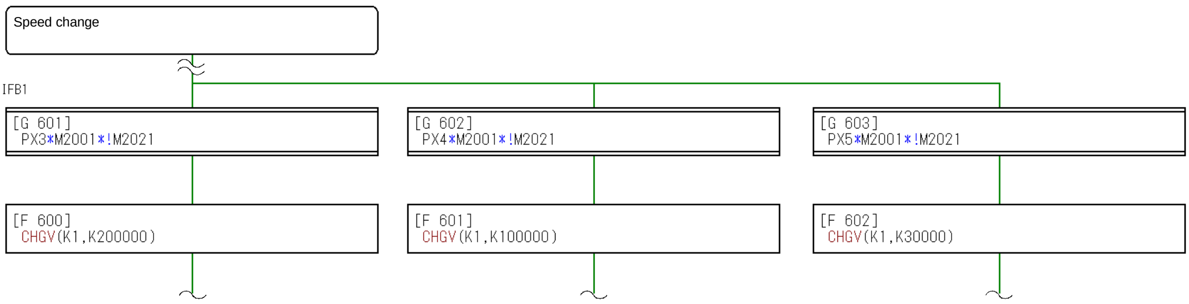


(4) Speed change (operation during continuous positioning, uniform speed control, speed control)

- When [X3] is turned ON, the speed is changed to 2000mm/min.
- When [X4] is turned ON, the speed is changed to 1000mm/min.
- When [X5] is turned ON, the speed is changed to 300mm/min.

(The speed may be changed repeatedly during operation.
However, the speed must not be changed during zero point return, circular interpolation or deceleration. A minor error could result.)

[Speed change] program (SFC program No.60)



The axis 1 speed is changed to 2,000mm/min when X3 is turned ON.

The axis 1 speed is changed to 1,000mm/min when X4 is turned ON.

The axis 1 speed is changed to 300 mm/min when X3 is turned ON.

(5) Current display value change

- When the [X20] switch is turned OFF, the current value of axis 1 is displayed as a [mm] unit on the digital display.
- When the [X20] switch is turned ON, on the other hand, the current value of axis 2 is displayed as a [mm] unit on the digital display.

(The lamps (Y30 to Y33) sequentially turn ON at a 0.5ms cycle when axis 1 or axis 2 is operating.)

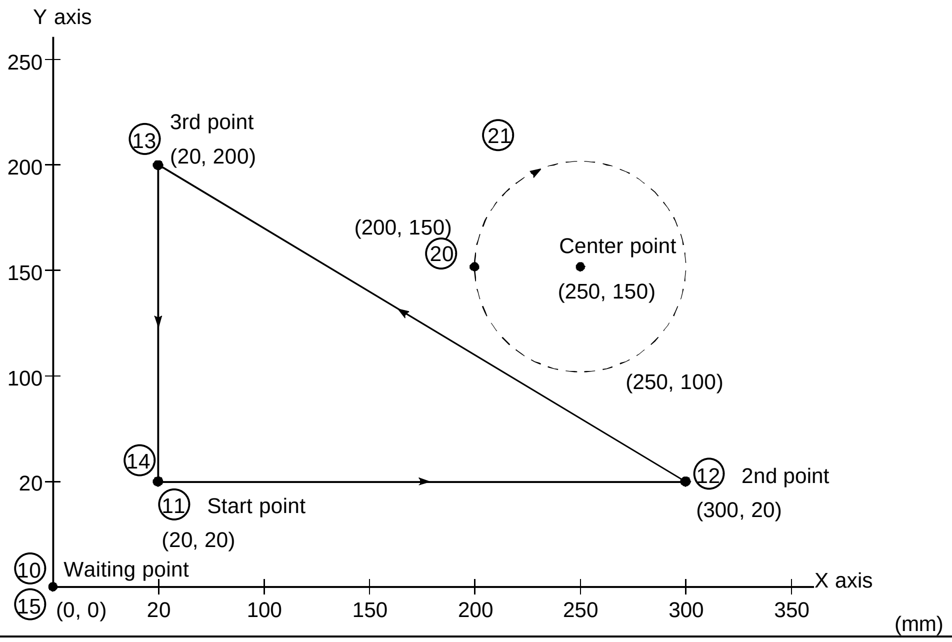
(Continued on next page)

(Continued from previous page)

[Starting continuous positioning]

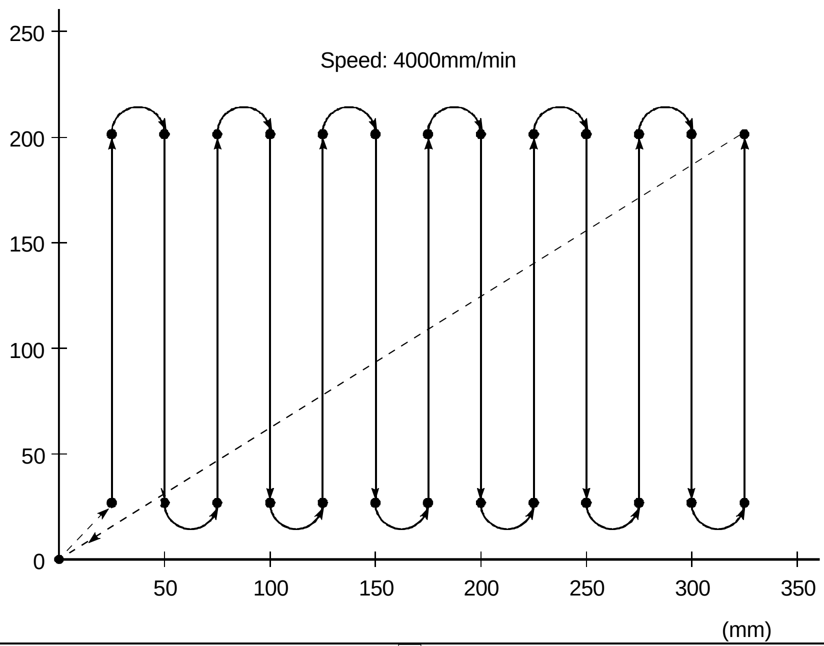
- Positioning will be carried out with the following path when X2 turns ON.
- The M code is detected to raise/lower the pen. (Y2B lamp: Flickering)

The circle number indicates the servo program No.



[Start of servo program No.33 for uniform speed control]

- When X1 is pressed with the digital switch set to 3 3, the uniform speed control is executed with the path shown below.

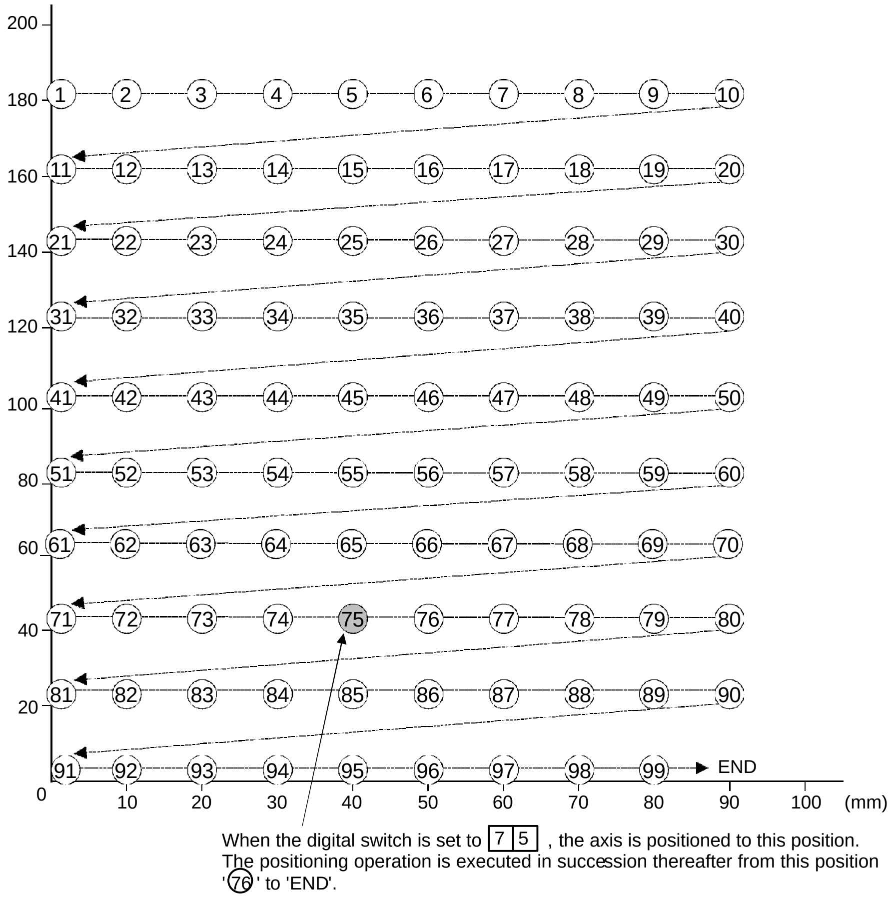


(Continued on next page)

(Continued from previous page)

[Indirect setting of positioning address]

- If X7 is pressed with the digital switch set to 7 5 , the address of '75' is stored in the D4006, D4007, D4008 and D4009 for positioning when the servo program No.25 is executed.
  The positioning control is executed in succession to the position in which the address is changed to '(76), ...' and then to the 'END' position.



The circled No. indicates the designated position No.

(Continued on next page)

(Continued from previous page)

[Speed control]

Set the mode selector switch to [REAL] X19.

- For speed control, the actual current value will be set to zero when starting, and the current value will not increase/decrease during operation.
- Turn ON X1 with the digital switch set to 3 0. Forward run will start.

  Turn ON X0D (stop) or the X0E (sudden stop) to stop the operation.
- Turn ON X1 with the digital switch set to 3 1. Reverse run will start.

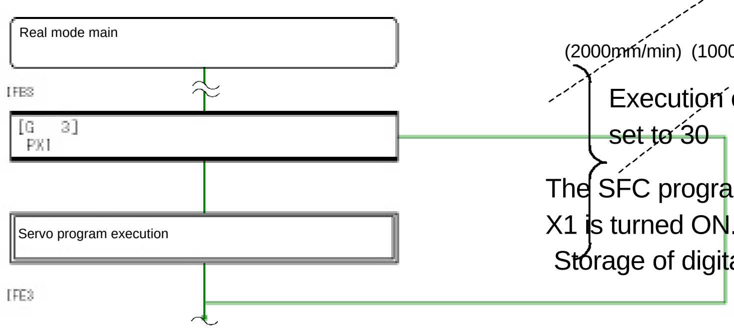
  Turn ON X0D (stop) or the X0E (sudden stop) to stop the operation.
- When the speed change X3, X4 or X5 is turned ON during operation, the speed can be changed.

[Real mode main] program (SFC program No. 0)

(Set to D4000)

(2000mm/min) (1000mm/min) (300mm/min)

(Set to D4000)



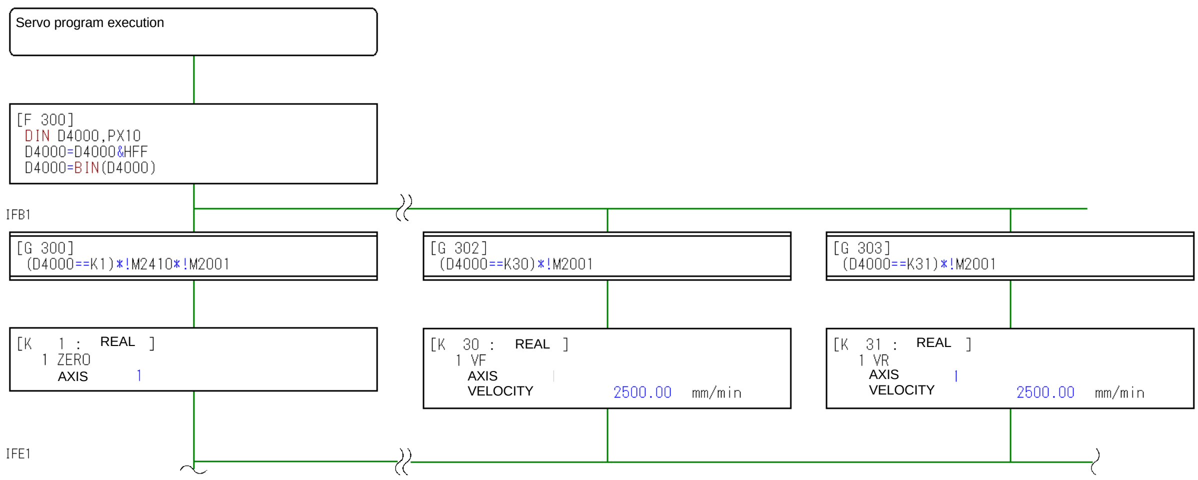
Execution of K30 when D4000 is set to 30

The SFC program [Servo program execution] is started when X1 is turned ON.

Storage of digital switch value to D4000

[Servo program execution] program (SFC program No. 30)

Execution of K31 when D4000 is set to 31



Execution of servo program No.30 (forward speed)

Execution of servo program No.31 (reverse speed)

- When the speed control is executed, the address will be set to zero so zero point return will be executed.

  Press X1 with the digital switch set to 0 0 .

Note: For data-set type zero point return, the motor will not rotate.

End of operation

Memo

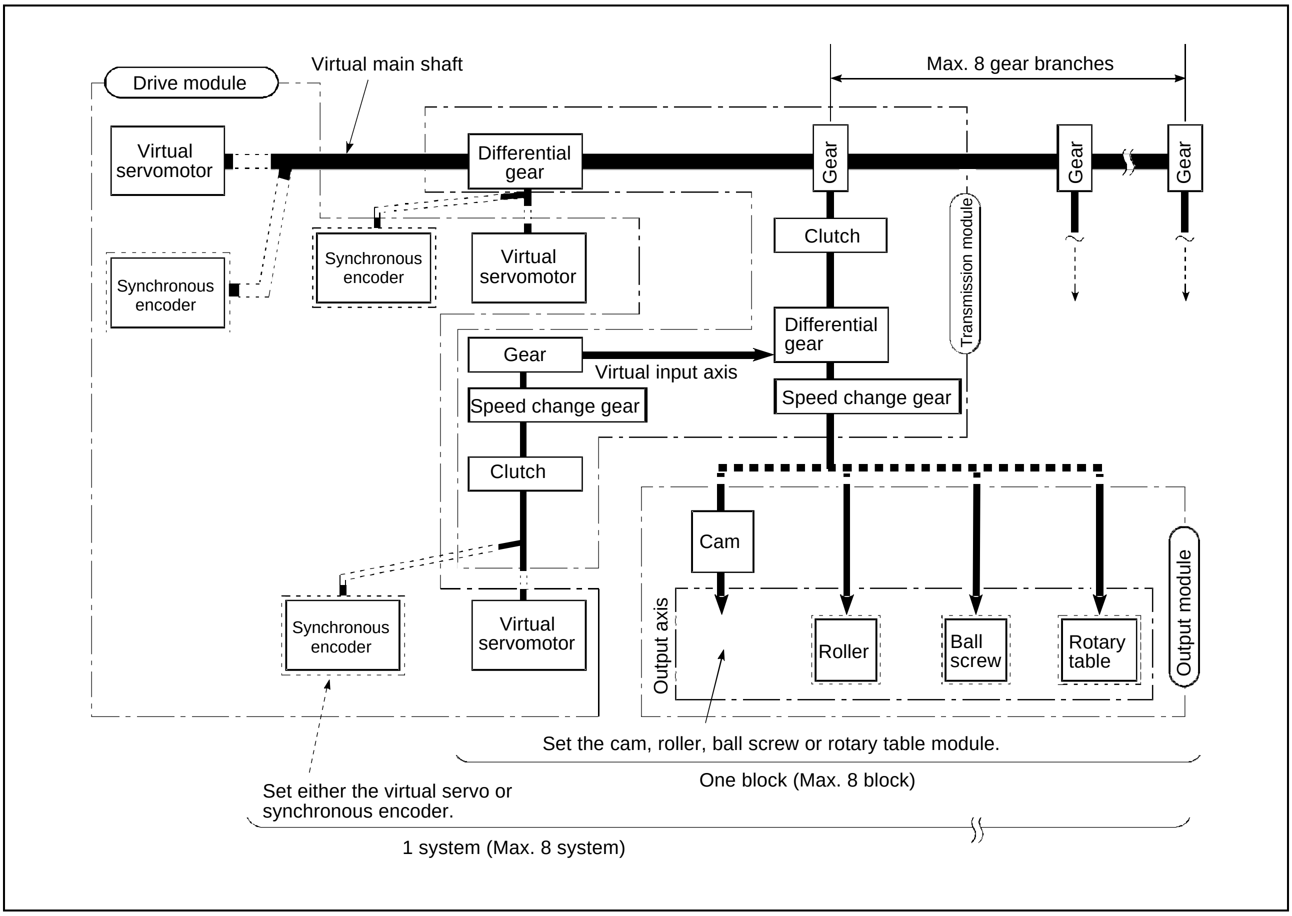
Chapter 11 Practicing with the SV22 Virtual Mode

11.1 Mechanism program

The mechanism program used for control in the virtual mode is configured of the mechanism module connection diagram and mechanism module parameters.

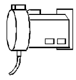
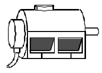
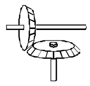
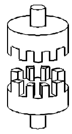
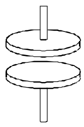
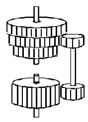
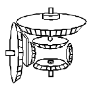
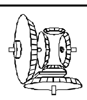
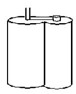
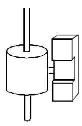
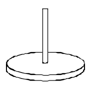
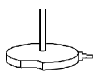
11.1.1 Mechanism module connection diagram

This is the virtual mechanism system diagram created by arranging virtual mechanism modules on the screen. (The following diagram is for the Q172.)



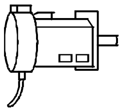
11.1.2 List of mechanism modules

The number of mechanism modules that can be used in the mechanism module connection diagram for the virtual mode is shown below. (The quantity for the Q172 is shown.)

Mechanism class	Mechanism module		Usable quantity						
	Name	Appearance	Quantity per CPU		Quantity per system		Quantity per block: Connection axis side	Quantity per block: Auxiliary input axis side	
Drive module	Virtual servomotor		8	Total 16	8	Total 16	–	–	–
	Synchronous encoder		8		8		–	–	
Virtual axis	Virtual main shaft	–	8	Total 16	1		–	–	
	Virtual auxiliary input axis	–	8		8		–	–	
Transmission module	Gear		16		16		1	1	
	Direct clutch		16		16		1	1	
	Smoothing clutch								
	Speed change gear		16		16		1	1	
	Differential gear		8		8		1	–	
					1		–		
Output module	Roller		8	Total 8	8	Total 8	1	1	
	Ball screw		8		8				
	Rotary table		8		8				
	Cam		8		8				

11.1.3 Virtual servomotor

The virtual servomotor is used to operate the virtual axis with the servo program or JOG operation.



No.	Parameter setting item	Default value	Setting range
1	Virtual axis No.	———	1 to 8
2	Stroke upper limit value	$2^{31}-1$ pulse	-2^{31} to $2^{31}-1$ pulse
3	Stroke lower limit value	0 pulse	-2^{31} to $2^{31}-1$ pulse
4	Command in-position range	100 pulse	1 to 32767 pulse
5	JOG parameter block	1	1 to 64
6	JOG speed limit value	20000 pulse/s	1 to 1000000 pulse/s
7	Operation mode during error	Intermittent	Intermittent/clutch OFF

11.1.4 Synchronous encoder

The synchronous encoder is used to operate the virtual axis with pulses input from the external encoder (A, B phase type can be connected; serial type cannot be connected).

No.	Parameter setting item	Default value	Setting range
1	Encoder No.	———	1 to 3
2	Operation mode during error	Intermittent	Intermittent/clutch OFF

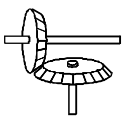
Servo input unit connection positions:
P1/E1 → 1
P2/E2 → 2
P3/E3 → 3
P4/E4 → 4
P5/E5 → 5
P6/E6 → 6
P7/E7 → 7
P8/E8 → 8

11.1.5 Virtual axis

The virtual axes include the virtual main shaft and virtual auxiliary input axis. The rotation of the drive module is conveyed to the conveyance module.

There are no parameters to be set for the virtual axis.

11.1.6 Gears



The gears are used to convey, to the output shaft, the number of pulses obtained by multiplying the movement amount (number of pulses) of the drive module (virtual servomotor, synchronous encoder) by the gear ratio set in the gear parameters, and to determine the output shaft rotation direction.

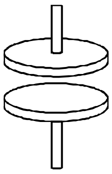
No.	Parameter setting item		Default value	Setting range
1	Gear ratio	Number of input axis gear teeth (GI)	1	1 to 65535
		Number of output axis gear teeth (GO)	1	1 to 65535
2	Output rotation direction (GS)		Forward	Forward Reverse

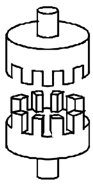
Note) The number of teeth can be indirectly designated with D and W.

(These parameters cannot be changed during operation.)

11.1.7 Clutch

The smoothing clutch and direct clutch can be used.
The control includes the ON/OFF mode (X, Y, M, L, B, F), address mode (D, W) and external input mode (TREN tracking enable signal of Q172EX serial ABS synchronous encoder input unit).
The address mode is always used together with the ON/OFF mode. In addition, the address clutch reference setting (M3213/Axis 1) must be turned and the reference must be set.

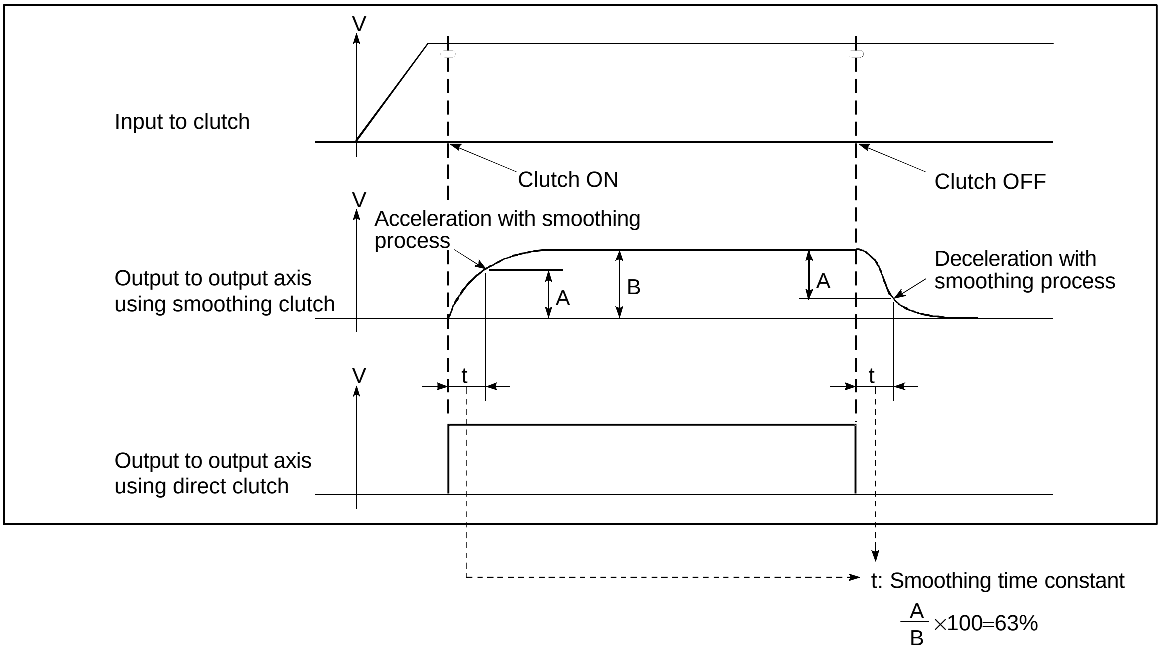




No.	Parameter setting item	Default value	Setting range		
1	Control mode	ON/OFF mode	ON/OFF mode	ON/OFF mode and address mode combination	ON/OFF mode and external input mode combination
2	Mode setting device (1-word)	–	–	Word device D,W	
3	Clutch ON/OFF designation device	–	Bit device X, Y, M, L, T, C, B, F		
4	Clutch ON address setting device (2-word)	–	–	Even number of word devices D, W	–
5	Clutch OFF address setting device (2-word)				
6	Clutch status storage device	–	———		
7	Clutch type	Direct clutch	Direct clutch/smoothing clutch		
8	Smoothing method	Time constant designation	Time constant designation/slip amount designation		
9	Smoothing time constant	0	0 to 65535ms		
10	Smoothing amount setting device (2-word)	–	Even number of word devices D, W		

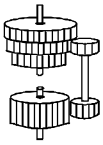
(a) Smoothing clutch
When the clutch turns ON/OFF, the acceleration/deceleration process (smoothing process) set with parameter No. 8 to 10 is carried out, and the control is output to the output shaft.

(b) Direct clutch
When the clutch turns ON/OFF, as nothing is set in parameter No. 8 to 10, the smoothing acceleration/deceleration process is not carried out, and the control is output to the output shaft.



11.1.8 Transmission

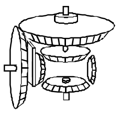
To lower the roller output speed, the transmission conveys, to the output shaft, the speed obtained by multiplying the input axis speed with the transmission ratio set in the speed ratio setting device. (The gears are used to increase the speed.)



No.	Parameter setting item	Default value	Setting range
1	Transmission ratio upper limit value	10000	1 to 10000
2	Transmission ratio lower limit value	1	1 to 10000
3	Transmission ratio setting device (1-word)	–	Word device D, W
4	Smoothing time constant	0	0 to 65535 (ms)

(It is possible to change it from motion SFC during operation.)

11.1.9 Differential gears

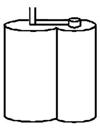


• Used to deviate the phase of output module or to adjust the operation start position.
• Independent operation, separated from the virtual main shaft, is possible.

To realize the above operations, the differential gears subtract the auxiliary input axis movement amount from the input axis movement amount, and convey the results to the output axis.

There are no parameters to be set for the differential gears.

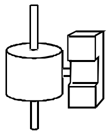
11.1.10 Rollers



The rollers are controlled at the speed obtained by multiplying the conveyance module's gear ratio and transmission ratio with the drive module's speed.
These are used when the final output is speed control.

No.	Parameter setting item	Default value	Setting range	
1	Output axis No.	0	1 to 8	
2	Unit setting	mm	mm	inch
3	Roller diameter (L)	0	0.1 to 214748364.7μm	0.00001 to 21474.83647inch
4	No. of pulses per roller rotation (NL)	0	1 to 2147483647 pulse	
5	Tolerable droop pulse value	65535	1 to 65535 pulse	
6	Speed limit value (VL)	0	0.01 to 6000000.00 mm/min	0.001 to 600000.000 inch/min
7	Torque limit value setting device (1-word)	–	– (300%) / Word device D, W	
8	Comment	None	16 kanji characters	

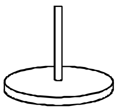
11.1.11 Ball screw



The ball screw outputs the movement amount obtained by multiplying the drive module's movement amount with the conveyance module's gear ratio.
This is used when the final output is linear positioning.

No.	Parameter setting item	Default value	Setting range	
1	Output axis No.	0	1 to 8	
2	Unit setting	mm	mm	inch
3	Ball screw pitch (P)	0	0.1 to 214748364.7	0.00001 to 21474.83647inch
4	No. of pulses per ball screw rotation (NP)	0	1 to 2147483647 pulse	
5	Tolerable droop pulse value	65535	1 to 65535 pulse	
6	Stroke limit upper limit value	$2^{31}-1$	–214748364.8 to 214748364µm	–21474.83648 to 21474.8364inch
7	Stroke limit lower limit value	0		
8	Speed limit value (VL)	–	0.01 to 6000000.00 mm/min	0.001 to 600000.000 inch/min
9	Limit switch output	Not used	ON/OFF	
10	Torque limit value setting device (1-word)	–	– (300%) / Word device D, W	
11	Comment	None	16 kanji characters	

11.1.12 Rotary table

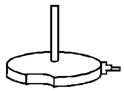


The rotary table outputs the movement amount obtained by multiplying the drive module movement amount with the conveyance module's gear ratio.
This is used when the final output is angle control.

No.	Parameter setting item	Default value	Setting range
1	Output axis No.	0	1 to 8
2	No. of pulses per rotary table rotation (ND)	–	1 to 2147483647 (pulse)
3	Tolerable droop pulse value	65535	1 to 65535 (pulse)
4	Stroke limit upper limit value	0	0 to 359.99999 (degree)
5	Stroke limit lower limit value	0	0 to 359.99999 (degree)
6	Speed limit value (VL)	0	0.01 to 6000000.00 (degree/min)
7	Limit switch output	Not used	ON/OFF
8	Torque limit value setting device (1-word)	–	– (300%) / Word device D, W
9	Comment	None	16 kanji characters
10	Current value per virtual axis rotation storage device, main shaft side (2-word)	–	– / Word device D, W even No.
11	Current value per virtual axis rotation storage device, auxiliary input axis side (2-word)	–	– / Word device D, W even No.

Note) The current value per virtual axis rotation is valid when using the clutch in the address mode.

11.1.13 Cam



The cam carries out cam output based on the cam stroke and cam curve data created with SW3RN-CAMP, and outputs the movement amount obtained by multiplying the drive module movement amount with the conveyance module's gear ratio.
This is used when the final output is reciprocating cam control or feed cam control.

No.	Parameter setting item	Default value	Setting range		
1	Output axis No.	0	1 to 8		
2	No. of pulses per cam axis rotation (Nc)	0	1 to 2147483647 (pulse)		
3	Applicable cam No.	–	–		
4	Cam No. setting device (1-word)	–	Word device D, W		
5	Tolerable droop pulse value	65535 (pulse)	1 to 65535 (pulse)		
6	Unit setting	mm	mm	inch	pulse
7	Stroke amount setting device (2-word)	–	Word device D, W even No.		
8	Limit switch output	Not used	ON/OFF		
9	Torque limit value setting device (1-word)	–	– (300%) / Word device D, W		
10	Comment	None	16 kanji characters		
11	Stroke lower limit value storage device (2-word)	–	– / Word device D, W even No.		
12	Current value per virtual axis rotation storage device, main shaft side (2-word)	–	– / Word device D, W even No.		
13	Current value per virtual axis rotation storage device, auxiliary input axis side (2-word)	–	– / Word device D, W even No.		

Note) The current value per virtual axis rotation is valid when using the clutch in the address mode.

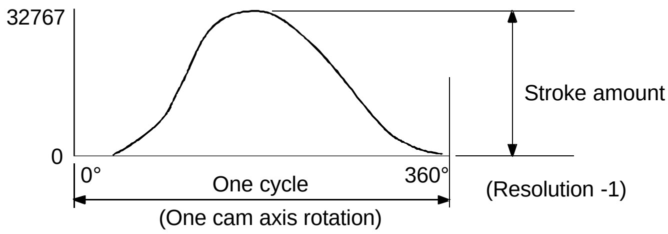
[Cam data created with SW3RN-CAMP]

The cam data is stored in the cam data dedicated internal memory in the motion CPU.

No.	Parameter setting item	Default value	Setting range
1	Cam No.	–	1 to 64
2	Resolution (3.5ms)	256	256, 512, 1024, 2048
3	Stroke amount, cam No., changeover position	0	0 to (resolution -1)
4	Control mode	Reciprocating cam mode	• Reciprocating cam mode • Feed cam mode
5	Cam data table (Stroke setting)	100	Differs according to the control unit: %, mm, inch, PULSE
6	Cam curve	–	Selection

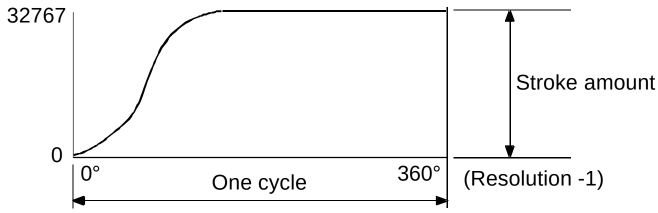
Reciprocating cam

Extrusion/return operation
Vertical operation
Left/right operation



Feed cam

Conveyor feed
Transfer machine feed



(The stroke amount is a position command value, and is not the speed. A minus value cannot be set.)

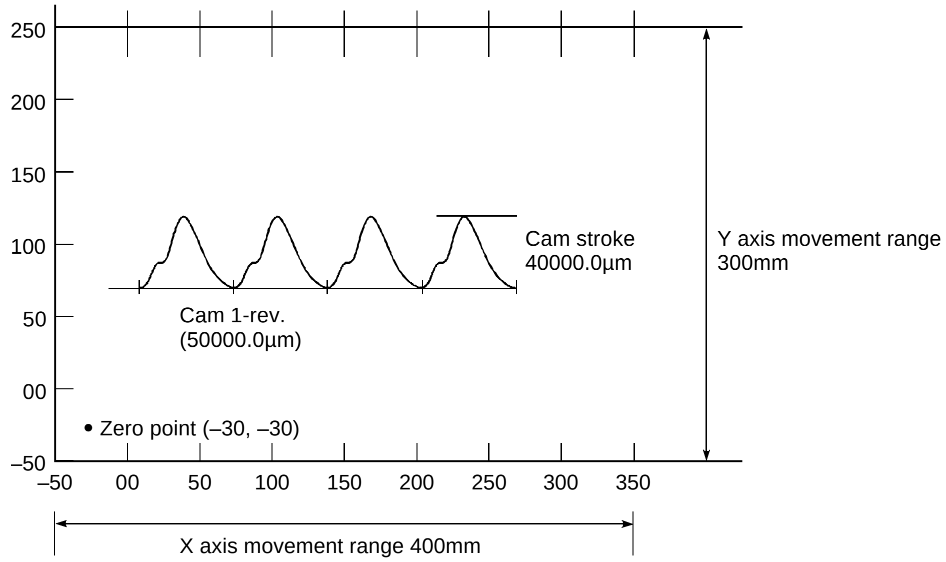
Cam curve table

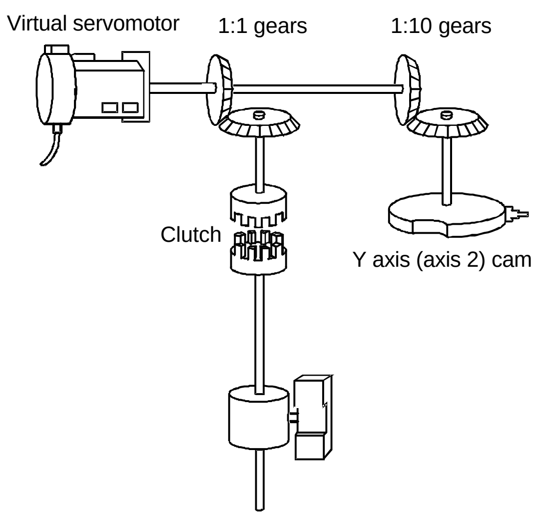
Class		Cam curve name	Acceleration curve shape	Vm	Am	(A•V)m	(V•V)m	(S•V)m	Remarks
Non-continuous curve		Uniform speed		1.00	–	–	1.00	1.00	
		Uniform acceleration		2.00	±4.00	±8.00	4.00	1.09	
Dwell-rise-dwell curve	Symmetrical curve	5th power		1.88	±5.77	±6.69	3.52	1.19	
		Cycloidal		2.00	±6.28	±8.16	4.00	1.26	
		Modified trapezoid		2.00	±4.89	±8.09	4.00	1.20	Ta = 1/8
		Modified sine		1.76	±5.53	±5.46	3.10	1.13	Ta = 1/8
		Modified constant speed		1.28	±8.01	±5.73	1.63	1.07	Ta = 1/16 Ta = 1/4
	Unsymmetrical curve	Trapechloid		2.18	±6.17	±10.84	4.76	1.28	m = 1
One-dwell curve		Multi-chord		2.04	+5.55 -9.87	+7.75 -9.89	4.16	1.39	
No-dwell curve		Harmonic		1.57	±4.93	±3.88	2.47	1.02	

V ... Dimensionless speed, A ... Dimensionless acceleration, S ... Dimensionless displacement, m ... Maximum value

11.2 Details of practice

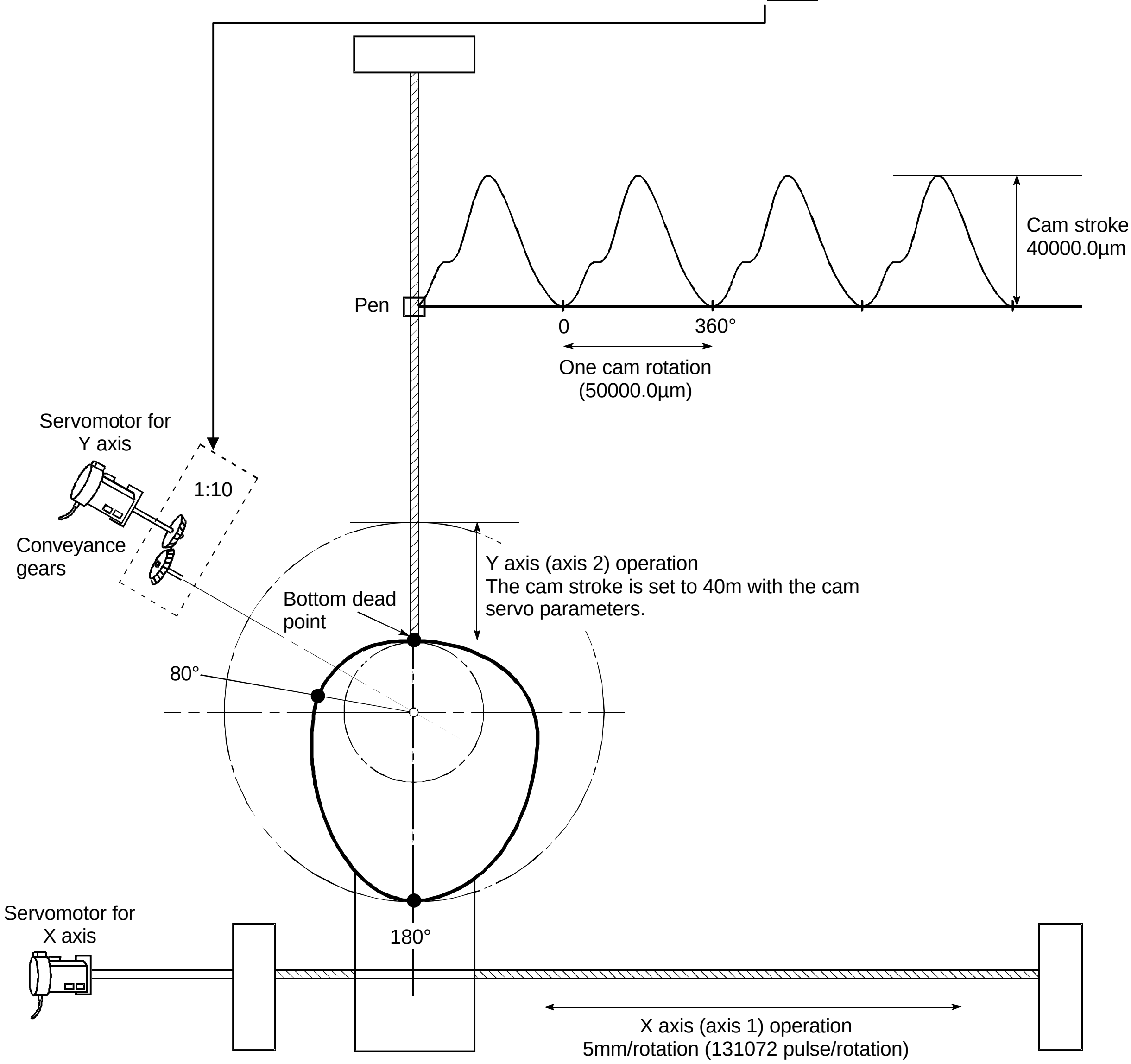
The X axis (axis 1) and Y axis (axis 2) are synchronously operated using the mechanical support language.
The X axis (axis 1) carries out left/right reciprocation with the ball screw, and the Y axis (axis 2) carries out forward/backward reciprocation with the cam output.





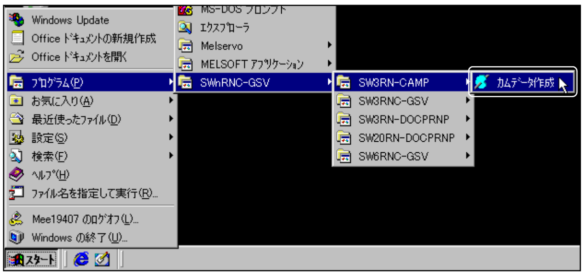
Ideology for moving along path

- The X axis (axis 1) ball screw is set to 5mm/rotation (131072 pulse/rotation), so the axis 1 output module is set as the "ball screw", and the No. of pulses per rotation is set to 131072 pulse with the ball screw parameter.
- The Y axis (axis 2) ball screw is also set to 5mm/rotation (131072 pulse/rotation), so to establish a 50mm/rotation movement amount rotary cam in the X axis direction, the No. of pulses per rotation is set to 131072 pulses with the cam parameter, and the gear ratio is set to 1:10.



Thus, to rotate the cam four times, 200mm (50mm × 4-fold) is set, and as the ball screw lead is 5mm, the X axis (axis 1) servomotor will rotate 40 times.

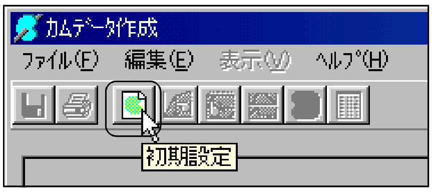
11.3 Starting up SW3RN-CAMP and creating the cam



1) Click on [Start], [Program], [SWnRNC-GSV], [SW3RN-CAMP] and then [Cam data creation].

2) The CAM DATA CREATION window will open, so click on [File], and then the [New creation] menu.

3) The dialog box to confirm the new creation will open, so click on the OK button.



4) Click on the initial setting tool button.

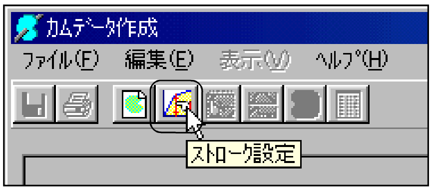
5) The INITIAL SETTING dialog box will open, so set as follows.

- "Cam mode" : 'Reciprocating'
- "Resolution" : '256'
- "Curve type" : 'Cam curve'
- "Unit" : '%'
- "Stroke rate change position" : '0'

Click on the Setting completed button.

(Continued on next page)

(Continued from previous page)



6) Click on the Stroke setting tool button.

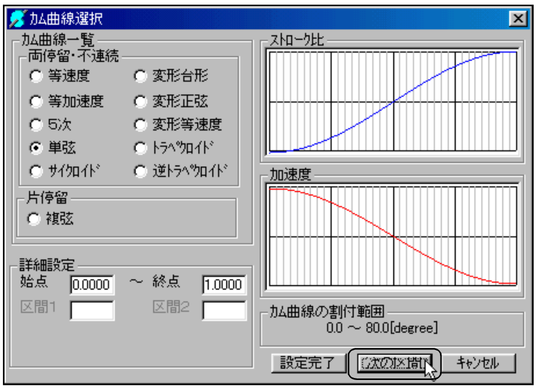
7) Set in the STROKE SETTING dialog box as shown in the following table.

Area No.	Start angle	End angle	Stroke
1	0.0	80.0	30.00
2		180.0	100.00
3		0.0	0.00

Stroke setting range "Minimum value": 0.00, "Maximum value": 100.00

Click on the Stroke setting completed button.

8) Click on the OK button.



9) Set the "Cam curve list" to 'Harmonic' in the CAM CURVE SELECTION dialog box. (The cam curve for area No.1 is selected.)
For area No.2 and No.3, click on the Next area button in succession and select 'Harmonic'.
After setting, click on the Setting completed button.

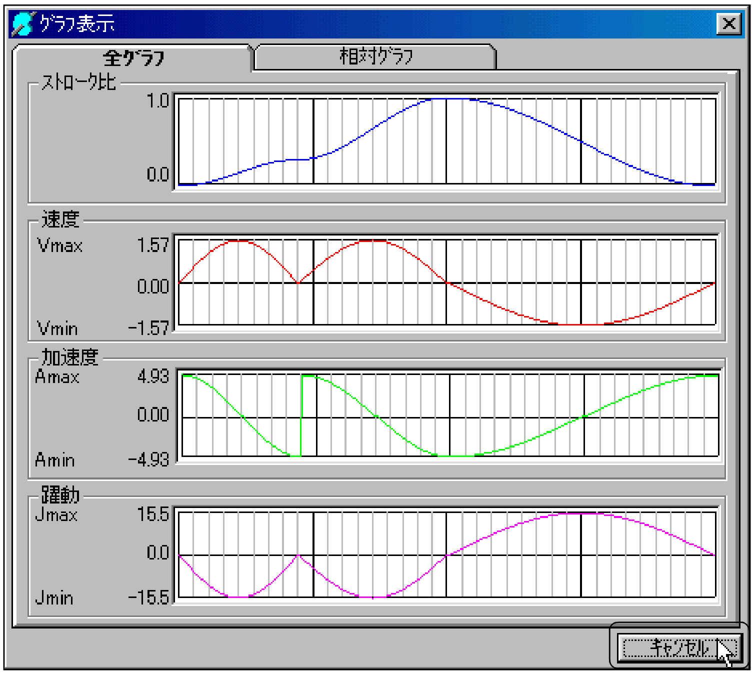
10) Click on the Cam curve selection completed button.

(Continued on next page)

(Continued from previous page)

11) Click on the OK button.

12) To see the [Stroke ratio], [Speed], [Acceleration] and [Saltarion] shown in the table for the operation angle, click on the graph display tool button.



After confirmation, click on the Cancel button.

13) To see the [Stroke ratio], [Speed], [Acceleration] and [Saltarion] for the operation angle as a value, click on the graph display tool button.

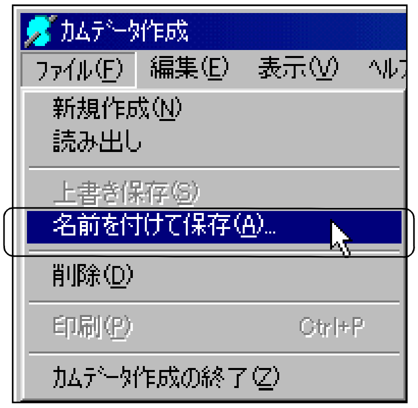
(Continued on next page)

(Continued from previous page)

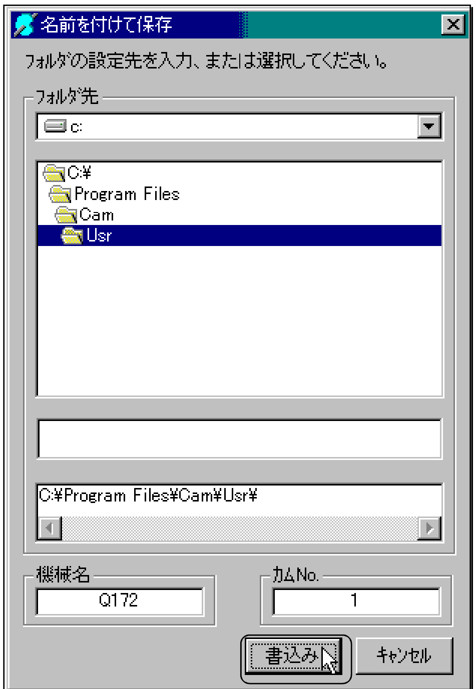
テーブルNo	絶対角	ストローク比	速度	加速度	躍動	カム曲線	カムデータ
0	0.00	0.00000	0.00	4.93	0.0	単弦	0000
1	1.41	0.00023	0.09	4.93	-0.9	単弦	0007
2	2.81	0.00091	0.17	4.90	-1.7	単弦	001E
3	4.22	0.00205	0.26	4.87	-2.6	単弦	0043
4	5.63	0.00364	0.34	4.81	-3.4	単弦	0077
5	7.03	0.00568	0.43	4.75	-4.2	単弦	00BA
6	8.44	0.00816	0.51	4.67	-5.0	単弦	010B
7	9.84	0.01107	0.59	4.57	-5.8	単弦	016B
8	11.25	0.01440	0.67	4.46	-6.6	単弦	01D8
9	12.66	0.01815	0.75	4.34	-7.4	単弦	0253
10	14.06	0.02230	0.82	4.20	-8.1	単弦	02DB
11	15.47	0.02683	0.90	4.05	-8.8	単弦	036F
12	16.88	0.03175	0.97	3.89	-9.5	単弦	0410
13	18.28	0.03702	1.03	3.72	-10.2	単弦	04BD
14	19.69	0.04264	1.10	3.53	-10.8	単弦	0575
15	21.09	0.04859	1.16	3.34	-11.4	単弦	0638
16	22.50	0.05484	1.21	3.13	-12.0	単弦	0705
17	23.91	0.06139	1.27	2.92	-12.5	単弦	07DB

After confirmation, click on the Cancel button.

The table is arranged from No.0 to No.255, and can be displayed by scrolling.



14) To save the set cam data, click on [File] and then the Save as] menu in the CAM DATA CREATION window.

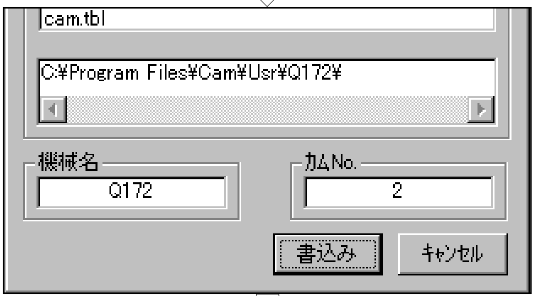


15) Set the "Machine name" to 'Q172', and the "Cam No." to '1' in the SAVE AS dialog box, and then click on the Write button.

(Continued on next page)

(Continued from previous page)

16) Create the cam data for cam No.2 with the same procedure as for cam No.1.
For cam No.2, change the "Cam curve selection" of cam No.1, to 'Uniform speed'.
(The other items are the same as cam No.1.)
Select [Edit] and then [Cam curve selection], and set the full-stroke to 'Uniform speed'.



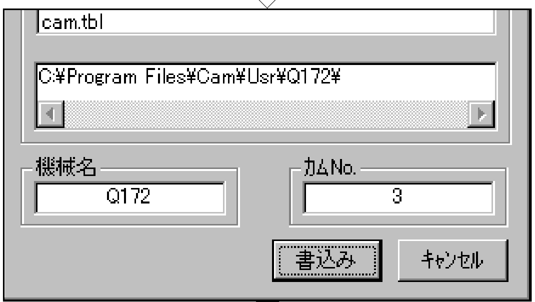
17) When saving the data with 'Save as', set the "Cam No." to '2'.

18) Set the cam No.3 with the same procedure.
Set the "Stroke setting" as shown below.

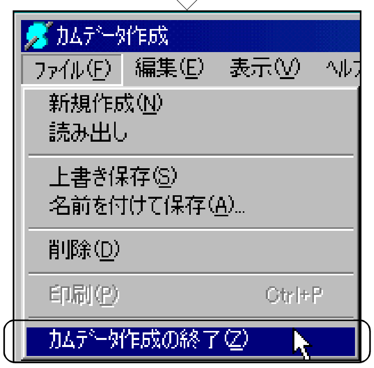
Area No.	Start angle	End angle	Stroke
1	0.0	80.0	30.00
2		150.0	100.00
3		220.0	100.00
4		310.0	0.00
5		0.0	0.00

Stroke setting range "Minimum value": 0.00, "Maximum value": 100.00

Set the "Stroke setting range" to 'Double circle'.



19) When saving the data with 'Save as', set the "Cam No." to '3'.



20) Click on [File], and then the [Cam data creation completed] menu.
This completes creation of the cam data.

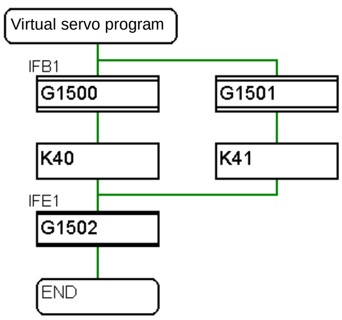
11.4 SFC program for virtual mode

The following lists the SFC programs in the virtual mode.

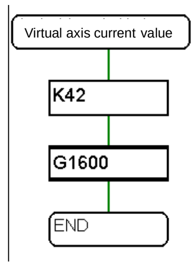
No.	Program name	Automatic start	END operation	Number of shifts	Execution timing
130	Virtual mode main	No			Normal
140	Virtual mode JOG operation	Yes			Normal
150	Virtual servo program	No			Normal
160	Virtual axis current value change	No			Normal
170	Virtual stop/sudden stop	Yes			Normal
180	Virtual error detection	Yes			Normal
190	Cam change	Yes			Normal
200	Clutch ON/OFF	Yes			Normal

• Program started by SFC program

[Virtual servo program] program No.150
Started by No. 130

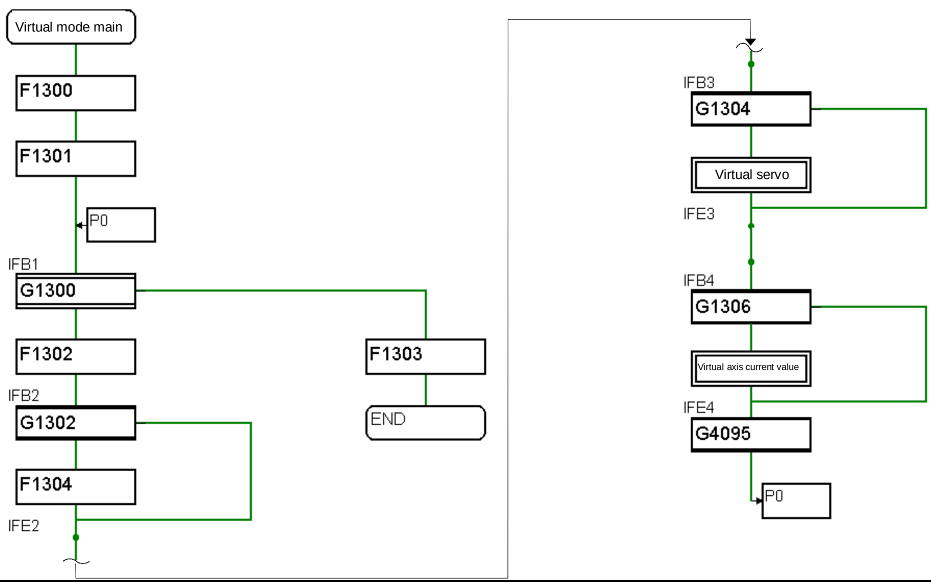


[Virtual axis current value change] program No.160
Started by No. 130



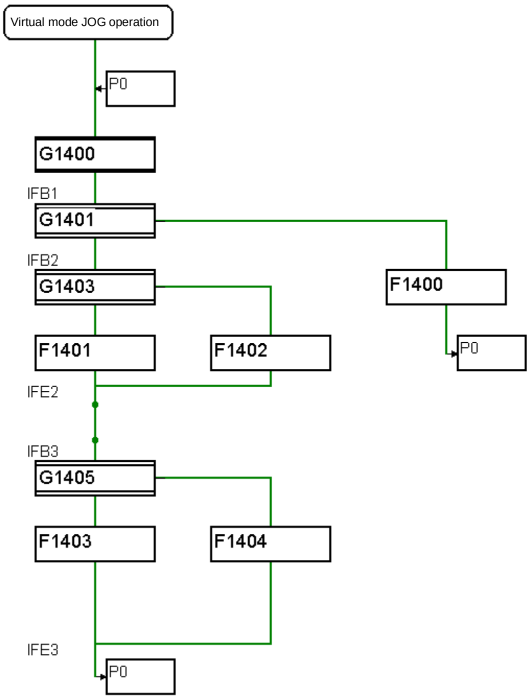
• Program started by sequence program

[Virtual mode main] program No.130
Started by sequence program

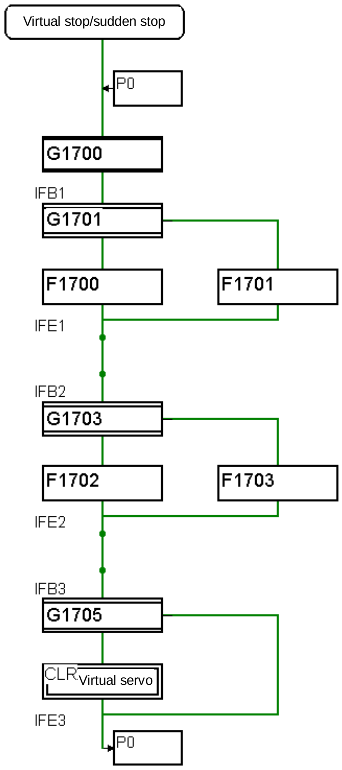


- **Normal execution program**

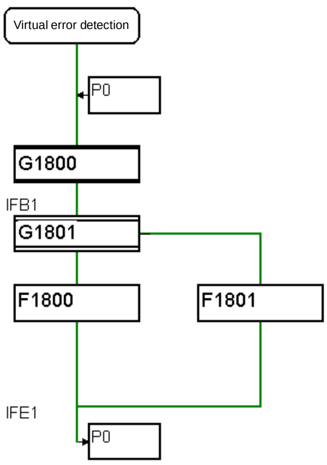
[Virtual mode JOG operation] program No.140
Started automatically.



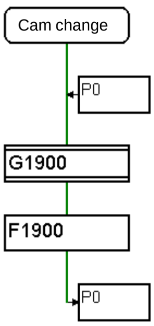
[Virtual stop/sudden stop] program No.170
Started automatically.



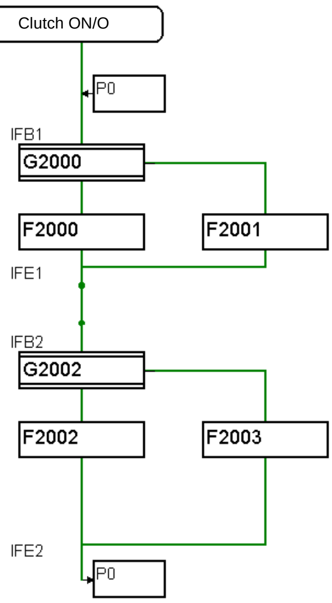
[Virtual error detection]
program No. 180
Started automatically.



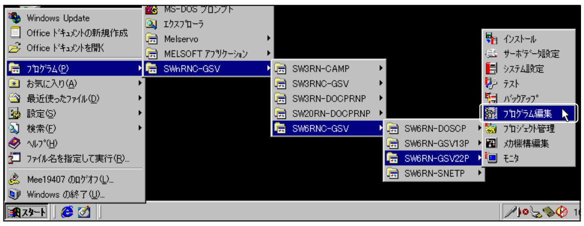
[Cam change]
program No. 190
Started automatically.



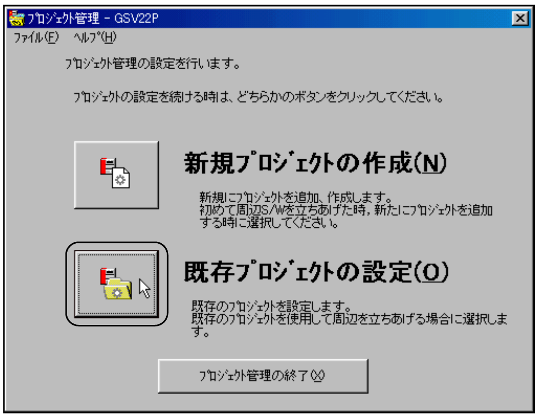
[Clutch ON/OFF]
program No. 200
Started automatically.



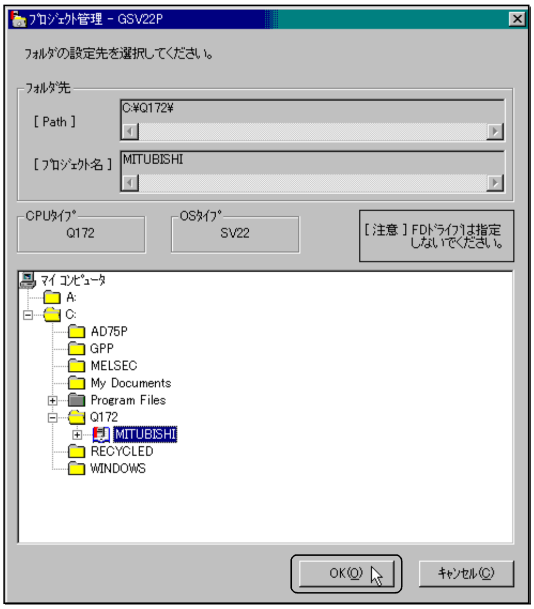
11.4.1 New creation of SFC program for virtual mode



1) Click on [Start], [Program], [SWnRNC-GSV], [SW6RNC-GSV], [SW6RN-GSV22P] and then the [Program edit].



2) The PROJECT CONTROL dialog box will open, so click on the EXISTING PROJECT SETTING button.



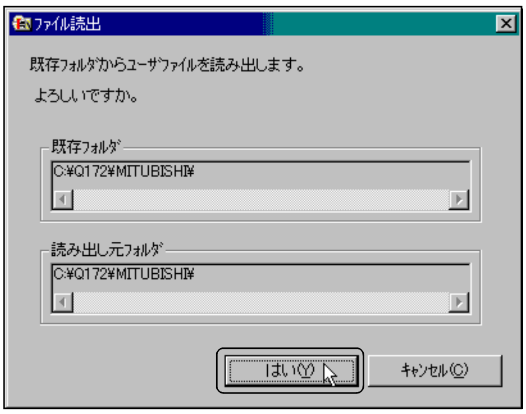
3) Check that the "Path" is set to 'C:\Q172', and that the "Project name" is the same as the project name set in the real mode. Then, click on the OK button.

(Continued on next page)

(Continued from previous page)

4) Check that the folder for setting the user file is the project folder set in the real mode, and click on the OK button.

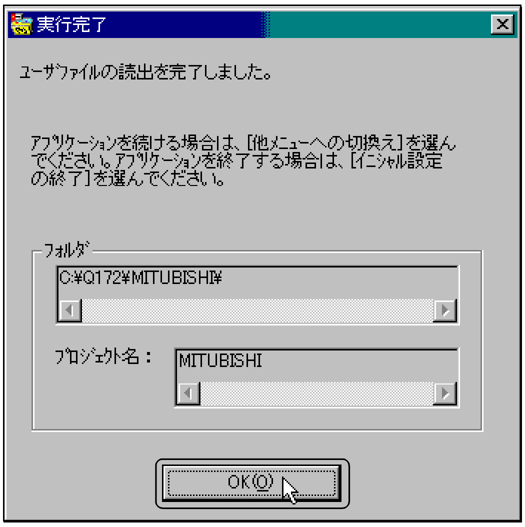
5) Click on the FILE READ button.



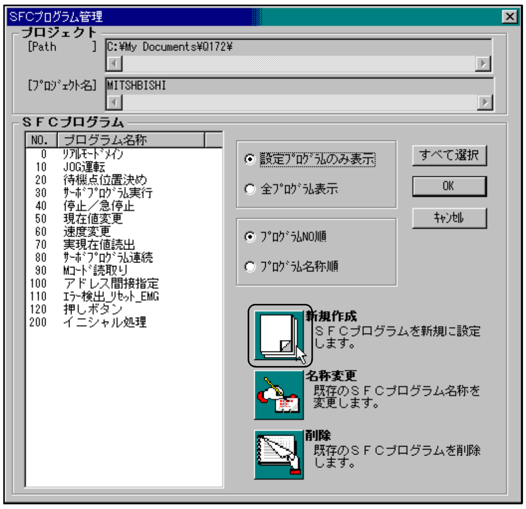
6) Click on the YES button.

(Continued on next page)

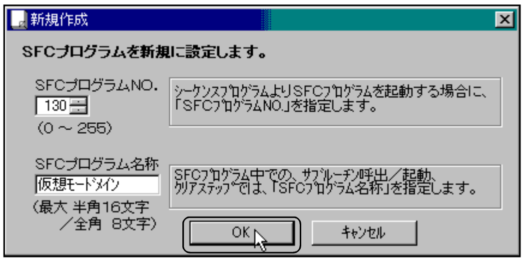
(Continued from previous page)



7) Click on the OK button in the EXECUTION COMPLETED dialog box.



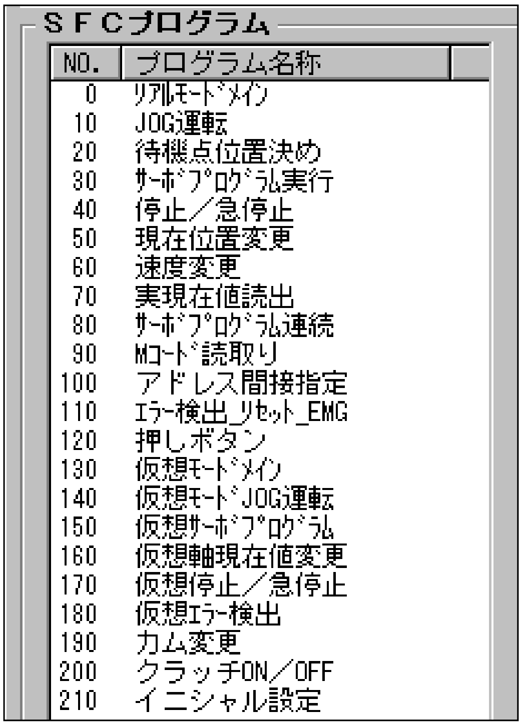
8) Click on the New creation button.



9) The NEW CREATION dialog box will open, so input the '130' for the SFC program No. and 'Virtual mode main' for the "SFC program name" before starting. After input, click on the OK button.

(Continued on next page)

(Continued from previous page)



(The specific procedures for creating the SFC program are not described in this section. Refer to the section "SFC program for operation" and create the program later.)

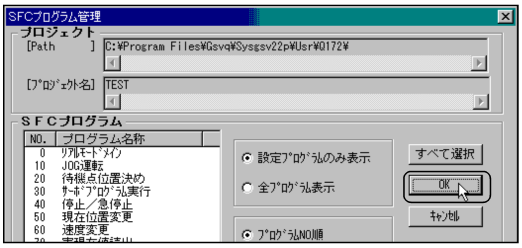
4) The set SFC programs will be listed.
Click on the New creation button again to create the SFC programs as shown below.

No.	Program name
130	Virtual mode main
140	Virtual mode JOG operation
150	Virtual servo program
160	Virtual axis current value change
170	Virtual stop/sudden stop
180	Virtual error detection
190	Cam change
200	Clutch ON/OFF

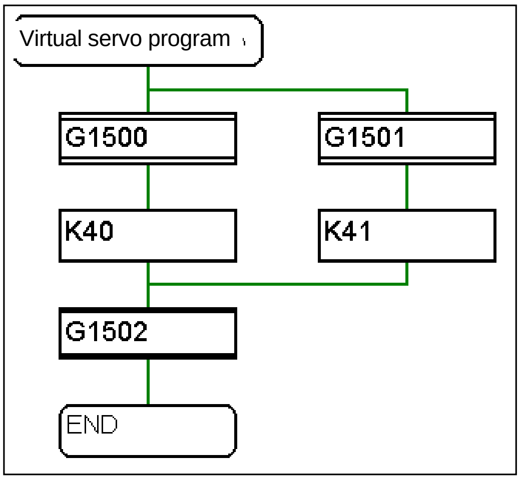
11.4.2 Inputting the motion control steps for the virtual mode

Set the motion control steps for the virtual mode.

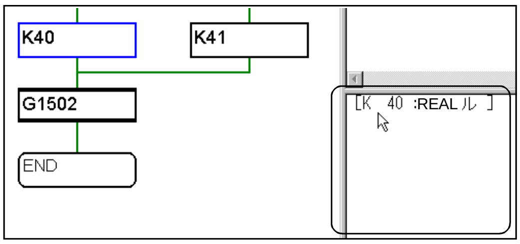
1) Create the SFC program for the virtual servo program.
Click on the SFC program control tool button on the PROGRAM EDIT screen.



2) Select "150 virtual servo program" from SFC program list in the SFC PROGRAM CONTROL dialog box, and click on the OK button.



3) Create an SFC program to change the current value as shown on the left.



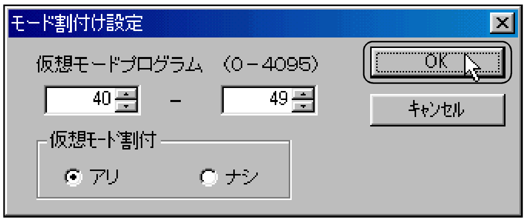
4) Click on the motion control step 'K40', and double-click on the PROGRAM EDIT screen.

(Continued on next page)

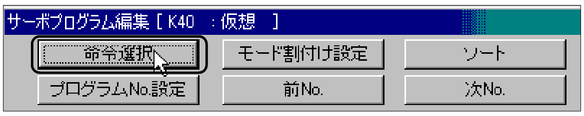
(Continued from previous page)

5) Click on the Cancel button in the COMMAND SELECTION dialog box.

6) Click on the Mode assignment setting button in the SERVO PROGRAM EDIT dialog box.



7) Set the "Virtual mode program" between '40' and '49', and the "Virtual mode assignment" to 'Yes' in the MODE ASSIGNMENT SETTING dialog box, and click on the OK button.



8) Click on the Command selection button in the SERVO PROGRAM EDIT dialog box.

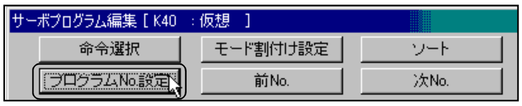
9) Set the "Command division" to 'Positioning' and the "Servo command" to 'ABS-1' in the COMMAND SELECTION dialog box, and click on the OK button.

(Continued on next page)

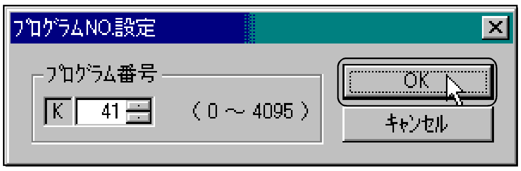
(Continued from previous page)

10) Input '1' and '0' in the "Axis: " text box, and '640000' in the "Speed" text box.

11) Click on the Store button.



12) Click on the Program No. setting button in the SERVO PROGRAM EDIT dialog box.



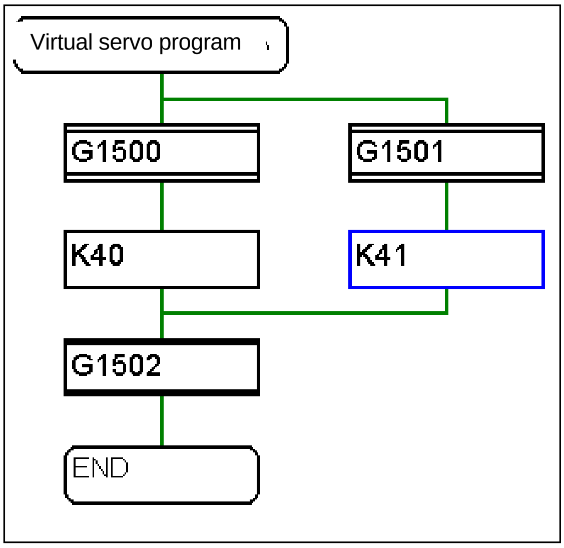
13) Set '41' for the "Program No." in the PROGRAM NO. SETTING dialog box, and click on the OK button.

```
[K  41  : VIRTUAL    ]
  1 ABS-1
    AXIS          1,        5242880    PLS
    VELOCITY                 640000    PLS/sec
```

14) Edit the program No.41 as shown on the left, and click on the Store button.
After editing, close the PROGRAM EDIT dialog box.

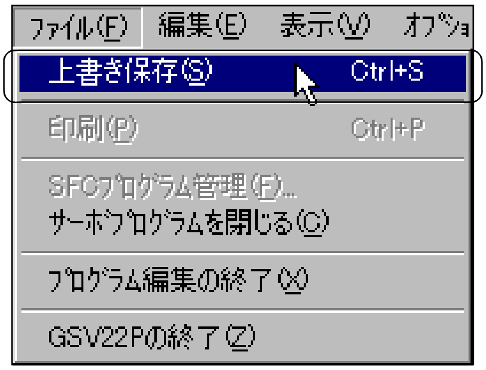
(Continued on next page)

(Continued from previous page)



15) Set the transition program shown below.

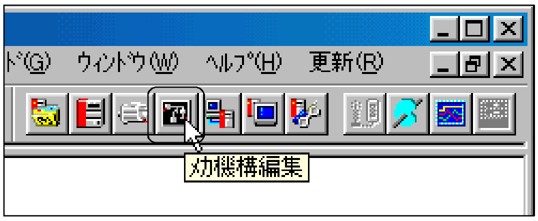
[G1500]	PX0*!M2001
[G1501]	PX1*!M2001
[G1502]	!(M2001*M2002)



16) To save the edited servo program, click on [File] and then the [Save] menu.
This completes editing of the servo programs No. 40 and No. 41 for the virtual mode.

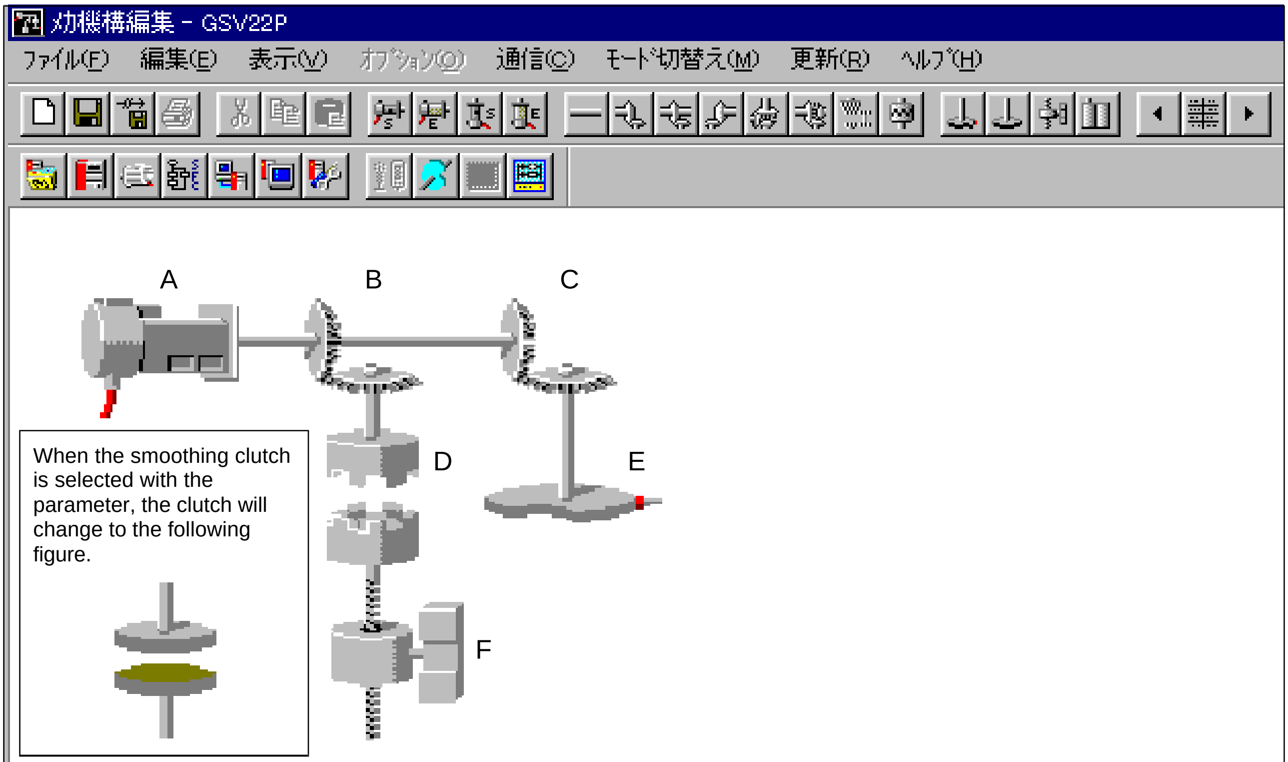
11.5 Editing the mechanism

The drive module, conveyance module and output modules for the virtual mode are set on the screen with the mouse.



1) Click on the mechanism editing tool button in the PROGRAM EDIT window.
After the MECHANISM EDIT window opens, close the PROGRAM EDIT window.

2) Edit the mechanism as shown below.



Note: The background of the actual screen is displayed in black.

[Module arrangement]

- Click the tool button of each module, and then click the EDIT screen. (Set A to F)
  The operation ends when the right-mouse button is clicked.

[Module deletion]

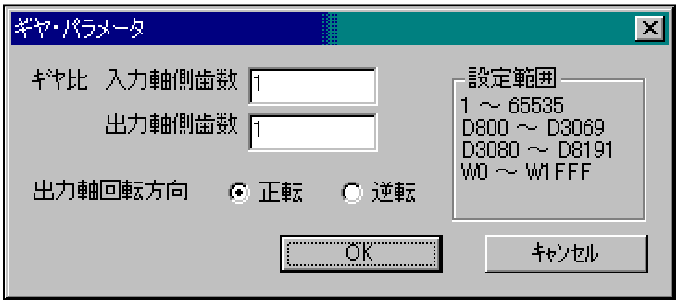
- End the module arrangement operation by clicking the right-mouse button, and drag the module to be deleted to a vacant space.
- Drag area around the module to be deleted.
  (In this case, drag an area slightly larger than the module to be deleted.)
- When the module is enclosed in a yellow frame, press the Delete key.

(Continued on next page)

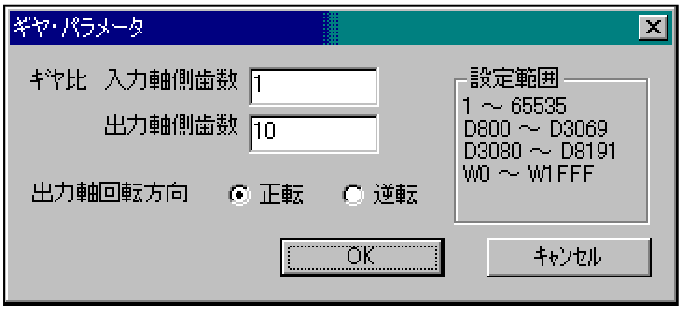
(Continued from previous page)

3) Double-click on the module A (virtual servomotor), and set the parameters as shown below.
After setting, click on the OK button.

4) Double-click on the module B (gear) and set the parameters as shown below.
After setting, click on the OK button.



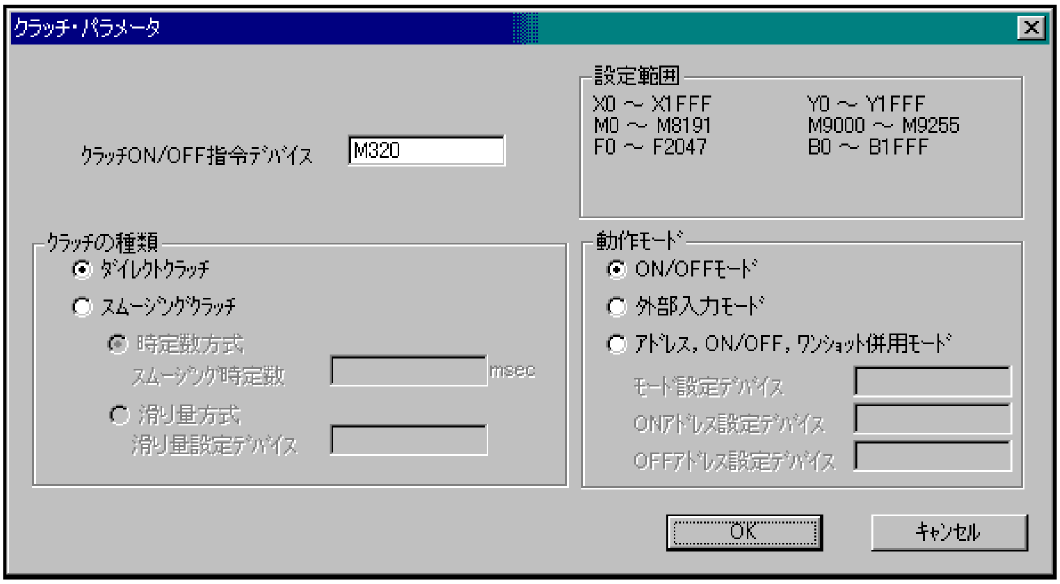
5) Double-click on the module C (gear) and set the parameters as shown below.
After setting, click on the OK button.



(Continued on next page)

(Continued from previous page)

6) Double-click on the module D (clutch) and set the parameter as shown below.
After setting, click on the OK button.

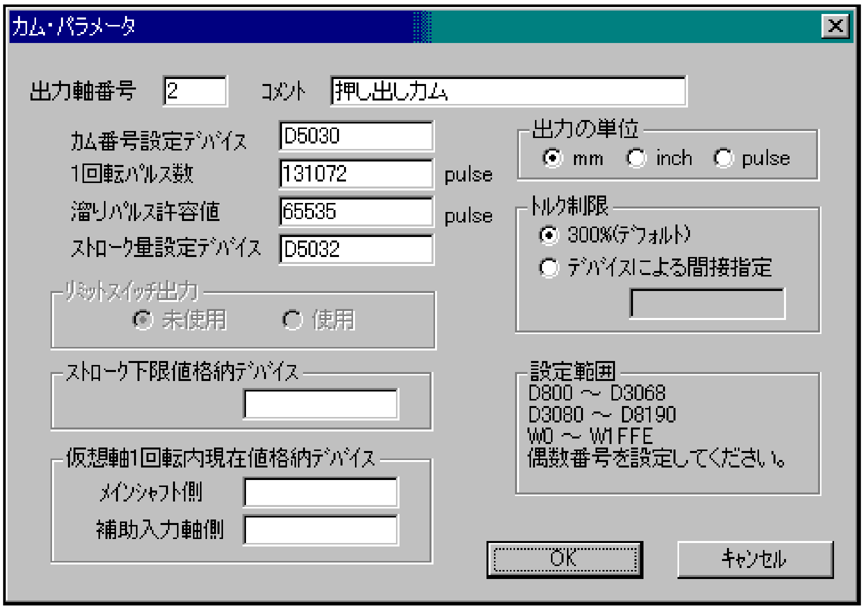


7) Double-click on the module E (ball screw) and set the parameters as shown below.
After setting, click on the OK button.

(Continued on next page)

(Continued from previous page)

8) Double-click on the module F (cam) and set the parameter as shown below.
After setting, click on the OK button.



9) To convert and save the mechanism, click on [File] and then the [Conversion/save] menu.

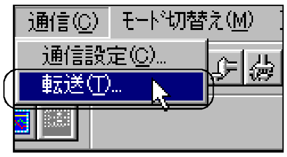
10) To designate the cam data, set the machine name 'Q172' set when the cam data was created, and then click on the OK button.

11) When the message "Completed normally." appears, click on the OK button.

11.6 Writing to the motion CPU

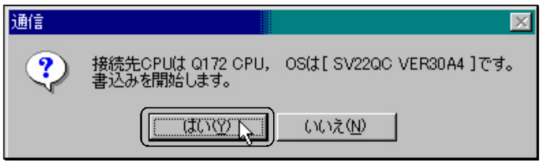
Write the following data to the motion CPU:

- Servo programs
- Mechanism programs
- Cam data



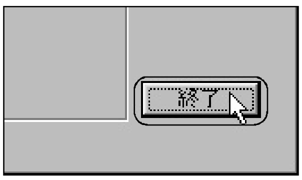
1) Stop the Q motion CPU.
2) Click on [Communication] and then the [Transfer] menu in the MECHANISM EDIT window.

3) Click and select the 'Servo program', 'Mechanism program' and then the 'Cam data' under "Transfer data" in the COMMUNICATION dialog box.
After selecting the data to transfer, click on the Write button.
(The servo setting data (system setting/servo setting data) does not need to be written as the same settings as the real mode set in chapter 6 are used)



4) A dialog box to confirm the connection destination CPU and main unit OS will open, so click on the YES button.

5) When the message "Completed normally." appears, click on the OK button.



6) Click on the END button in the COMMUNICATION dialog box to close the dialog box.

7) Reset the Q-PLC CPU to start the Q motion CPU.
This completes writing of the data to the Q motion CPU.

11.7 Reading of sequence program from Q-PLC CPU

(When the sequence program has been read from FD during "Practice with real mode" in Chapter 9, it is not necessary to execute this operation.)
In this practice, do not create the sequence program, but read it from the Q-PLC CPU, and monitor the circuit during practice operation.

1) Click on [Start], [Program], [MELSOFT application] and then the [GX Developer], to start the GX Developer.

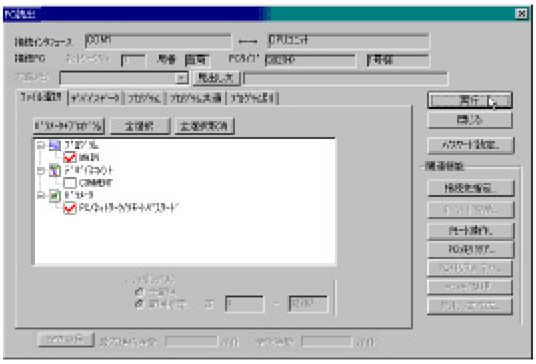
2) Click on [Project] and then the [Project new creation] menu.

3) The PROJECT NEW CREATION dialog box will open, so set the "PC series" to 'QCPU (Q mode)', the "PC type" to 'Q02(H)' and the "Program type" to 'Ladder'. Then, click on the OK button.

4) When the project is newly created, click on [Online] and then the [PC read] menu.

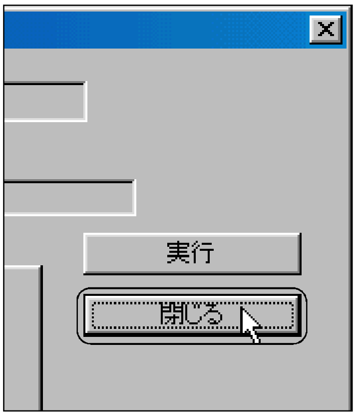
(Continued on next page)

(Continued from previous page)



5) The PC READ dialog box will open, so click on the Parameter + Program button to select the data to be read.
After selecting, click on the Execute button.

6) When the message "Completed." appears, click on the OK button.

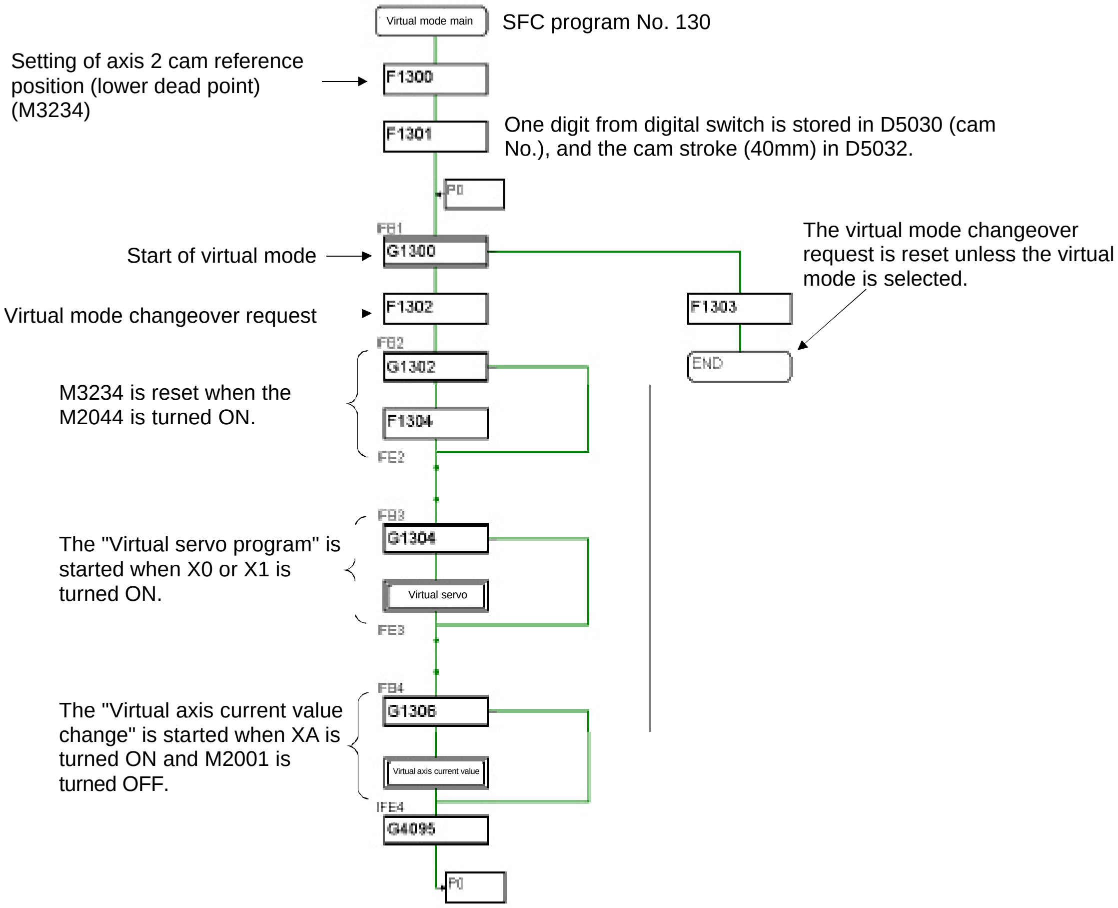


7) Click on the Close button to close the PC READ dialog box.
This completes reading of the sequence program from the Q-PLC CPU.

Click on [On-line], [Monitor] and then the [Monitor mode] menu to execute the circuit monitor.

11.8 SFC program for practice

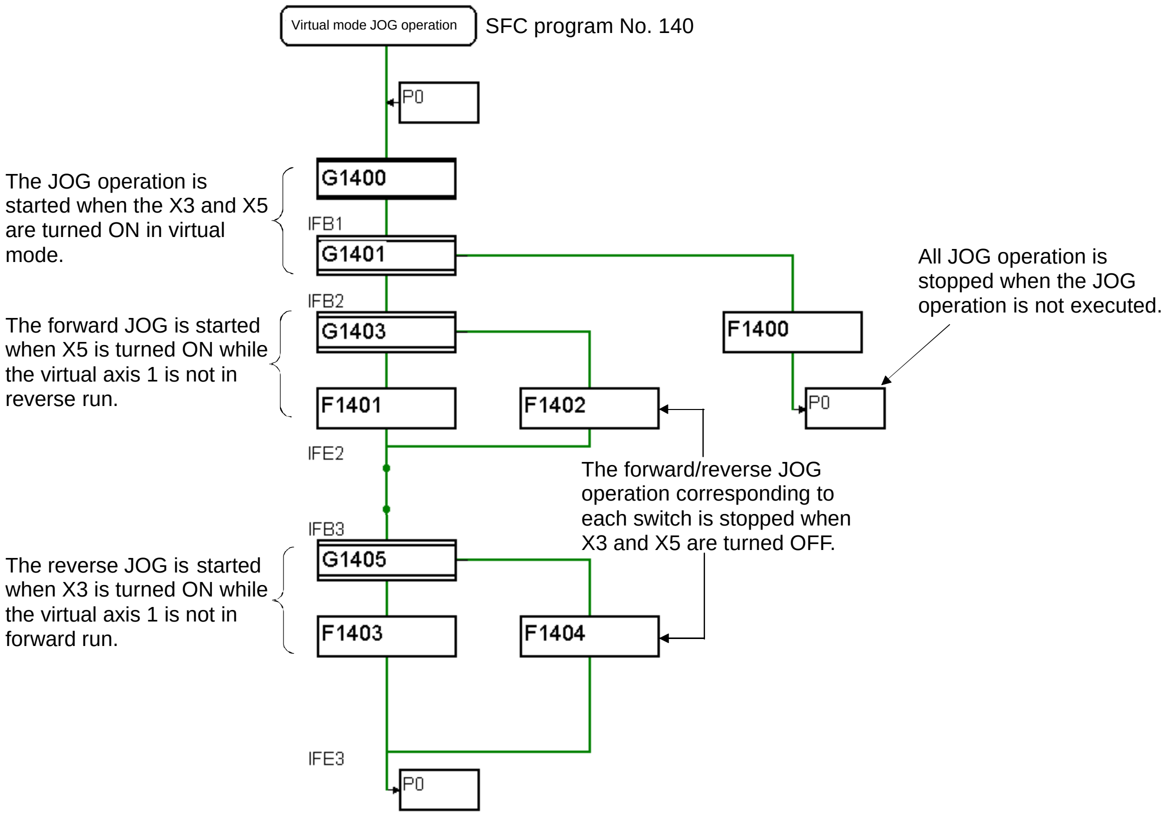
[Virtual mode main] program No. 130



[Transition]	G1300 PX1A	G1302 M2044
	G1304 PX0+PX1	G1306 PXA*!M2001
	G4095 NOP	
[Operation control step]	F1300 SET M3234	F1301 DIN D5028,PX10 D5028 = D5028&HF D5030 = BIN(D5028) D5032L = K400000
	F1302 SET M2043	F1303 RST M2043 RST M3234
	F1304 RST M3234	

M2043	: Real/virtual mode select request
M2004	: Real/virtual mode select status
M3234	: Cam reference position setting
X0	: Servo program No.40 start command
X1	: Servo program No.41 start command
XA	: Virtual mode select
D5030	: Cam No.
D5032, D5033	: Can stroke rate

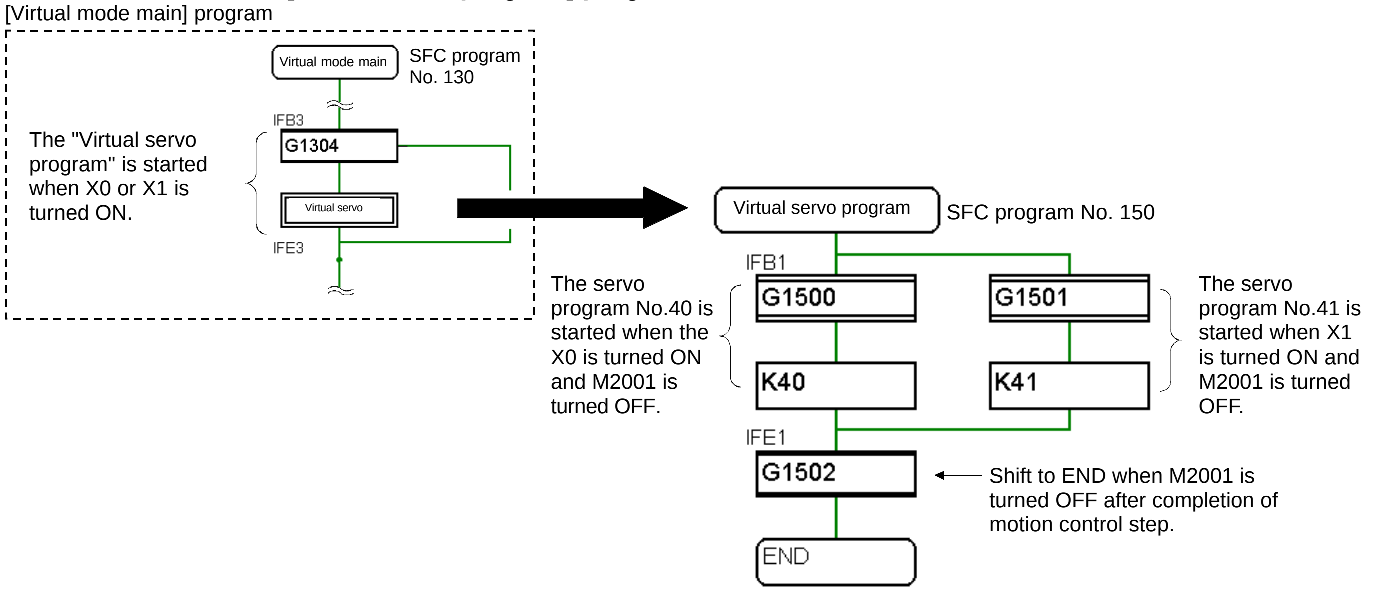
[Virtual mode JOG operation] program No.140



[Transition]	G1400 M2044	G1401 M2044
	G1403 PX5*!M4803	G1405 PX3*!M4802
[Operation control step]	F1400 RST M4802 RST M4803	F1401 D640L = K320000 SET = M4802
	F1402 RST M4802	F1403 D640L = K320000 SET = M4803
	F1404 RST M4803	

M4802 : Virtual axis 1 forward JOG start
M4083 : Virtual axis 1 reverse JOG start
X5 : Virtual axis 1 forward JOG command
X3 : Virtual axis 1 reverse JOG command
D640, D641 : Axis 1 JOG speed setting register

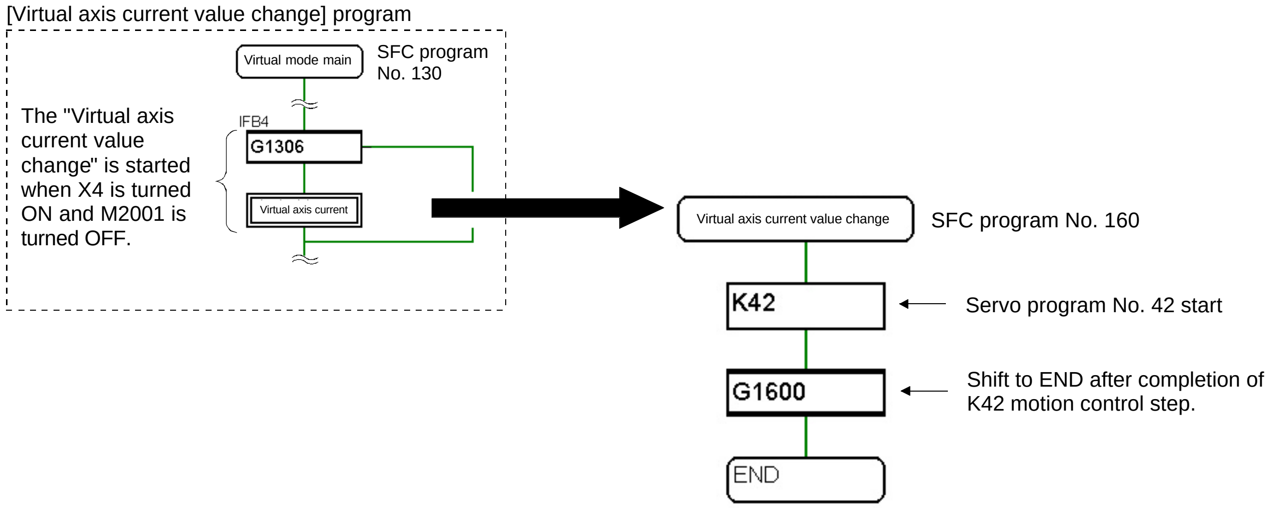
[Virtual servo program] program No. 150



[Transition]	G1500 PX0*!M2001 G1502 !M2001	G1501 PX1*!M2001
[Motion control step]	K40 VIRTUAL 1 ABS-1 AXIS 1, 0 PLS VELOCITY 640000 PLS/sec	K41 VIRTUAL 1 ABS-1 AXIS 1, 5242880 PLS VELOCITY 640000 PLS/sec

M2001 : Axis 1 start accept flag
X0 : Servo program No. 40 start command
X1 : Servo program No. 41 start command

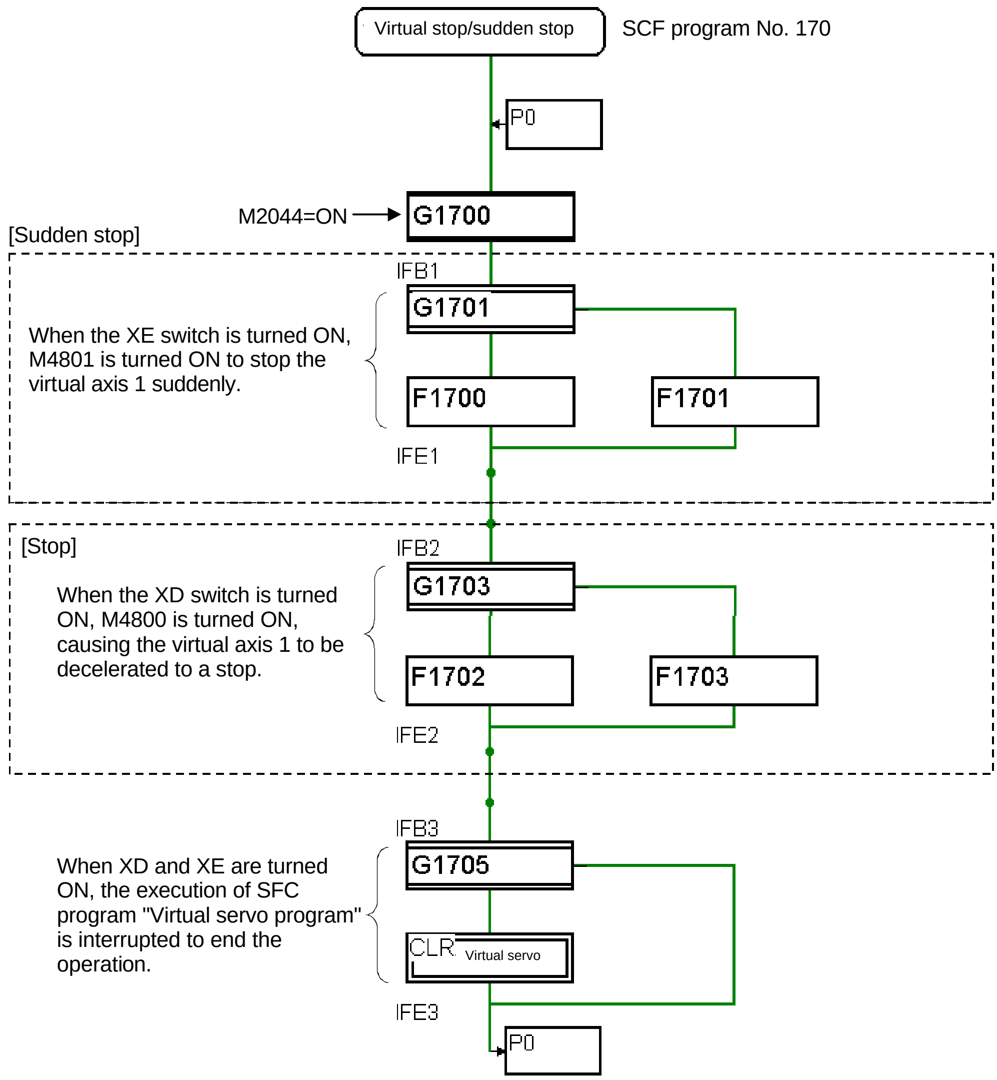
[Virtual axis current value change] program No. 160



[Transition]	G1600 NOP
[Motion control step]	K42 VIRTUAL 1 CHGA AXIS 1, 0 PLS

M2001 : Axis 1 start accept flag

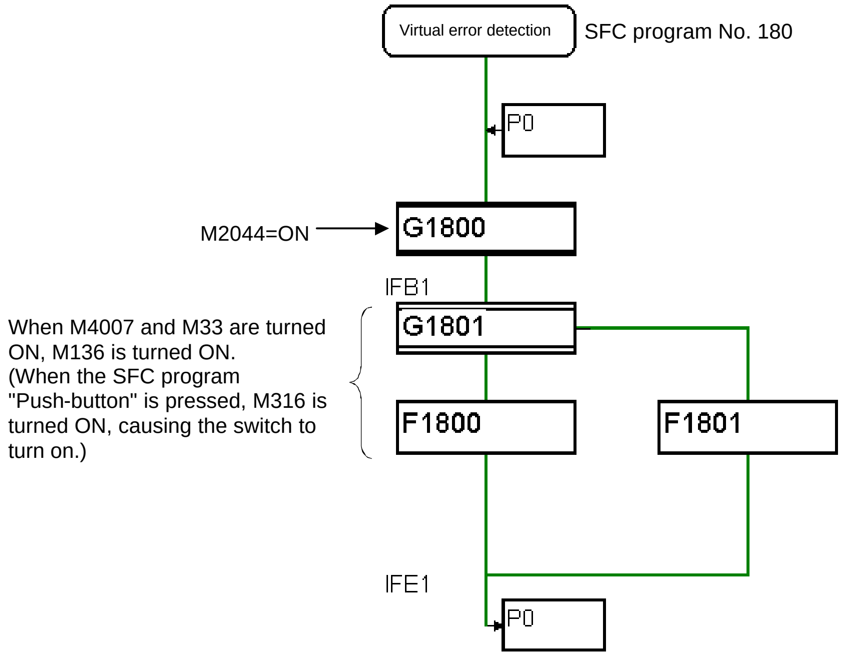
[Virtual stop/sudden stop] program No. 170



[Transition]	G1700 M2044	G1701 PXE
	G1703 PXD*!M4801	G1705 PXD+PXE
[Operation control step]	F1700 SET M4801	F1701 RST M4801
	F1702 SET M4800	F1703 RST M4800

M4808 : Deceleration stop signal
M4801 : Sudden stop signal
XD : Deceleration stop command
XE : Sudden stop command

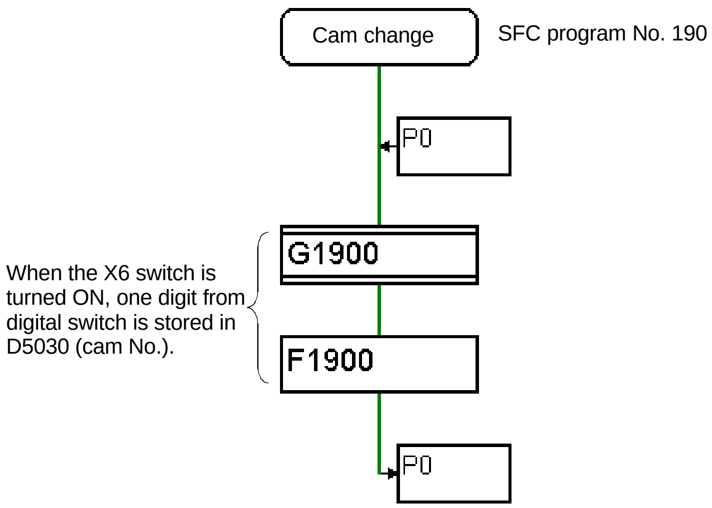
[Virtual error detection] program No. 180



[Transition]	G1800 M2044	G1801 M4007*M33
[Operation control step]	F1800 SET M316	F1401 RST M316

M33 : 2-second clock
M316 : Error detection signal
M4007 : Virtual servo error detection

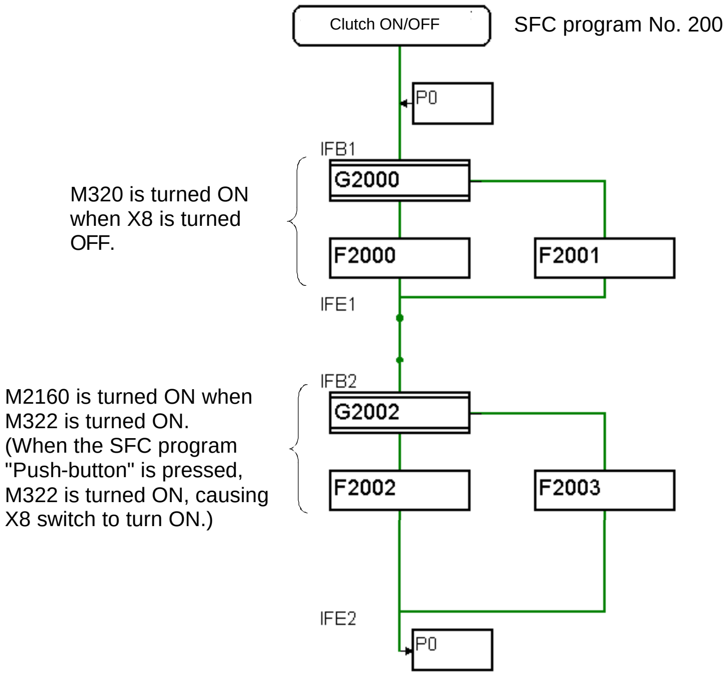
[Cam change] program No. 190



[Transition]	G1900 PX6
[Operation control step]	F1900 DIN D5028,PX10 D5028 = D5028 & HF D5030 = BIN(D5028)

M6 : Cam No. change command
D5030 : Cam No.

[Clutch ON/OFF] program No. 200



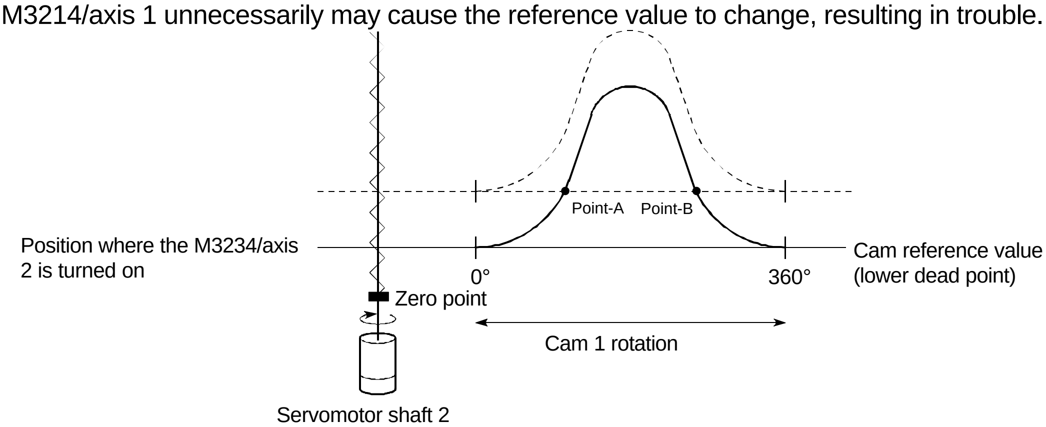
[Transition]	G2000 !PX8	G2002 M2160
[Operation control step]	F2000 SET M320	F2001 RST M320
	F2002 SET M322	F2003 RST M322

M320 : Clutch ON/OFF command device
M322 : Clutch ON signal
M2106 : 1-axis clutch status (main shaft)

Precautions for creation of virtual mode program

To use the cam, change the mode to the virtual mode and turn ON M2043 while the cam reference value (lower dead point) setting M3214 and axis 1 are on.

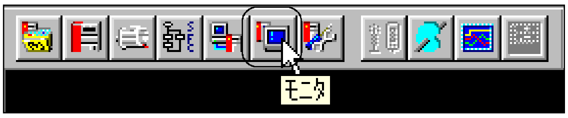
If the reference value is determined, M3214/axis 1 do not need to be turned ON/OFF each time. Note that turning M3214/axis 1 unnecessarily may cause the reference value to change, resulting in trouble.



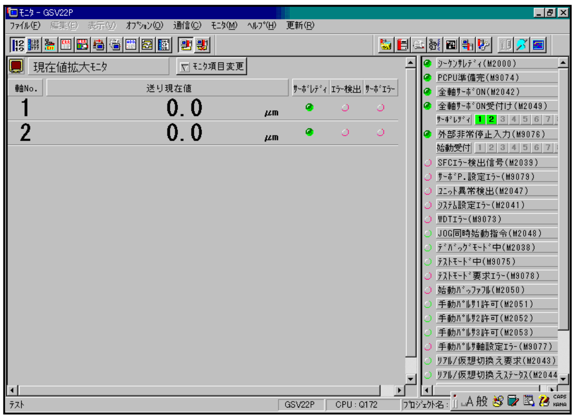
If the cam is stopped at the point A or point B during the operation shown with the solid line, turn ON/OFF M3214/axis 1. The reference value (lower dead point) will change to the position where the cam is stopped, and the operation will follow the dotted line when started next.

11.9 Practice machine operations

Monitor the operation with the X-Y table movement and a personal computer.



1) Click on the monitor tool button in the MECHANISM EDIT window.



2) The CURRENT VALUE ENLARGED MONITOR will open in the MONITOR window.

Start the Q-PLC CPU and Q motion CPU.

[Execution of zero point return]

- Set the mode selector switch to [REAL] X19.
- Set the digital switches X10 to X17 to | 0 | 0 |.
- When the | X1 | switch is pressed, axis 1 and axis 2 will return to the zero point.

The current feed value will change to -30000.0μm for both axis 1 and axis 2.

[Positioning to waiting point]

- When | X0 | is pressed, the axis will be positioned the waiting point (X-axis address: '0', Y-axis address: '0').

[Setting of cam No.1 to '1']

- Set the digital switches X10 to X17 to | 0 | 1 |.

(Continued on next page)

(Continued from previous page)

[Changing to virtual mode]

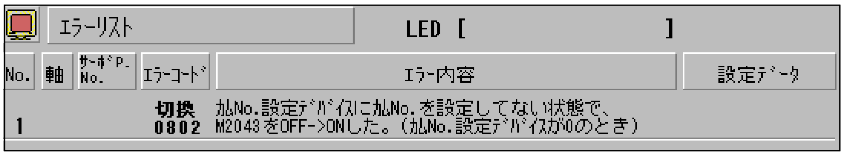
Set the mode selector switch from [REAL] (X19 ON) to [VIRTUAL] (X1A ON).

The virtual mode is entered if the X0A lamp is on.

If the X0A lamp does not turn on, and the X0C error lamp flickers, check the details of the error, and correct the settings.

Confirming the error

[Error list] → [Error list] menu

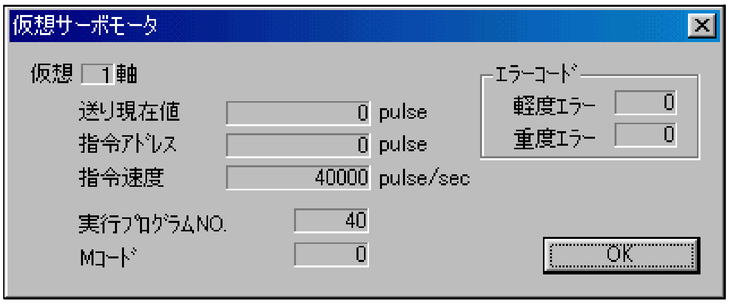


[Virtual drive servomotor axis 1 current value monitor]

Enlarge the current value monitor in the MONITOR window.

[Mechanism monitor]

- Change the window to the MECHANISM EDIT window (active window).
- Click on [Mode change] and then the [Monitor] menu in the MECHANISM EDIT window.
- Double-click on the virtual servomotor to display the DETAILS MONITOR.



If the current value change X0A for virtual axis 1 is pressed when the "Current feed value" is not '0', the value will be set to '0'.

Click on the OK button to close the window.

[Starting in the virtual mode (cam No. 1)]

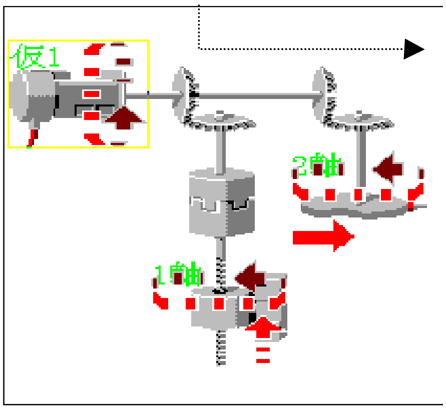
- When X1 is pressed, the cam curve will be drawn.
  ([FEED PRE. VAL.] will stop at 5242880 pulse.)
- The axis will return to the standby point when X0 is pressed.
  ([FEED PRE. VAL.] will stop at 0 pulse.)
- X0D stop and X0E sudden stop are valid during this operation.
- X05 forward jog and X03 reverse jog are valid when stopped.

(Continued on next page)

(Continued from previous page)

[Mechanism monitor]

Close the virtual servomotor's DETAILS MONITOR dialog box.
Double-click on the cam position to display the CAM DETAILS MONITOR dialog box.



| X1 | Move to the right
| X0 | Move to the left

• '1' is displayed for "Execution cam No.".

[Setting of cam No. to '2']

• Set the digital switches X10 to X17 to | 0 | 2 |.
• Press | X6 | cam data change request.
• '2' will appear at "Execution Cam No."

[Start in the virtual mode (cam No. 2)]

| X1 | Move to the right
| X0 | Move to the left

[Setting of cam No. to '3']

• Set the digital switches X10 to X17 to | 0 | 3 |.
• Press | X6 | cam data change request.
• '3' will appear at "Execution Cam No."

[Start in the virtual mode (cam No. 3)]

| X1 | Move to the right
| X0 | Move to the left

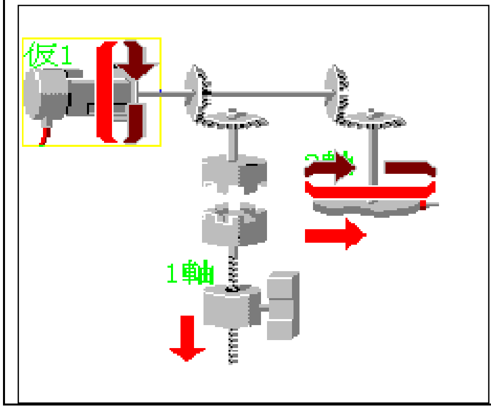
(Continued on next page)

(Continued from previous page)

[Clutch operation]

Press X8 clutch OFF while operating in the virtual mode.

The mechanism monitor clutch will open, and movement in the X axis direction will stop.



Caution

- If the clutch is turned OFF, the X axis movement will stop. When the clutch is turned ON, the X axis direction position will change by the amount that the axis did not move.

[Items to confirm]

- Is the cam changed to No. 1, No. 2 or No. 3?
- Observe the stop, sudden stop, forward run JOG and reverse run JOG states.
- Observe the details monitor for each module.
- Observe the state when the current value for the virtual axis 1 is changed to "0".
- Observe the movement when the clutch is turned OFF.

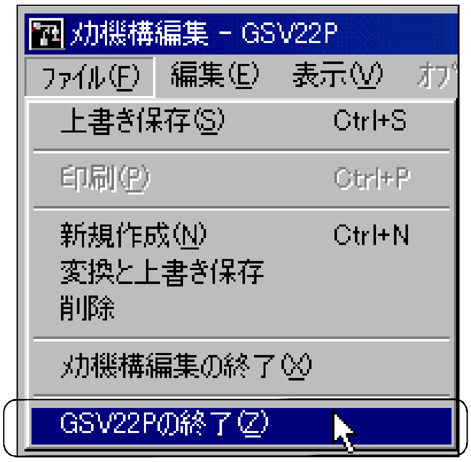
The practice session is done when the series of operations are completed.

REMARK

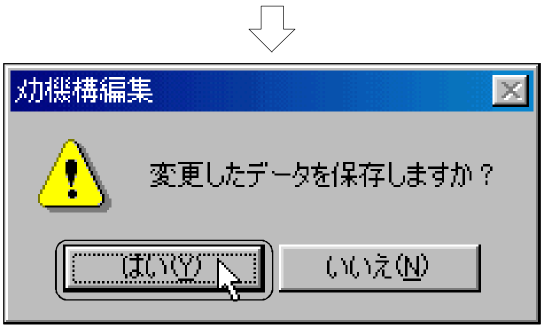
The window for each SW6RN-GSV22P function or GX Developer window can be changed with one-touch operations.
Click on the windows task bar icon for the window to be opened.

The window clicked on will open.

[END operation]



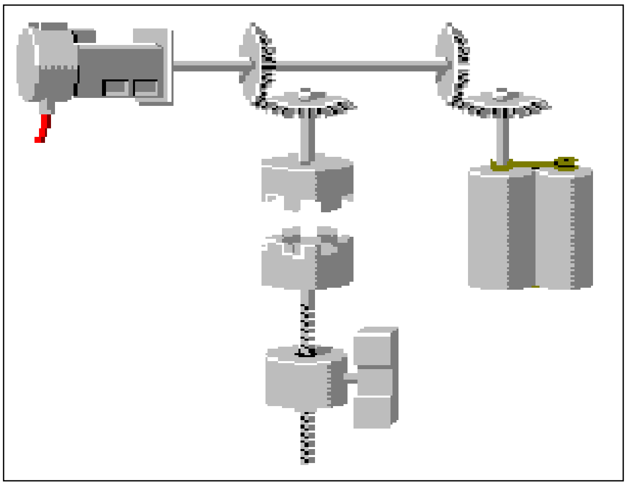
1) Click on [File] and then the [GSV22P END] menu in the MECHANISM EDIT window.



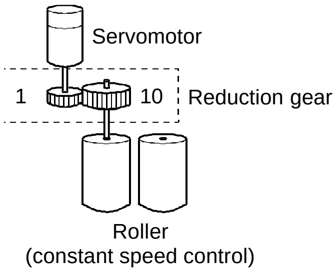
2) If the edited data is not saved, a dialog to confirm the overwriting of the data will appear. Click on the YES button.
This completes the operation.

11.10 Exercise (Roller setting)

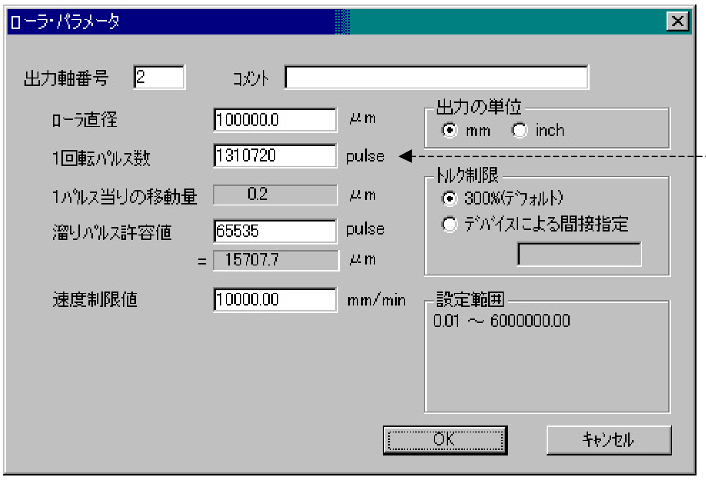
Change the cam to the following roller and move it.



Conditions: It is assumed that a reduction gear is installed externally to increase the torque.



ROLLER PARAMETERS



The feedback pulse is set to 10-fold, so the roller will actually rotate once with 10 servomotor rotations.

Click on the [OK] button in the above setting.

Select [File], [Conversion/Save as] menu to save the setting contents.

Select [Communication] and then the [Transfer] menu to write the 'Mechanism program' into the motion CPU.

Select [Mode change] and then the [Monitor] menu to change the MECHANISM EDIT window to 'Monitor mode'.

Start in the virtual mode (Roller)

X0B	Move pen down
X1	Move to right
X0	Move to left

Appendix

Appendix 1 Examples of programs for SV22 virtual mode

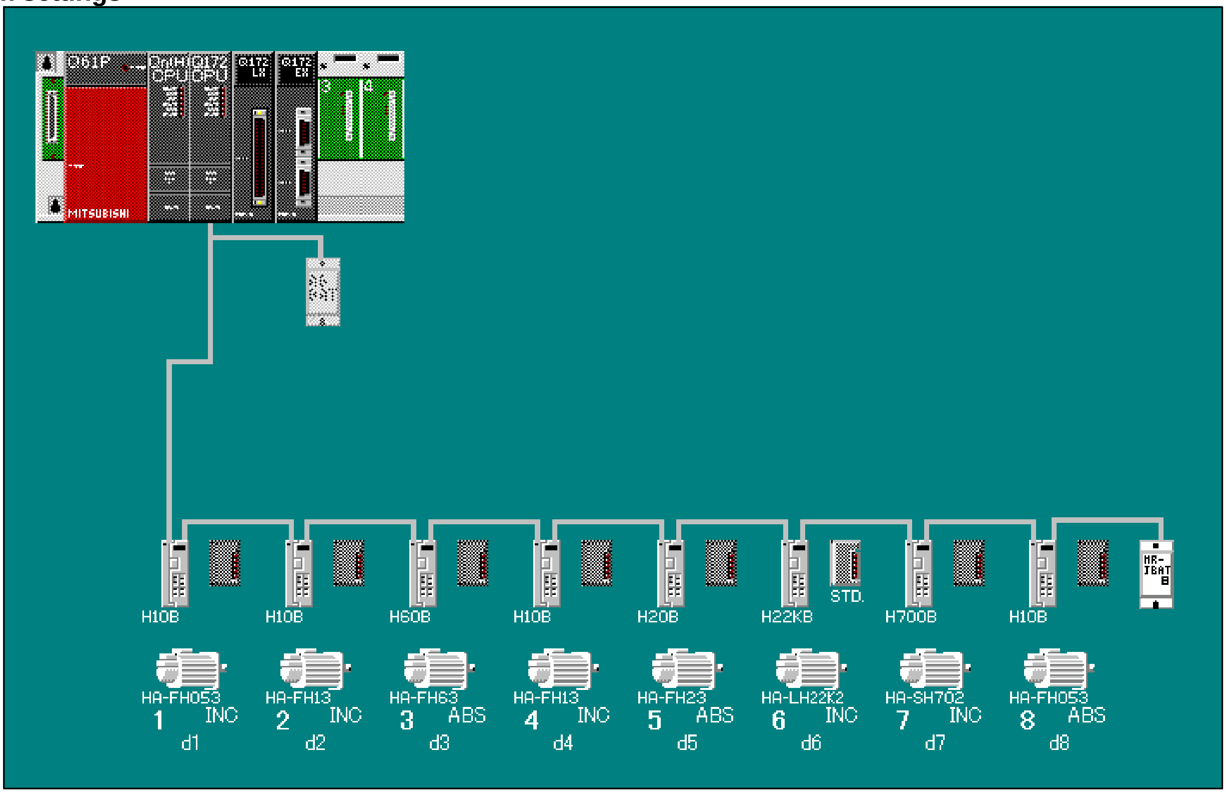
Program example 1.................................. A-2
Program example 2.................................. A-4
Program example 3.................................. A-6
Program example 4.................................. A-8

Program example 1

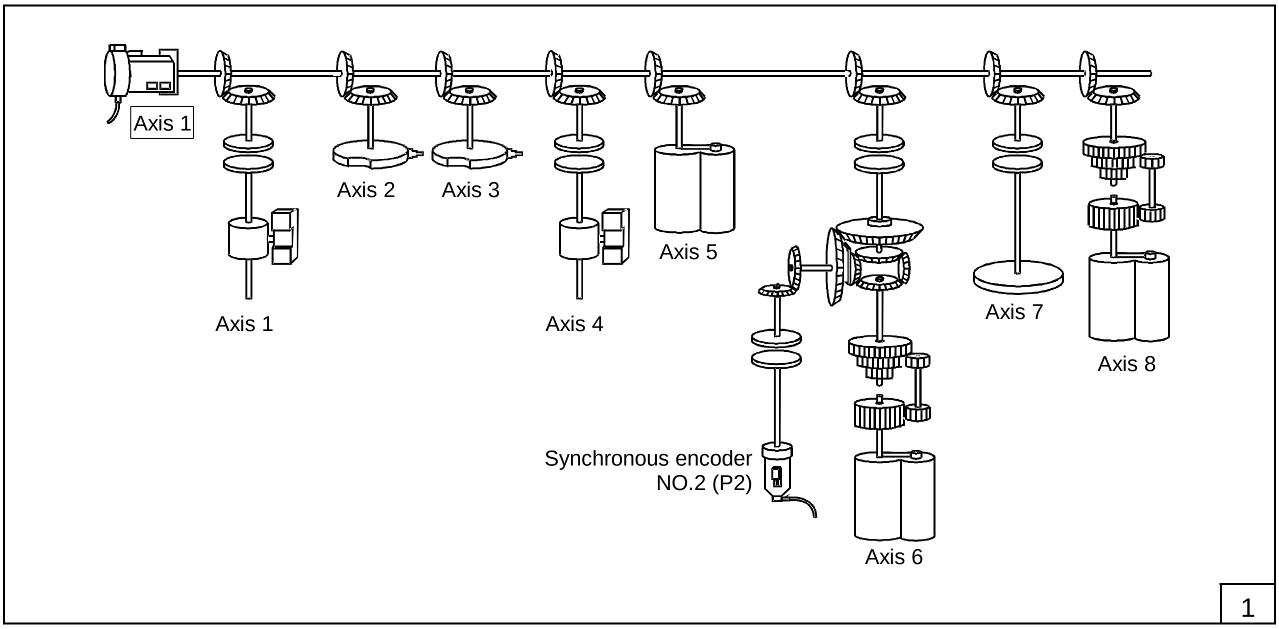
(1) Synchronously operate axis 1, axis 2, axis 3, axis 4, axis 5, axis 6, axis 7 and axis 8 with virtual servomotor $\boxed{\text{Axis 1}}$.

(2) Wire synchronous encoder to P2, and make auxiliary input to axis 6.

System settings



Mechanism connection diagram



Servo program

```
[K  44 : VIRTUAL ]
  1  VF
       AXIS          1
       SPEED                 50000   PLS/sec
```

Speed control forward run

The drive module's virtual servomotor axis 1 will move in the forward run direction.

```
[K  45 : VIRTUAL ]
  1  VR
       AXIS          1
       SPEED                 50000   PLS/sec
```

Speed control reverse run

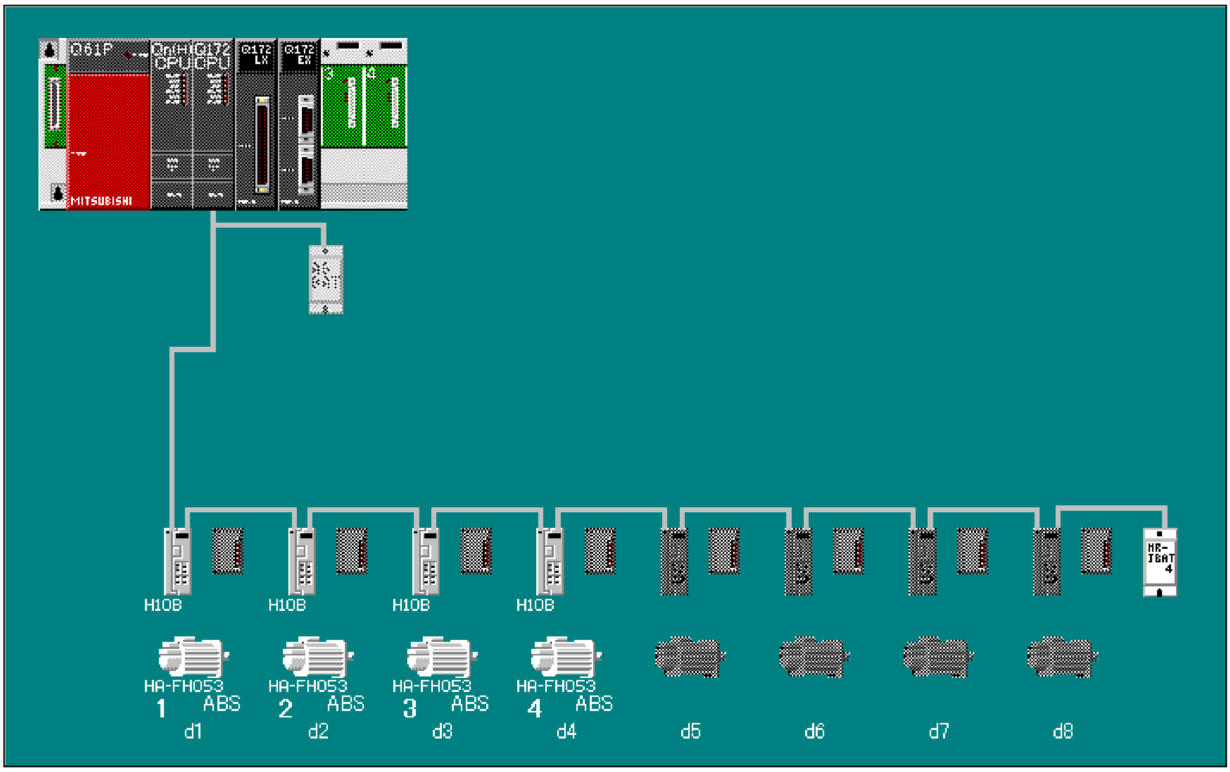
The drive module's virtual servomotor axis 1 will move in the reverse run direction.

Program example 2

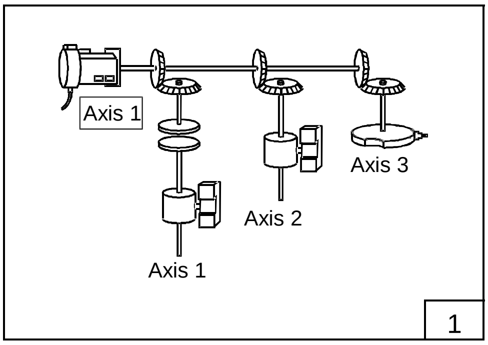
Control details

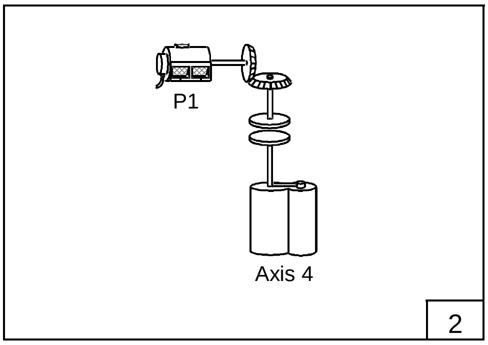
(1) Synchronously operate axis 1, axis 2, and axis 3 with virtual servomotor $\boxed{\text{Axis 1}}$.
(2) Wire synchronous encoder to P1, and make auxiliary input to axis 4.

System settings



Mechanism connection diagram





Servo program

```
[K   44 : VIRTUAL ]
  1   VF
        AXIS          1
        SPEED                    50000   PLS/sec
```

Speed control forward run
The drive module's virtual servomotor axis 1 will move in the forward run direction.

```
[K   45 : VIRTUAL ]
  1   VR
        AXIS          1
        SPEED                    50000   PLS/sec
```

Speed control reverse run
The drive module's virtual servomotor axis 1 will move in the reverse run direction.

Program example 3

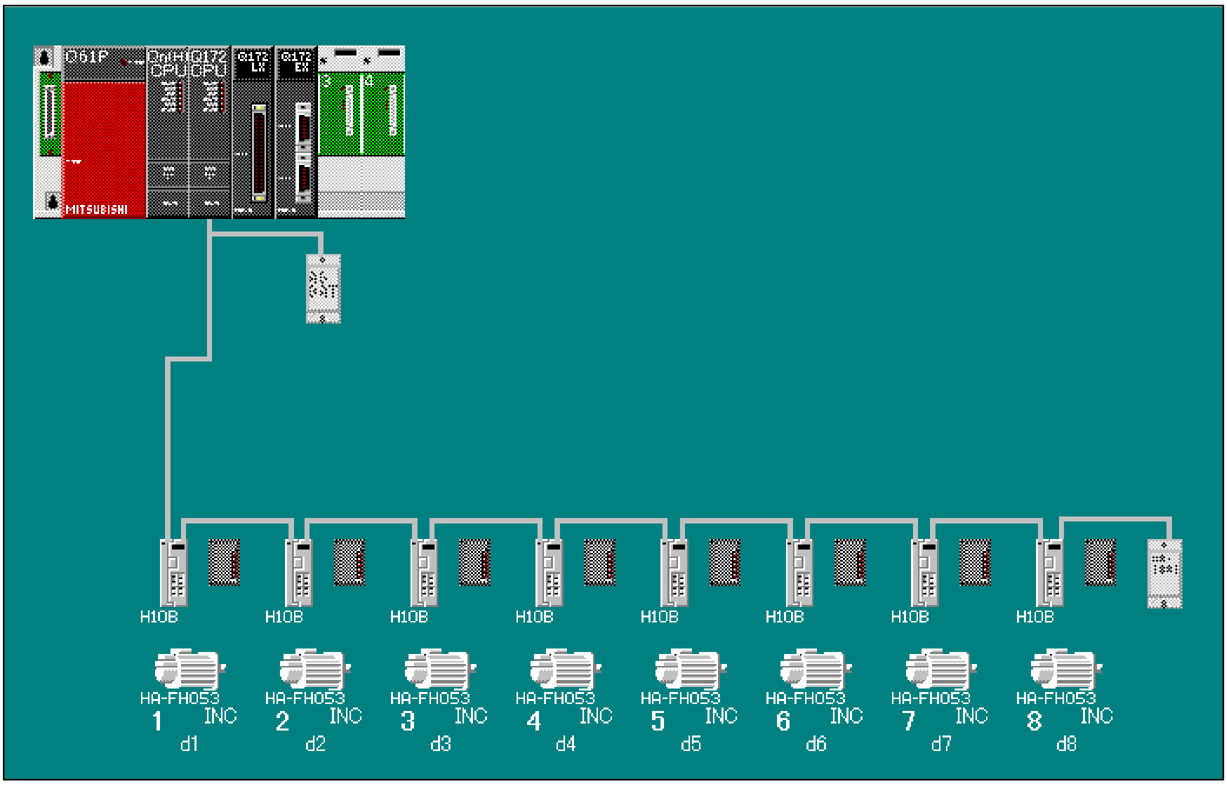
Control details

(1) Synchronously operate axis 1, axis 2, and axis 3 with virtual servomotor $\boxed{\text{Axis 1}}$.

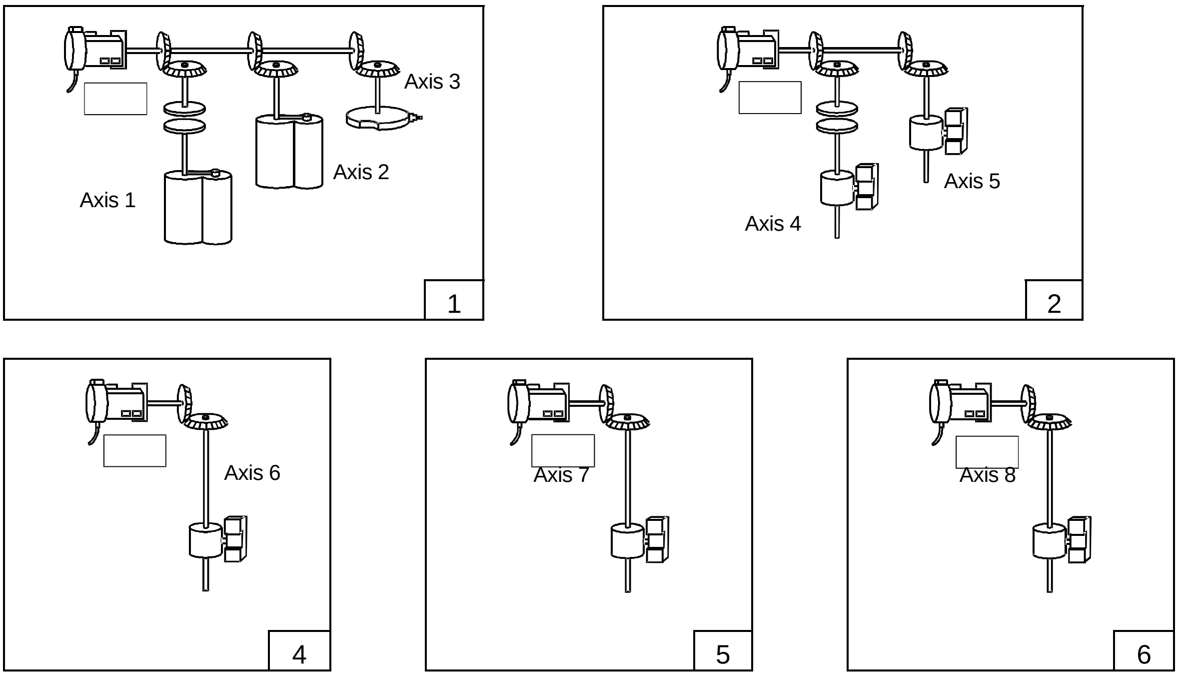
(2) Synchronously operate axis 4 and axis 5 with virtual servomotor $\boxed{\text{Axis 2}}$.

(3) Carry out 3-axis linear interpolation or 2-axis circular interpolation of axis 6, axis 7 and axis 8 with virtual servomotors $\boxed{\text{Axis 3}}$, $\boxed{\text{Axis 4}}$ and $\boxed{\text{Axis 5}}$.

System settings



Mechanism connection diagram



Servo program

```
[K  44 : VIRTUAL ]
  1   VF
        AXIS          1
        SPEED                     50000    PLS/sec
```

Speed control forward run

The drive module's virtual servomotor axis 1 will move in the forward run direction.

```
[K  45 : VIRTUAL ]
  1   VR
        AXIS          1
        SPEED                     50000    PLS/sec
```

Speed control reverse run

The drive module's virtual servomotor axis 1 will move in the reverse run direction.

```
[K  46 : VIRTUAL ]
  1   ABS-1
        AXIS          2,              0    PLS
        SPEED                     40000    PLS/sec
```

Absolute 1-axis linear control

The drive module's virtual servomotor axis 2 rotates to address "0".

```
[K  47 : VIRTUAL ]
  1   ABS-1
        AXIS          2,         655360    PLS
        SPEED                     40000    PLS/sec
```

Absolute 1-axis linear control

The drive module's virtual servomotor axis 2 rotates to address "655360".
(Rotates 40 times at 16384 pulse/rotation.)

```
[K  42 : VIRTUAL ]
  1   ABS-3
        AXIS          3,         655360    PLS
        AXIS          4,        1310720    PLS
        AXIS          5,        1966080    PLS
        SYN. SPEE                 50000    PLS/sec
```

Absolute 3-axis linear interpolation control

The drive module's virtual servomotor axis 3, axis 4 and axis 5 rotate to the designated address.

Program example 4

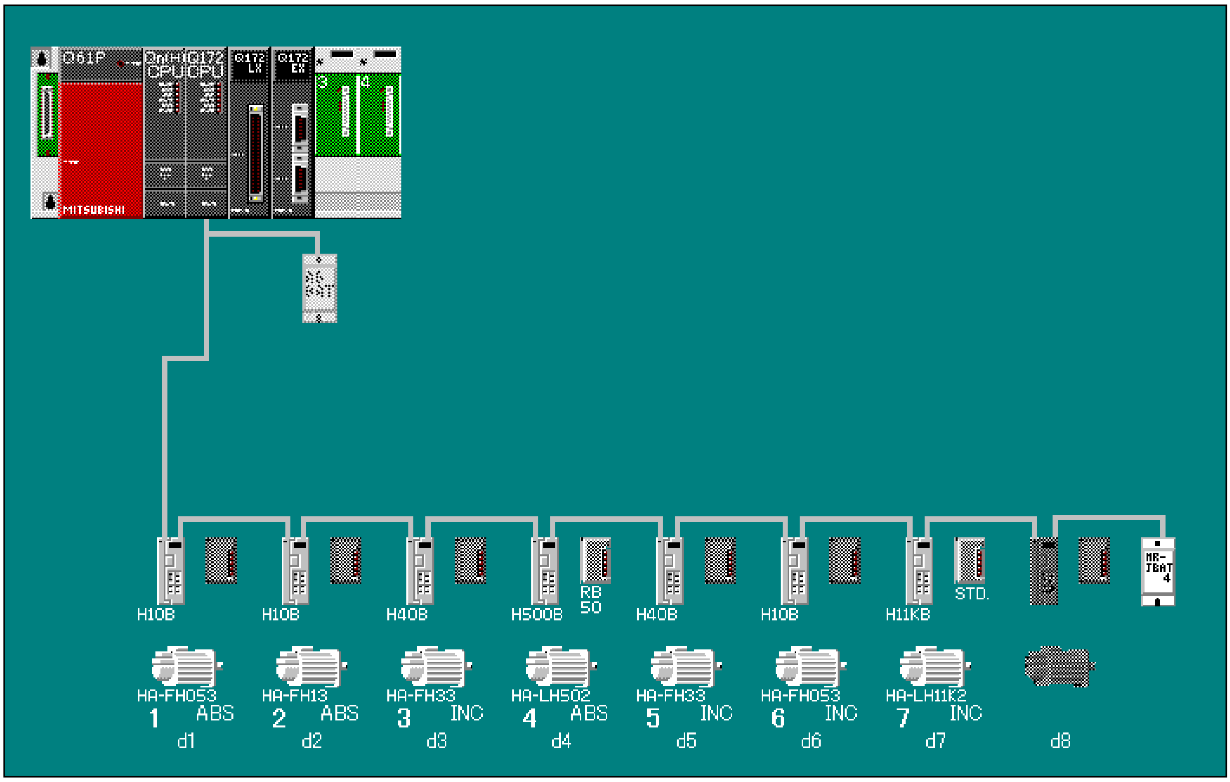
Control details

(1) Synchronously operate axis 1 and axis 2 with virtual servomotor [Axis1].

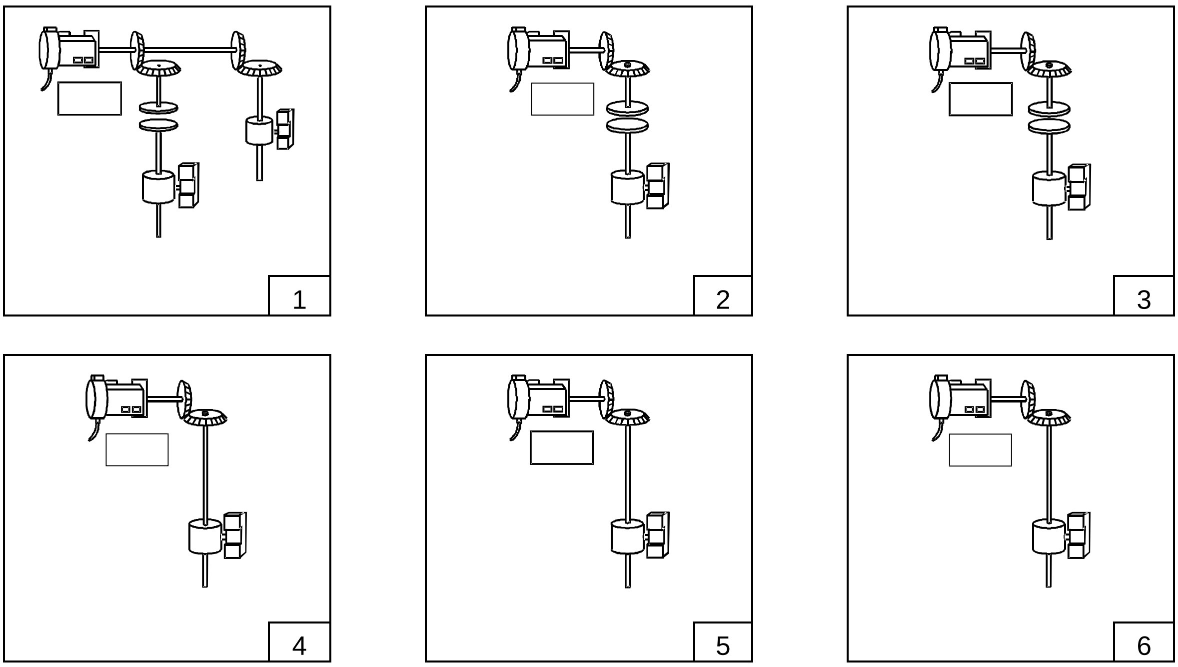
(2) Carry out 3-axis linear interpolation of axis 3, axis 4 and axis 5 with virtual servomotors [Axis 3], [Axis 4] and [Axis 5].

(3) Carry out 2-axis circular interpolation of axis 6 and axis 7 with virtual servomotors [Axis 6] and [Axis 7].

System settings



Mechanism connection diagram



Servo program

```
[K  44 : VIRTUAL ]
  1   VF
        AXIS          1
        SPEED                    50000   PLS/sec
```

Speed control forward run

The drive module's virtual servomotor axis 1 will move in the forward run direction.

```
[K  45 : VIRTUAL ]
  1   VR
        AXIS          1
        SPEED                    50000   PLS/sec
```

Speed control reverse run

The drive module's virtual servomotor axis 1 will move in the reverse run direction.

```
[K  42 : VIRTUAL ]
  1   ABS-3
        AXIS          3,        655360   PLS
        AXIS          4,       1310720   PLS
        AXIS          5,       1966080   PLS
        SYN. SPEE                50000   PLS/sec
```

Absolute 3-axis linear interpolation control

The drive module's virtual servomotor axis 3, axis 4 and axis 5 rotate to the designated address.

```
[K  43 : VIRTUAL ]
  1   ABS⌒
        AXIS          6,        655360   PLS
        AXIS          7,        655360   PLS
        SPEED                    50000   PLS/sec
        PAS-PT.       6,        163840   PLS
        PAS-PT.       7,        491520   PLS
```

Absolute auxiliary point designation circular interpolation control

The drive module's virtual servomotor Axis 6 and Axis 7 rotate to the end point address while passing through the designated auxiliary points.

Appendix 2 Sample motion SFC

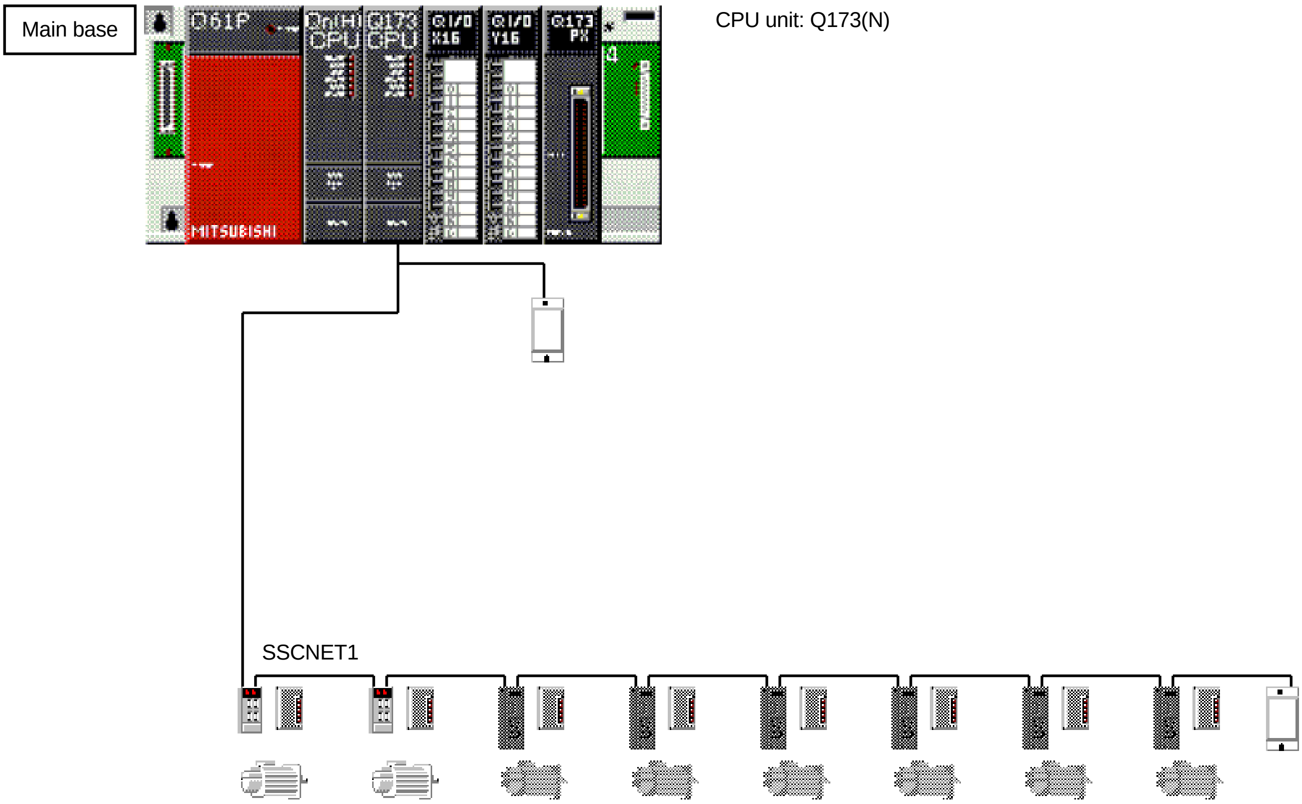
The sample program stops all motion control upon reception of emergency stop input, and re-starts motion control when reset. This sample program also monitors the dedicated positioning devices on the PLC side.

(1) Sample program functions

This sample program is provided with the following functions.

No.	Item	Contents
1	Monitor of dedicated positioning device	Reflects the status of Q173CPU(N) and dedicated positioning device (#2 machine) in the M2400 and following, and D0 and following of QnHCPU (#1 machine).
2	Clock data reading	Turns ON the clock data read request (M9028) so that the clock data is set in the error history.
3	Emergency stop	Turns ON the servo for all axes when the emergency stop input assigned to PX0 is turned ON (emergency stop reset), and executes motion control. Stops the servo amplifier in an emergency state when the emergency stop input is turned OFF and stops the motion control. The actual output (PY) is also turned OFF.
4	Motion control	Executes the motion control in each mode shown below according to the status of PX1, PX2. • PX2:OFF PX1:OFF "JOG mode" • PX2:OFF PX1:ON "Manual pulse generator mode" • PX2:ON PX1:OFF Zero point return mode" • PX2:ON PX1:ON Program operation mode"
5	JOG mode	Executes the following JOG operations when the signals PX3 to PX6 are turned ON. • PX3: 1-axis forward JOG • PX4: 1-axis reverse JOG • PX5: 2-axis forward JOG • PX6: 2-axis reverse JOG
6	Manual pulse generator mode	Executes the following manual pulse generator operation. • Executes the manual pulse generator operation of 1-axis with manual pulse generator P1. • Executes the manual pulse generator operation of 2-axis with manual pulse generator P2.
7	Zero point return mode	Executes the following zero point return. • When the PX3 is turned ON, the 1-axis is returned to the zero point. • When the PX4 is turned ON, the 2-axis is returned to the zero point.
8	Program operation mode	Executes the following program operations. • Waits for 1,000ms when the PX3 is detected to change from OFF to ON after the 1-axis is positioned to position the 2-axis. • Executes the in-position check when the PX4 is turned ON afte the 1-axis and 2-axis positioned for linear interpolation to position the 1-axis and 2-axis for linear interpolation, and to wait until the PX4 is turned OFF.

(2) Q173CPU(N) system setting



	d1	d2	d3	d4	d5	d6	d7	d8
	MR-J2S-10B	MR-J2S-10B						
Regenerative option	Without regene-rative option	Without regene-rative option	Without regene-rative option	Without regene-rative option	Without regene-rative option	Without regene-rative option	Without regene-rative option	Without regene-rative option
	Automatic setting	Automatic setting						
Dynamic brake	None	None	None	None	None	None	None	None
Tolerable movement amount during power OFF	10 (turns)	10 (turns)	0 (turns)	0 (turns)	0 (turns)	0 (turns)	0 (turns)	0 (turns)
	1-axis (INC)	2-axis (INC)	0-axis (INC)	0-axis (INC)	0-axis (INC)	0-axis (INC)	0-axis (INC)	0-axis (INC)

[Unit settings]

• Manual pulse generator input unit Q173PX: Slot 3

P1	Manual pulse generator/synchronous encoder (INC)
P2	Manual pulse generator/synchronous encoder (INC)
P3	Manual pulse generator/synchronous encoder (INC)

I/O response time	0.4ms

Basic settings

[Multi-CPU settings]

Number of multi-CPUs	2 units
Operation mode	All machines are stopped due to error in No. 1 machine/No. 2 machine.

• Automatic refresh setting 1

CPU	Transmission range of each CPU			CPU side device	
	CPU shared memory G			Head device	W0
	Number of points	Head	End	Head	End
No. 1 machine	0	---	---	---	---
No. 2 machine	50	0800	0831	W0	W31
No. 3 machine					
No. 4 machine					

For No. 1 machine, set to "M2400".
(For Q173CPU(N) side, the monitor bit device is transferred to W0 and following by motion SFC program.)

• Automatic refresh setting 2

CPU	Transmission range of each CPU			CPU side device	
	CPU shared memory G			Head device	W100
	Number of points	Head	End	Head	End
No. 1 machine	0	---	---	---	---
No. 2 machine	640	0832	0AB1	W100	W37F
No. 3 machine					
No. 4 machine					

For No. 1 machine, set to "D0".
(For Q173CPU(N) side, the monitor word device is transferred to W100 and following by the motion SFC program.)

• Automatic refresh setting 3

CPU	Transmission range of each CPU			CPU side device	
	CPU shared memory G			Head device	W100
	Number of points	Head	End	Head	End
No. 1 machine					
No. 2 machine					
No. 3 machine					
No. 4 machine					

• Automatic refresh setting 4

CPU	Transmission range of each CPU			CPU side device	
	CPU shared memory G			Head device	W100
	Number of points	Head	End	Head	End
No. 1 machine					
No. 2 machine					
No. 3 machine					
No. 4 machine					

Use for applications other than positioning device for monitor.

[Basic system settings]

Operation cycle setting	Automatic setting
Operation mode	M2000 turns ON when the switch changes from STOP to RUN.
Emergency stop input setting	PX0

• Latch range

	Symbol	Latch (1) head	Latch (1) end	Latch (2) head	Latch (2) end
Internal relay	M				
Link relay	B				
Annunciator	F				
Data register	D				
Link register	W				

Latch (1): Range that can be cleared by Latch clear

Latch (2): Range that cannot be cleared by Latch clear

[PLC setting list]

Unit type	Number of points	Occupied I/O No.	Base	Slot No.	I/O response time
Input	16	000-00F	Main base	1	10ms
Output	16	010-01F	Main base	2	

(3) No. 1 machine QnHCPU PC parameter

1	Number of CPUs		2 units			
2	Operation mode		Error operation mode in the event of CPU stop			
	No. 1 machine		All machines are stopped due to stop error.			
	No. 2 machine		All machines are stopped due to stop error.			
3	Out-of-group input setting		The status is retrieved.			
	Out-of-group output setting		The status is not retrieved.			
4	Refresh setting					
	Setting 1 CPU	Transmission range of each CPU			CPU side	
		CPU shared memory G			Head device	M2400
		Number of points	Head	End	Head	End
	No. 1 machine	0	----	----	----	----
	No. 2 machine	50	0800	0831	M2400	M3199
	Setting 2 CPU	Transmission range of each CPU			CPU side	
		CPU shared memory G			Head device	D0
		Number of points	Head	End	Head	End
	No. 1 machine	0	----	----	----	----
	No. 2 machine	640	0832	0AB1	D0	D639

(4) SFC program list

No.	Program name	Task	Auto-matic start	END operation	Setting of number of continuous shifts	Contents of processing
0	Positioning device	Normal	Yes	–	3	(1) Started automatically, and executed at all times when the Q173CPU(N) is running. (2) Transfers the dedicated positioning device (bit data) for monitor to W0 and following. (3) Transfers the dedicated positioning device (word data) for monitor to W100 and following. When automatic refresh is set, W0 and following are assigned to the QnHCPU's M2400 and following, and W100 and following to D0 and following
20	Main	Normal	Yes	–	3	(1) Started automatically, and executed at all times when the Q173CPU(N) is running. (2) Turns ON the clock data read request (M9028). (3) Starts the "No.110: Motion control" subroutine when the emergency stop is reset. (4) Stops the "No.110: Motion control" in the event of emergency stop, and turns OFF the actual output (PY).
110	Motion control	Normal	No	–	3	(1) Turns ON the servo for all axes. (2) Calls the following program subroutines depending on status of PX1, PX2. • PX2:OFF PX1:OFF "No. 120: JOG" • PX2:OFF PX1:ON "No. 130: Manual pulse generator" • PX2:ON PX1:OFF "No. 140: Zero point return" • PX2:ON PX1:ON "No. 150: Program operation"
120	JOG	Normal	No	–	3	(1) Sets the JOG operation speed for 1-axis and 2-axis. (2) Turns ON the 1-axis JOG forward run command when PX3 is turned ON, and turns ON the reverse run command when PX4 is turned ON. (3) Turns ON the 1-axis JOG forward run command when PX5 is turned ON, and turns ON the reverse run command when PX6 is turned ON. (4) Executes the above steps (2) and (3) repeatedly when PX2 and PX1 are turned OFF (JOG mode). In other cases, the JOG forward run command and JOG reverse run command for 1-axis/2-axis are turned OFF to end the program.
130	Manual pulse generator	Normal	No	–	3	(1) Sets the 1-pulse input magnification for 1-axis and 2-axis. (2) Sets to control the 1-axis by P1, and the 2-axis by P2 to turn OFF the manual pulse generator permit flag for P1 and P2. (3) When PX2 not OFF and PX1 not ON (manual pulse generator mode), the manual pulse generator permit flag is turned OFF for P1 and P2 to end the program.
140	Zero point return	Normal	No	–	3	(1) Starts the "K140 1-axis zero point return program" when PX3 is turned ON, and the "K141 2-axis zero point return program" when PX4 is turned ON. (2) When PX2 is not turned ON and PX1 is not turned OFF (zero point return mode), the program is ended.
150	Program operation	Normal	No	–	3	(1) Waits for 1,000ms after positioning the 1-axis when PX3 changes from OFF to ON to position the 2-axis. (2) Executes the in-position check after positioning the 1-axis/2-axis for linear interpolation when PX4 is turned ON, positions the 1-axis/2-axis for linear interpolation in opposite direction at a double speed, and waits until PX4 is turned OFF. (3) Ends the program when PX1 and PX2 are turned ON (program operation mode).

(5) Motion SFC program detail

Positioning device

P0

```
[F0]
//Status M2400 to M3039 (40 words) of each axis
//M2400 and after (No. 1 machine QnHCPU)
DIN W00L, M2400
DIN W02L, M2432
DIN W04L, M2464
DIN W06L, M2496
DIN W08L, M2528
DIN W0AL, M2560
DIN W0CL, M2592
DIN W0EL, M2624
DIN W10L, M2656
DIN W12L, M2688
DIN W14L, M2720
DIN W16L, M2752
DIN W18L, M2784
DIN W1AL, M2816
DIN W1CL, M2848
DIN W1EL, M2880
DIN W20L, M2912
DIN W22L, M2944
DIN W24L, M2976
DIN W26L, M3008

// Common device M2000 to M2063 (4 words)
// M3040 and following (No. 1 machine QnHCPU)
DIN W28L, M2000
DIN W2AL, M2032

// Special relay M9000 to M9015 (1 word)
// M3104 and following (No. 1 machine QnHCPU)
DIN W2C, M9000

// Special relay M9064 to M9079 (1 word)
// M31110 and following (No. 1 machine QnHCPU)
DIN W2D, M9064
```

(1) The statuses M2400 to M3039 (for 32 axes) of each axis are transferred to W0 and following.
(2) The common devices M2000 to M2064 are transferred to W28 and following.
(3) The special relays M9000 to M9015 are transferred to W2C and following.

Since W0 and following of Q173CPU(N) (No. 2 machine) are so set as to be refreshed to the statues (M2400 and following) of QnHCPU (No. 1 machine) using the multi-CPU automatic refresh, the QnHCPU (No. 1 machine) can grasp the status of Q173CPU(N) (No. 2 machine) by monitoring the following devices.

QnHCPU (No. 1 machine) Device	Device of corresponding Q173CPU(N) (No. 2 machine)
M2400 to M3039	M2400 to M3039
M3040 to M3103	M2000 to M2064
M3104 to M3119	M9000 to M9015

* The data (for 32 axes) is refreshed in this sample example. However, it is advised to minimize the number of refresh points according to the system to shorten the time required for processing.

```
[F1]
// Monitor device D0000 to D0639 (640 words)
// of each axis
//D000 and following (No. 1 machine QnHCPU)
BMOV W100, D0, K640

// Special register D9000 to D9015 (16 words)
//D640 and following (No. 1 machine QnHCPUCPU)
W380 = D9000
W381 = D9005
W382 = D9008
W384L = D9010L
W386L = D9012L
W388L = D9014L

// Special register D9182 to D9197 (16 words)
//D656 and following (No. 1 machine QnHCPU)
W38AL = D9182L
W38CL = D9184L
W38EL = D9186L
W390L = D9188L
W392L = D9190L
W394L = D9192L
W396L = D9194L
W398L = D9196L
```

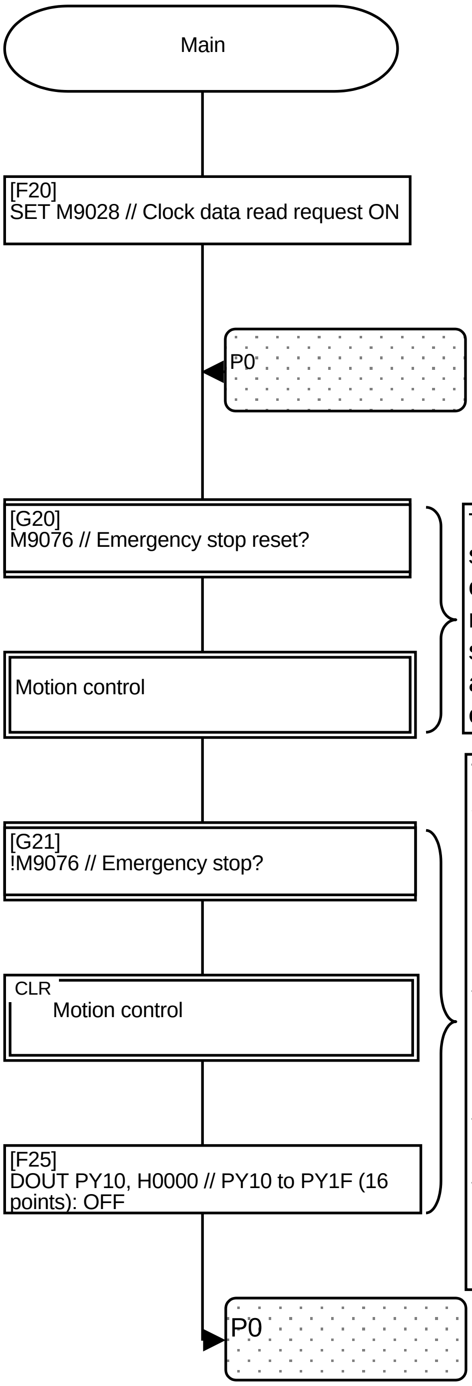
(1) The monitor devices D0 to D639 (for 32 axes) of each axis are transferred to W100 and following.
(2) The special registers D9000 to D9015 are transferred to W380 and following.
(3) The special resisters D9182 to D9197 are transferred to W38A and following.

Since W100 and following of Q173CPU(N) (No. 2 machine) are so set to be refreshed to the statues (D0 and following) of QnHCPU (No. 1 machine) using the multi-CPU automatic refresh, for the QnHCPU (No. 1 machine) to grasp the status of Q173CPU(N) (No. 2 machine) by monitoring the following devices.

QnHCPU (No. 1 machine) Device	Device of corresponding Q173CPU(N) (No. 2 machine)
D0000 to D0639	D0000 to D0630
D0640 to D0655	D9000 to D9015
D0656 to D0671	D9182 to D9197

* The data (for 32 axes) is refreshed in this sample example. However, it is advised to minimize the number of refresh points according to the system to shorten the time required for processing.

P0

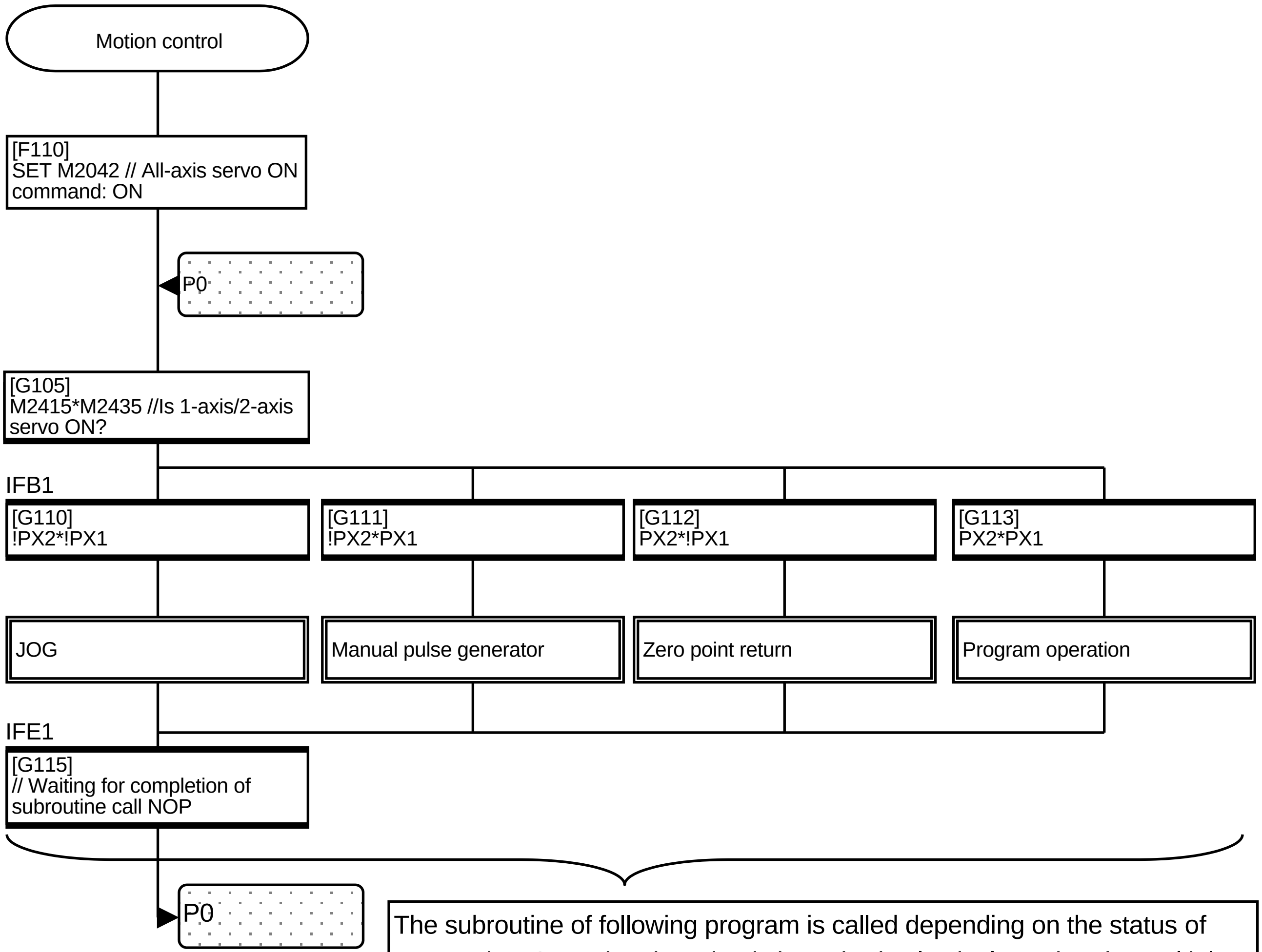


The "No.110: Motion control" subroutine is started when the emergency stop is reset. (Since the next step is shifted, the subroutine is started, and the next step is executed as soon as the subroutine is executed.)

The "No.110: Motion control" subroutine is stopped when the operation is stopped for emergency (the program called from No.110 is also stopped) to turn OFF the actual output (PY).

* The program subroutine started is not stopped. When the subroutine start program is added, therefore, it is stopped as required.
* The actual output is turned OFF as required.
* The detection of servo error occurred as well as emergency stop is added to the stop condition as required.

This sample program is so designed that the program to execute the motion control in the event of emergency stop reset is re-started from the initialization. Consequently, the system example shown is such that the motion control is re-started when the emergency stop is reset after stopping the operation for emergency.



The subroutine of following program is called depending on the status of PX1 and PX2. Each subroutine is branched selectively so that the multiple subroutines are not be called simultaneously. If this program is stopped with the clear step in "No. 20: Main", each subroutine will also stop. Thus, the next step is set to WAIT so that the subroutine can be called.

Status of PX1, PX2		**Subroutine call program**	
PX2	**PX1**	**No.**	**Program name**
OFF	OFF	120	JOG
OFF	ON	130	Manual pulse generator
ON	OFF	140	Zero point return
ON	ON	150	Program operation

JOG

```
[F120]
//1-axis JOG operation speed =
100000PLS/sec
D640L = K100000
//2-axis JOG operation speed =
100000PLS/sec
D642L = K100000
```

```
[G120]
//1-axis forward JOG command SET/RST
SET M3202 = PX3 * !M3203
RST M3202 = !PX3
//1-axis reverse JOG command SET/RST
SET M3203 = PX4 * !M3202
RST M3203 = !PX4
//2-axis forward JOG command SET/RST
SET M3222 = PX5 * !M3223
RST M3222 = !PX5
//2-axis reverse JOG command SET/RST
SET M3223 = PX6 * !M3222
RST M3223 = !PX6

// Performed repeatedly to the end of JOG
mode.
!(!PX1*!PX2)
```

The corresponding JOG command device is set or reset when the PX3 to PX6 signals are turned ON and OFF. In this case, be careful not to allow the forward JOG command and reverse JOG command for the same axis to be turned ON at the same time.

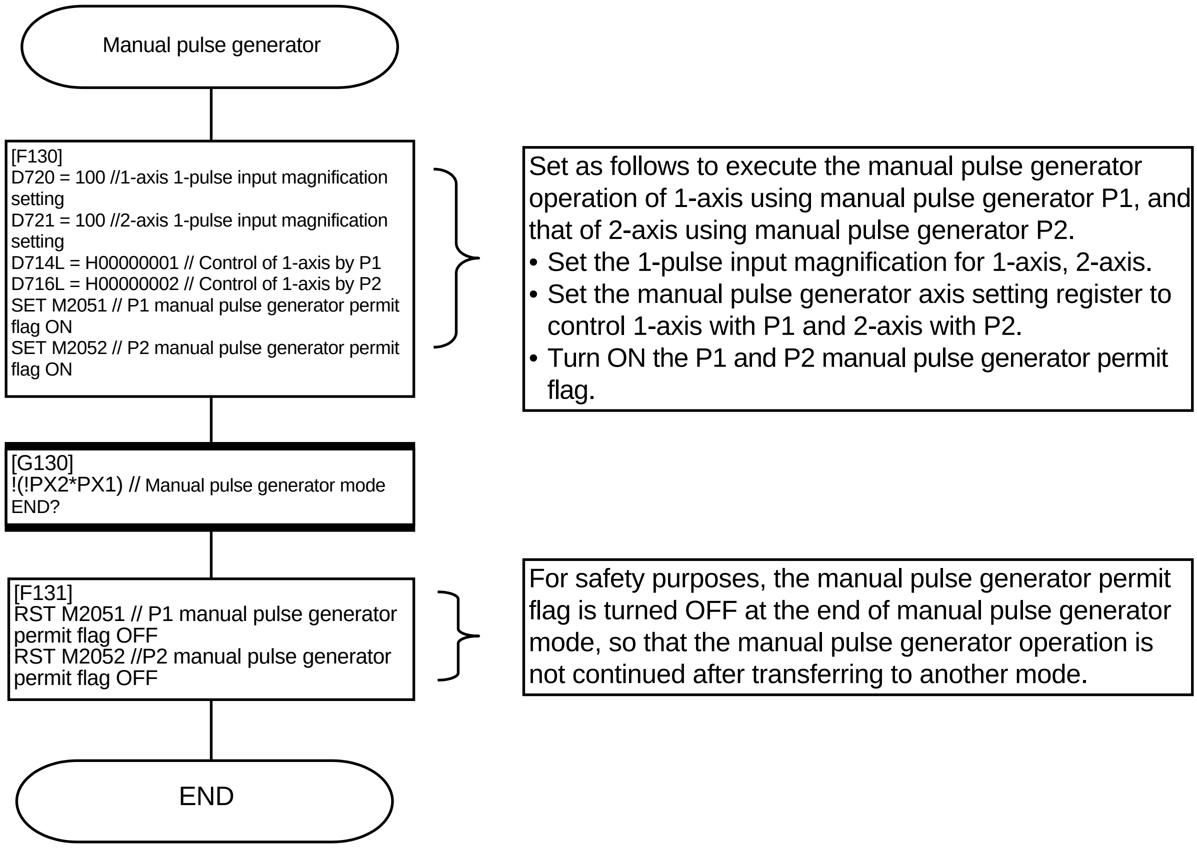
Signal name	Corresponding JOG command device
PX3	M3202 (1-axis forward JOG)
PX4	M3203 (1-axis reverse JOG)
PX5	M3222 (2-axis forward JOG)
PX6	M3223 (2-axis reverse JOG)

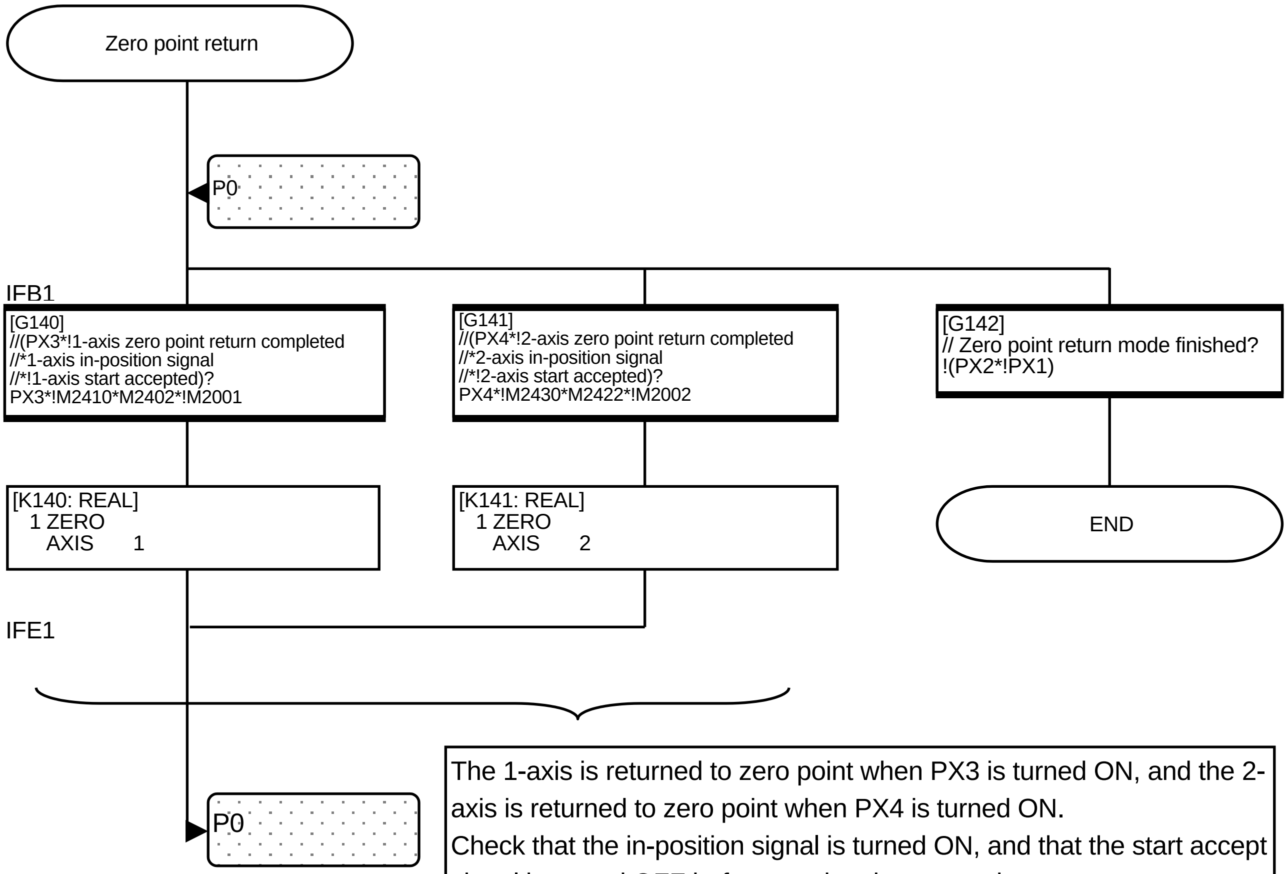
* The ON/OFF judgment of each signal can be described with the Y/N transition. However, if the process can only be described with SEE/RST=:, the number of steps can be reduced and the processing time can be shortened by describing as shown on the left.

```
[F122]
//1-axis, 2-axis forward/reverse JOG
command RST
RST M3202
RST M3203
RST M3222
RST M3223
```

For safety purposes, the 1-axis/2-axis forward/reverse JOG command is turned OFF at the end of the JOG mode, so that the JOG operation does not continue after transferring to another mode.

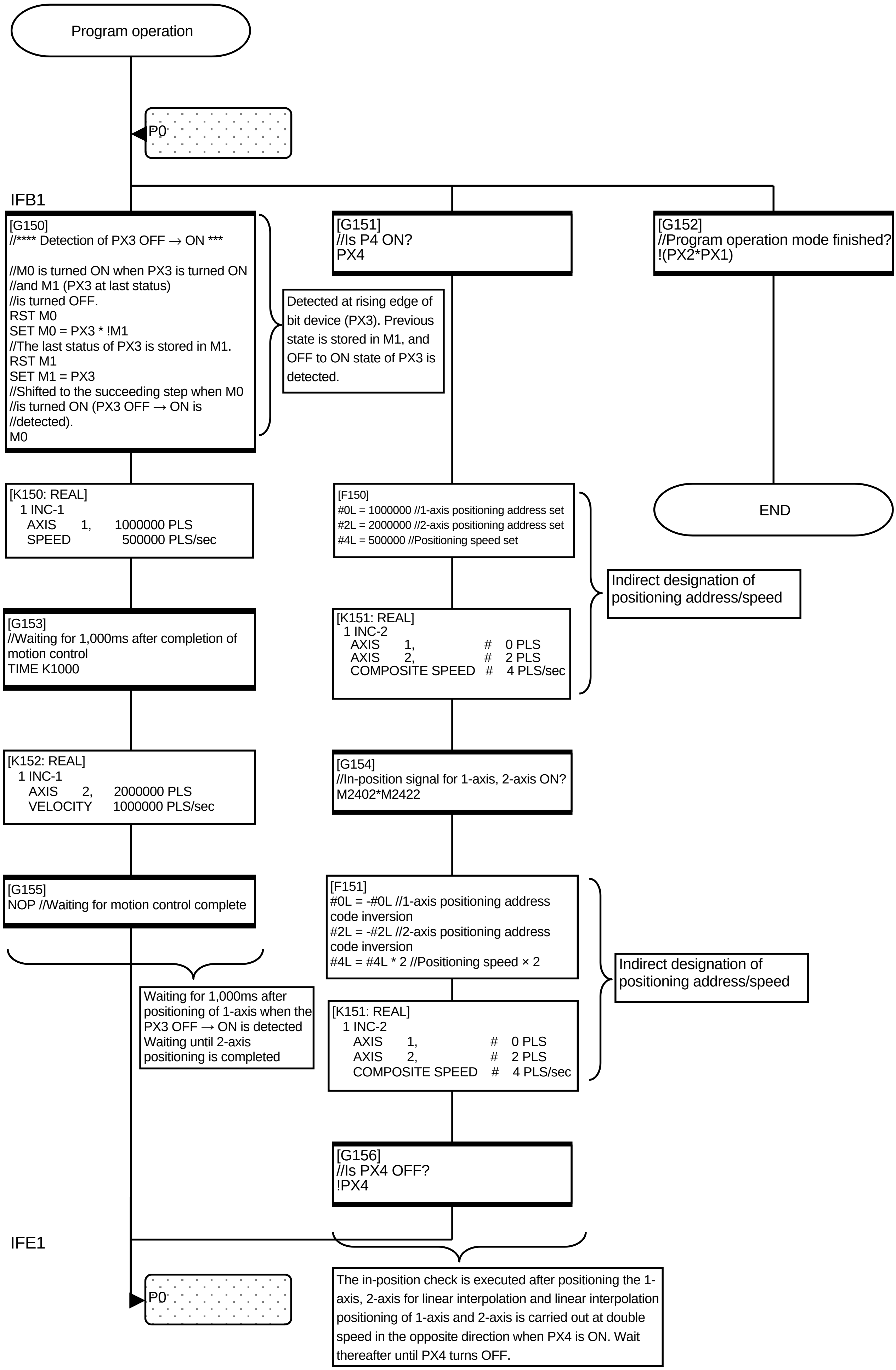
END


Manual pulse generator
[F130]
D720 = 100 //1-axis 1-pulse input magnification setting
D721 = 100 //2-axis 1-pulse input magnification setting
D714L = H00000001 // Control of 1-axis by P1
D716L = H00000002 // Control of 1-axis by P2
SET M2051 // P1 manual pulse generator permit flag ON
SET M2052 // P2 manual pulse generator permit flag ON
Set as follows to execute the manual pulse generator operation of 1-axis using manual pulse generator P1, and that of 2-axis using manual pulse generator P2.
• Set the 1-pulse input magnification for 1-axis, 2-axis.
• Set the manual pulse generator axis setting register to control 1-axis with P1 and 2-axis with P2.
• Turn ON the P1 and P2 manual pulse generator permit flag.
[G130]
!(!PX2*PX1) // Manual pulse generator mode END?
[F131]
RST M2051 // P1 manual pulse generator permit flag OFF
RST M2052 //P2 manual pulse generator permit flag OFF
For safety purposes, the manual pulse generator permit flag is turned OFF at the end of manual pulse generator mode, so that the manual pulse generator operation is not continued after transferring to another mode.
END



The 1-axis is returned to zero point when PX3 is turned ON, and the 2-axis is returned to zero point when PX4 is turned ON.
Check that the in-position signal is turned ON, and that the start accept signal is turned OFF before starting the zero point return program.

* To allow the "K141" to be started during execution of "K140", this program is structed so that there is no WAIT command to wait for the completion of zero point return after the motion control step. (This program must contain the start accept for each axis in the interlock conditions, so that K140 and K141 do not start simultaneously.)

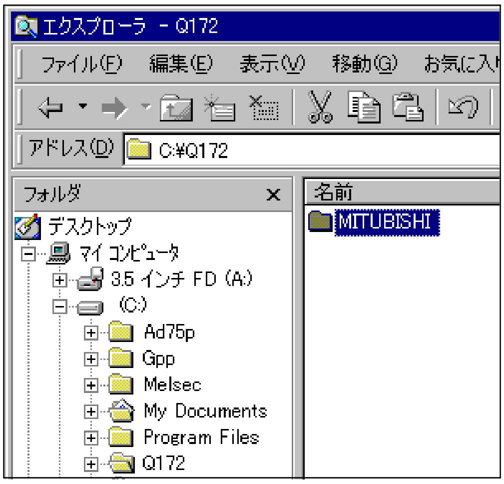

Program operation
P0
IFB1
[G150]
//**** Detection of PX3 OFF → ON ***
//M0 is turned ON when PX3 is turned ON
//and M1 (PX3 at last status)
//is turned OFF.
RST M0
SET M0 = PX3 * !M1
//The last status of PX3 is stored in M1.
RST M1
SET M1 = PX3
//Shifted to the succeeding step when M0
//is turned ON (PX3 OFF → ON is
//detected).
M0
Detected at rising edge of bit device (PX3). Previous state is stored in M1, and OFF to ON state of PX3 is detected.
[G151]
//Is P4 ON?
PX4
[G152]
//Program operation mode finished?
!(PX2*PX1)
END
[K150: REAL]
1 INC-1
AXIS 1, 1000000 PLS
SPEED 500000 PLS/sec
[F150]
#0L = 1000000 //1-axis positioning address set
#2L = 2000000 //2-axis positioning address set
#4L = 500000 //Positioning speed set
Indirect designation of positioning address/speed
[G153]
//Waiting for 1,000ms after completion of motion control
TIME K1000
[K151: REAL]
1 INC-2
AXIS 1, # 0 PLS
AXIS 2, # 2 PLS
COMPOSITE SPEED # 4 PLS/sec
[K152: REAL]
1 INC-1
AXIS 2, 2000000 PLS
VELOCITY 1000000 PLS/sec
[G154]
//In-position signal for 1-axis, 2-axis ON?
M2402*M2422
[G155]
NOP //Waiting for motion control complete
[F151]
#0L = -#0L //1-axis positioning address code inversion
#2L = -#2L //2-axis positioning address code inversion
#4L = #4L * 2 //Positioning speed × 2
Indirect designation of positioning address/speed
Waiting for 1,000ms after positioning of 1-axis when the PX3 OFF → ON is detected
Waiting until 2-axis positioning is completed
[K151: REAL]
1 INC-2
AXIS 1, # 0 PLS
AXIS 2, # 2 PLS
COMPOSITE SPEED # 4 PLS/sec
[G156]
//Is PX4 OFF?
!PX4
IFE1
P0
The in-position check is executed after positioning the 1-axis, 2-axis for linear interpolation and linear interpolation positioning of 1-axis and 2-axis is carried out at double speed in the opposite direction when PX4 is ON. Wait thereafter until PX4 turns OFF.

Appendix 3 Operating the Windows personal computer

Appendix 3.1 Backing up an FD

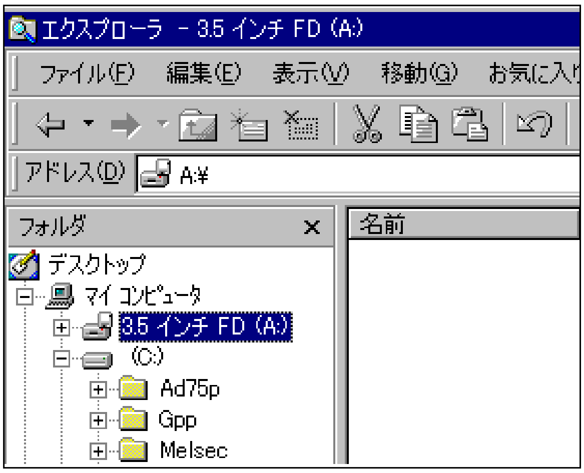
Back up your school textbook.

1) Insert a formatted FD in the FD drive, and click on [Start], [Program] and then [Explorer] to start up Explorer.



2) Select the project folder created with Explorer.

3) Click on [Edit] and then the [Copy] menu in Explorer.

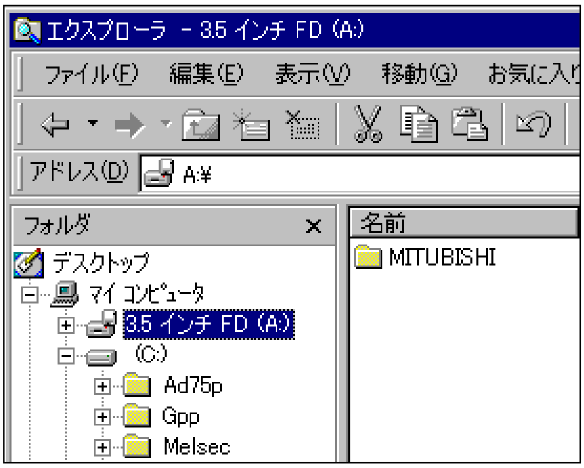


4) Select drive A (3.5-inch FD) with Explorer.

(Continued on next page)

(Continued from previous page)

5) Click on [Edit] and then the [Paste] menu.



6) Copying of the data is completed when the project folder is saved in drive A (3.5-inch FD).
(SW6RN-GSV22P cannot directly recognize a folder in the FD. Read the file after designating the project.)

Appendix 3.2 Installing SW6RN-GSV22P

(1) The SW6RNC-GSV general start-up support software includes the following, each of which is installed as required.
Install SW3RN-SNETP. The other software packages may be installed thereafter in any sequence.

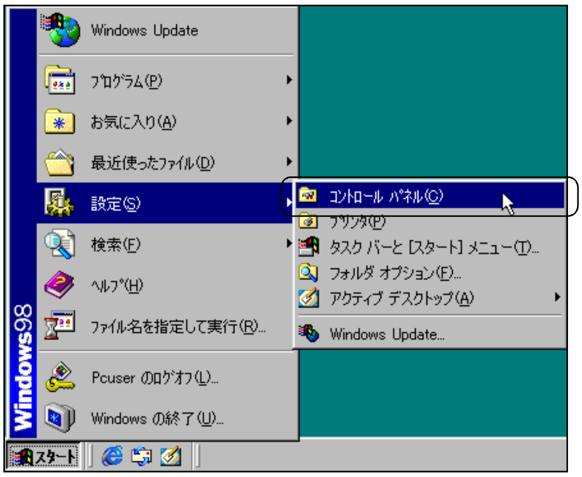
	Type name	Item name	Contents
User application design environment	SW6RN-SNETP	SSCNET communication software	Makes an access to the data at high speed with the motion controller connected to the PC through the SSCNET communication. Supports the communication API for user application software (VB, VC++).
Control program design environment	SW6RN-GSV13P	Independent transfer software	Supports programming, monitor, and test for main unit OS (SW13).
	SW6RN-GSV22P	Automatic machine software	Supports programming, monitor, and test for main unit OS (SW22).
	SW3RN-CAMP	Cam data creation software	Creates the cam pattern for cam control of the main unit OS (SY22). It is necessary when using the cam.
	SW3RN-DOCPRNP SW20RN-DOCPRNP	Document printing software	Converts the program and parameter data set by GSV13P, GSV22P and CAMP into a Word/Excel file format, and to supports printing. Converts the sampling data saved by DOSCP into a Word/Excel file format, dumps it, creates a graph, and supports printing.
Maintenance engineering environment	SW6RN-DOSCP	Digital oscilloscope software	Provides the maintenance engineering environment. Displays the control status of motion system as a waveform, which is effective for the investigation, error cause investigation and analysis.
Contents of HELP	SW3RN-GSVHELP	HELP software	Contains the HELP contents for software shown above.

(2) The following operation environments are available for installing the SW6RN-GSV22P.
Refer to the SW6RNC-GSV/GSVHELP Installation Manual before installing other software as the required open HD capacity differs according to each software.

Item	Contents
Computer main unit	Personal computer (PC/AT compatible) in which Windows NT/98 is running (with Pentium133MHz or larger mounted), or personal computer (PC/AT compatible) in which Windows2000 is running (with Pentium233MHz or larger mounted)
Basic software	Windows NT 4.0 (Service Pack2 or after), Japanese version Windows 98, Japanese version Windows 2000, Japanese version
Memory required	Recommended: 32MB or more (Windows NT/98) Recommended: 64MB or more (Windows 2000)
Open hard disk capacity	15MB or more
Disk drive	3.5-inch (1.44MB) floppy disk drive CD-ROM drive
Display	SVGA resolution (800 × 600) dots, display color: 256 colors or more

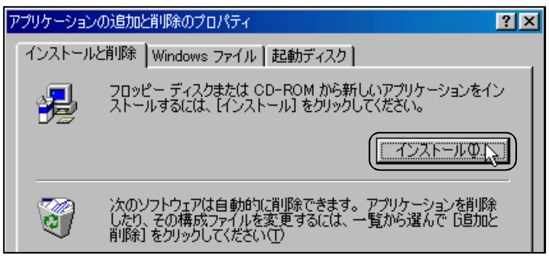
(3) The SW6RN-GSV22P installation procedures are described below.
The other software packages may differ in part, however, the installation procedures are the same.
Refer to SW6RNC-GSV/GSVHELP Installation Manual for details.

1) Turn the personal computer ON, start up Windows, and insert the CD-ROM disk in the CD-ROM drive.



2) Click on [Start], [Setting] and then [Control Panel] in the task bar.

3) Double-click on the Control Panel's "Add/Remove Programs" icon.

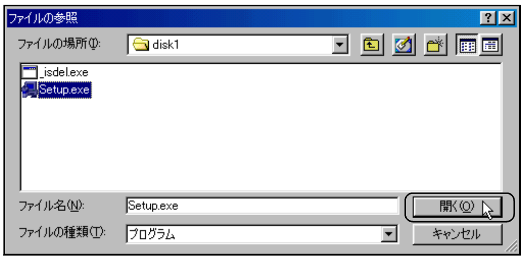


4) The ADD/REMOVE PROGRAM dialog box will open, so click on the Install button.

(Continued from previous page)

5) The INSTALL FROM FLOPPY DISK OR CD-ROM dialog box will open. Click on the Next button.

6) The EXECUTE INSTALLATION PROGRAM dialog box will open, so click on th Reference button.

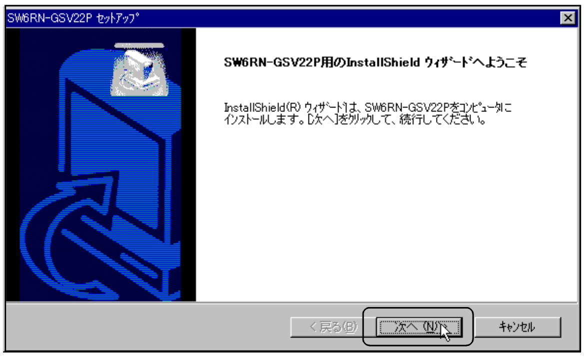


7) The FILE REFERENCE dialog box will open. Designate the drive and folder in the following sequence.
"CD-ROM drive" → "SW6RN-GSV22P" → "00h" → "disk1"
Select the 'Set-up' file in the "disk1" folder, and click on the Open button.

8) Check that the path and file name are displayed at "Installation Program Command Line" in the EXECUTE INSTALLATION PROGRAM dialog box, and click on the Completed button.

(Continued on next page)

(Continued from previous page)



9) A screen showing the cautions and warnings will open.
Click the Next button.

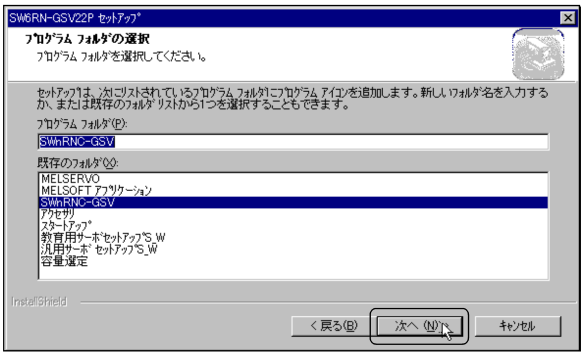
10) Input your "Name" and "Company name", and click on the Next button.

11) The SELECT INSTALLATION DESTINATION dialog box will open
The default is 'C:\Program files\'.
If the installation destination does not need to be changed, click on the Next button to skip to step 13).
If the installation destination must be changed, click on the Reference button to advance to step 12).

12) Select the folder to be installed from the SELECT DIRECTORY dialog box, and click on the OK button.
Check that the selected folder is displayed at "Install destination folder" in the SELECT INSTALLATION DESTINATION dialog box, and click on the Next button.

(Continued on next page)

(Continued from previous page)



13) Input the program folder name to be registered in the Start menu.
The default is 'SWnRNC-GSV'.
If the program folder name does not need to be changed, click on the Next button.

If the program folder name must be changed, select the program folder name from "Existing folder", or input the folder name to be newly created in "Program folder", and click on the Next button.

Installation will start.

14) When the installation is completed, the message indicating that SW6RN-SNETP is required will appear.
Click on the OK button.

15) The SET-UP COMPLETED dialog box will open
Select the necessary items such as 'Display REDME.TXT file' or 'Start SW6RN-GSV22P', and click on the Completed button.

16) This completes installation of SW6RN-GSV22P.

Appendix 4 Comparison between A173UHCPU/A172SHCPUN

The following shows the comparison between Q173CPU(N)/Q172CPU(N) and A173UHCPU/A172SHCPUN.

	Item	Q173CPU(N)	Q172CPU(N)	A173UHCPU	A172SHCPUN
Motion control	Number of control axes	32-axis	8-axis	32-axis	8-axis
Motion control	Control operation cycle – SV13	1 to 8-axis: 0.88ms 9 to 16-axis: 1.77ms 17 to 32-axis: 3.55ms (Settable by parameter)	1 to 8-axis: 0.88ms (Settable by parameter)	1 to 20-axis: 3.55ms 21 to 32-axis: 7.11ms	1 to 8-axis: 3.55ms
Motion control	Control operation cycle – SV22	1 to 4-axis: 0.88ms 5 to 12-axis: 1.77ms 13 to 24-axis: 3.55ms 25 to 32-axis: 7.11ms (Settable by parameter)	1 to 4-axis: 0.88ms 5 to 8-axis: 1.77ms (Settable by parameter)	1 to 12-axis: 3.55ms 13 to 24-axis: 7.11ms 25 to 32-axis: 14.22ms	1 to 8-axis: 3.55ms
Motion control	Program capacity	At step of 14K			At step of 13K
Motion control	Number of positioning points	32,000 points/axis (Indirect setting possible)			
Motion control	Program tool	PC/AT compatible machine		PC/AT PC9800 Series compatible machine (A30TU, A31TU)	
Motion control	Peripheral unit I/F	USB (12Mbps), RS-232C (Max. 115Kbps), SSCNET (5.6Mbps)		RS-422 (9600bps)/SSCNET (5.6Mbps)	
Motion control	Zero point return function	Contact dog type, count type, data set type 1, data set type 2		Contact dog type, count type, data set type 1	
Motion control	Manual pulse generator operation function	3 units are connectable.			1 unit is connectable.
Motion control	Synchronous encoder operation function	12 units are connectable.	8 units are connectable.	4 units are connectable.	1 unit is connectable.
Motion control	Limit switch output function	–		ON/OFF setting point of each axis can be set up to 10 points. (Number of output points: 8 points/axis)	
Motion control	Limit switch output function	Number of output points: 32 points, watch dog data: motion control data/word device			
Motion control	Number of SSCNET I/Fs	5CH *1	2CH	4CH	2CH
Motion control	Number of motion slots	64 slots (max.) (Number of stages extended for Q Series: 7 stages, max.)		8 slots	2 slots
Motion control	Pulse generator, synchronous encoder External signal input unit Number of units attached	It is possible to use 4 Q172LX units. It is possible to use 6 Q172EX units. It is possible to use 4 Q173PX units.	It is possible to use 1 Q172LX units. It is possible to use 4 Q172EX units. It is possible to use 3 Q173PX units.	It is possible to use 4 A172SENC units.	It is possible to use 1 A172SENC units.
Motion SFC function	Execution specification – Normal task	Executed at motion main cycle.			
Motion SFC function	Execution specification – Execution task – Event task (maskable) – Constant cycle	Executed at every constant cycle (0.88ms, 1.77ms, 3.55ms, 7.11ms and 14.22ms).		Executed at every constant cycle (1.77ms, 3.55ms, 7.11ms and 14.22ms).	
Motion SFC function	Execution specification – Execution task – Event task (maskable) – External interrupt	Executed when the setting inputs of those (16 points) input to the external interrupt unit QI60 are turned ON.		Executed when the setting inputs of those (16 points) input to the external interrupt unit AI61 are turned ON.	
Motion SFC function	Execution specification – Execution task – Event task (maskable) – PLC interrupt	Executed at interrupt from PLC.		Executed at interrupt (1 point) from PLC.	
Motion SFC function	Execution specification – Execution task – NMI task	Executed when the setting inputs of those (16 points) input to the external interrupt unit QI60 are turned ON.		Executed when the setting inputs of those (16 points) input to the external interrupt unit AI61 are turned ON.	
Motion SFC function	Number of input/output (X/Y) points	8192 points			2048 points
Motion SFC function	Number of actual input/output (PX/PY) points	256 points			
Motion SFC function	Device – Number of internal relay (M) points Number of latch relays (L) points	8,192 points: Total for M and L		8,192 points: Total for M and L (S)	2,048 points: Total for M and L (S)
Motion SFC function	Device – Number of link relay (B) points	8192 points			1024 points
Motion SFC function	Device – Number of annunciator (F) points	2048 points			256 points
Motion SFC function	Device – Number of timer contact (TT) points	–		2048 points	256 points
Motion SFC function	Device – Number of timer coil (TC) points	–		2048 points	256 points
Motion SFC function	Device – Number of counter contact (CT) points	–		1024 points	256 points
Motion SFC function	Device – Number of counter coil (CC) points	–		1024 points	256 points
Motion SFC function	Device – Number of special relay (M) points	256 points		256 points	256 points
Motion SFC function	Only for inside of motion CPU – Number of data register (D) points	8192 points		8192 points	1024 points
Motion SFC function	Only for inside of motion CPU – Number of link register (W) points	8192 points		8192 points	1024 points
Motion SFC function	Only for inside of motion CPU – Number of current timer value (T) points	–		2048 points	256 points
Motion SFC function	Only for inside of motion CPU – Number of current counter value (C) points	–		1024 points	256 points
Motion SFC function	Only for inside of motion CPU – Number of special register (D) points	256 points		256 points	
Motion SFC function	Only for inside of motion CPU – Number of motion device (#) points	8192 points		8192 points	
Motion SFC function	Only for inside of motion CPU – Number of coast timer (FT) points	1 point (888μs)		1 point (888μs)	

*1 The wire dividing unit (Q173DV) or branch cable (Q173J2B∆CBL☐M/Q173HB∆CBL☐M) is used.

Items			Q173CPU(N)	Q172CPU(N)	A173UHCPU	A172SHCPUN
Others	Device memory		Independent		Common	
	Data exchange between PCPU and SCPU		Data exchange method using automatic refresh between multi-CPU		Direct data exchange method using device memory for 2-port memory	
	Fixed parameter	Number of pulses per rotation	1 to 2147483647 [pls]		1 to 65535 [pls]	
		Movement amount per rotation	Unit setting PLS: 1 to 2147483647 [pls]		Unit setting PLS: 1 to 65535 [pls]	
		Unit magnification	–		×1, ×10, ×100, ×1000	
	PLC READY flag (X2000)		M2000 is set to RUN when the switch changes from STOP to RUN, or M2000 is turned ON when the switch changes from STOP to RUN and the setting register is set to "1".		Execution of M2000 from sequence program	
	Emergency stop input		Optional bit device (PX, M) is designated by parameter.		Emergency stop terminal for main base	
	Back-up battery for internal memory		Secondary battery (built-in charging type) If the power interruption continues for over one month, an external type primary battery (A6BAT) is installed additionally.		Primary battery (A6BAT)	
	Outside dimension [mm]		122.4(H) × 27.4(W) × 89.3(D): Q173CPU/Q172CPU 104.4(H) × 27.4(W) × 114.3(D): Q173CPUN/Q172CPUN		130(H) × 220(W) × 110(D)	

Additional function/applicable version list

The combination of versions to which additional function is applicable is as shown below.

Functions		Main unit OS version (*1)	Peripheral S/W version	CPU version (*2)	
				Q173CPU	Q172CPU
ROM operation function		H	H	N	M
Improvement of automatic refresh function for shared memory of CPU		H	H	N	M
Communication through network		H	H	N	M
Main cyclic monitor		D	D	–	–
Reading of servo parameters from servo amplifier		D	–	–	–
Motion SFC command					
	MULTR	D	D	–	–
	MULTW	D	D	K	J
	OUT	D	D	–	–
	FROM	H	H	–	–
	TO	H	H	–	–
Addition of dedicated motion CPU commands (SVST command, etc.)		H	–	N	M

(*1) Same version for all SV13/SV22

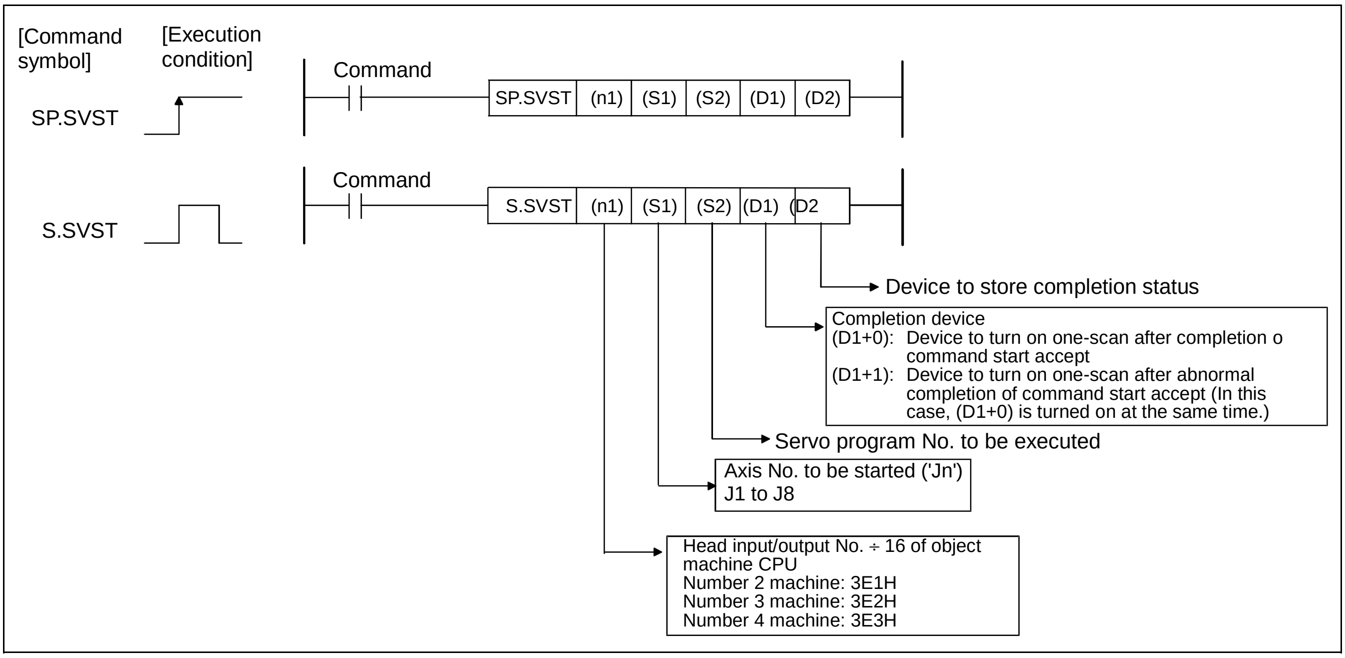
(*2) The countermeasure has been taken to all Q172CPUN/Q173CPUN (fan-less product).

Appendix 5 Sequence command dedicated to motion

This appendix describes the details of the SVST command, CHGA current value change command, CHGV speed change command and CHGT torque change command.

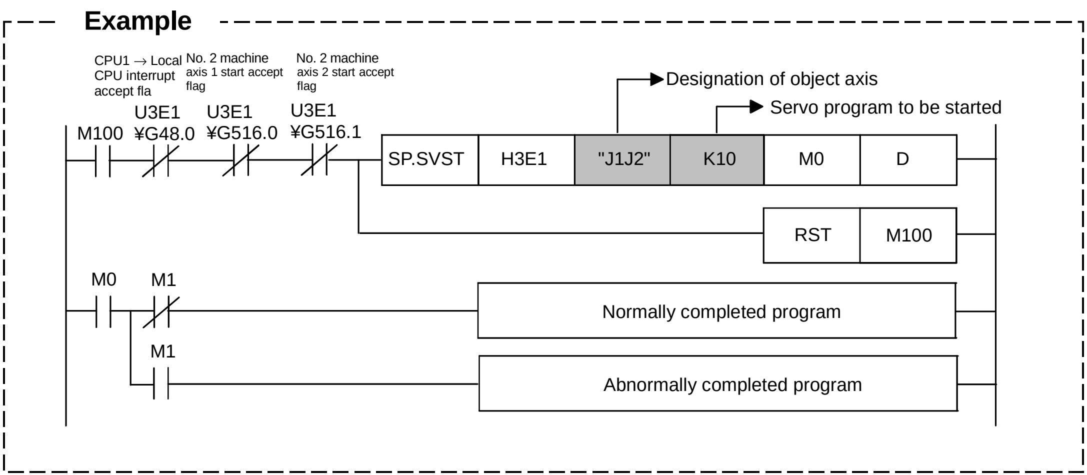
Appendix 5.1 SVST servo program start request command

This command is used to request starting of the designated servo program for start.



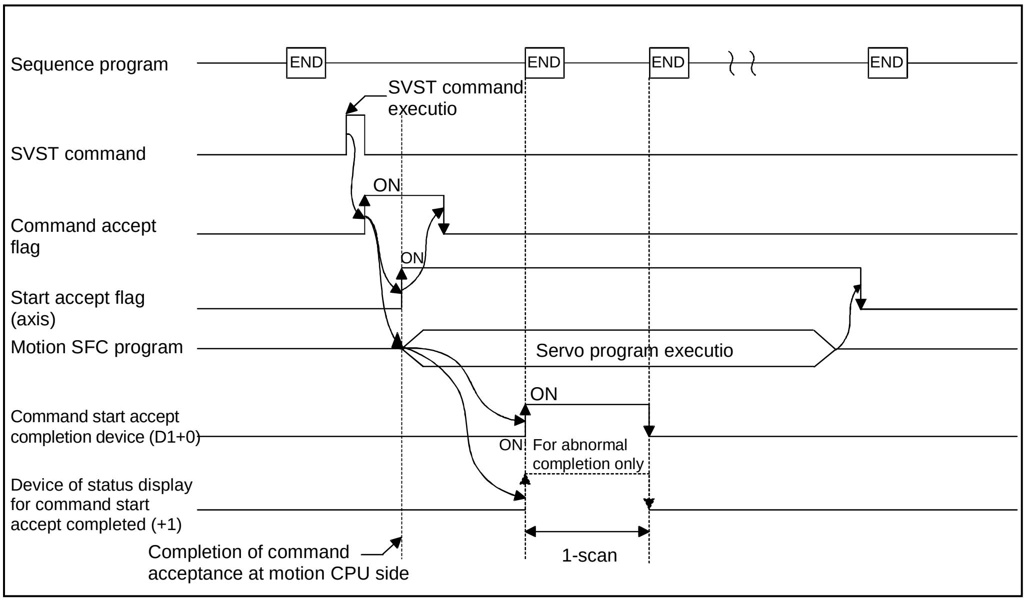
(1) SVST command program example

This program is used by the No. 1 machine's PLC CPU to request starting of the servo program No.10 for the axis No. 1 and 2 of the No. 2 machine's motion CPU.



(2) Execution timing

Starting of the designated servo program is requested at the rising edge of the SVST command (OFF → ON).



(3) Operation error conditions

An operation error occurs in the following cases and the SVST command is not executed.

(a) When the machine (set for reservation) is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(b) When the local machine is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

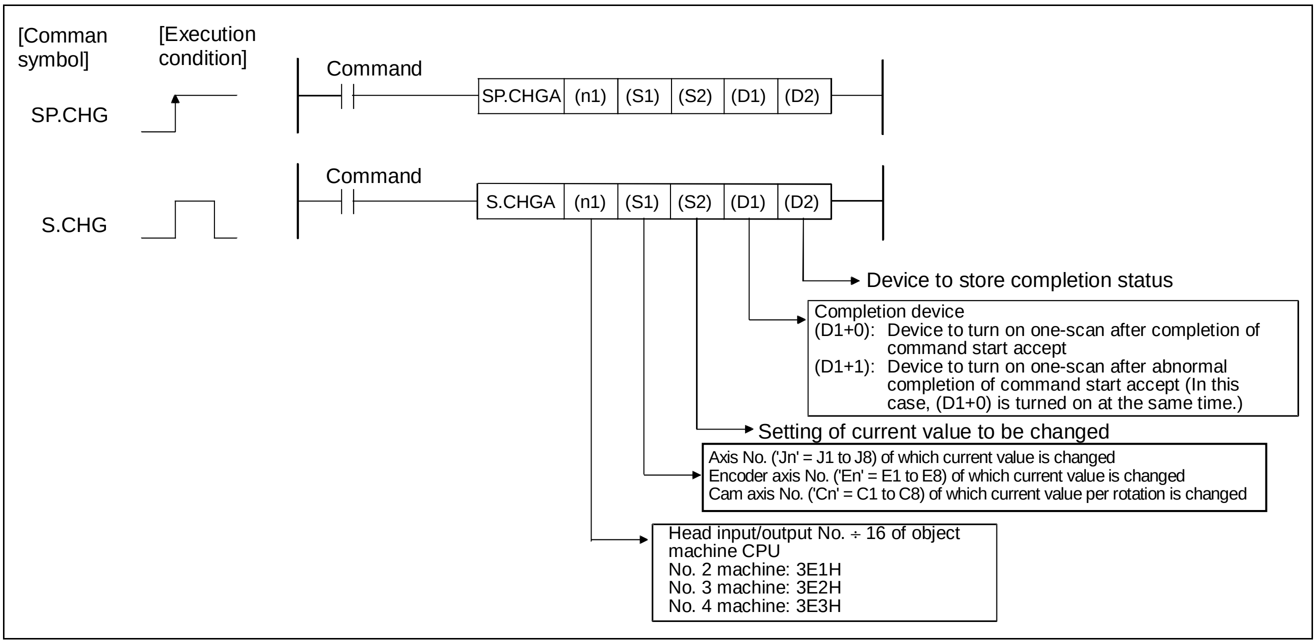
(c) When a CPU other than Q motion CPU is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(d) When the command is configured of devices other than usable device

(e) When 0 to 3DFH, 3E4H and following are designated by object machine CPU head No. input/output No. ÷ 16 [nl]

Appendix 5.2 CHGA current value change command

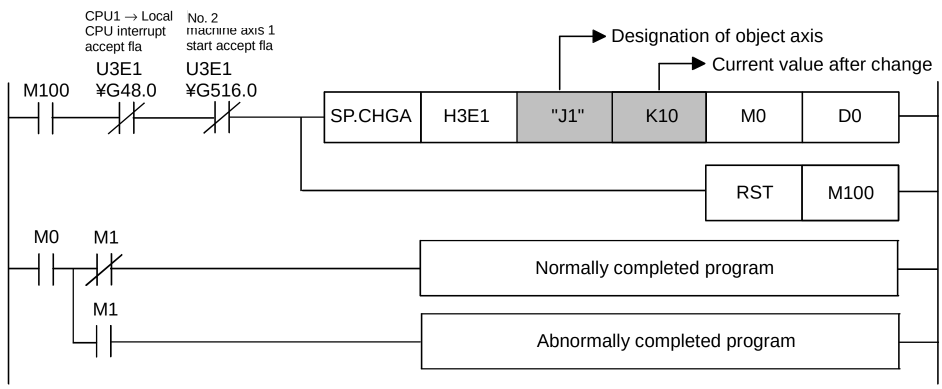
This command is used to change the current value of a stopped axis.



(1) CHGA command program example

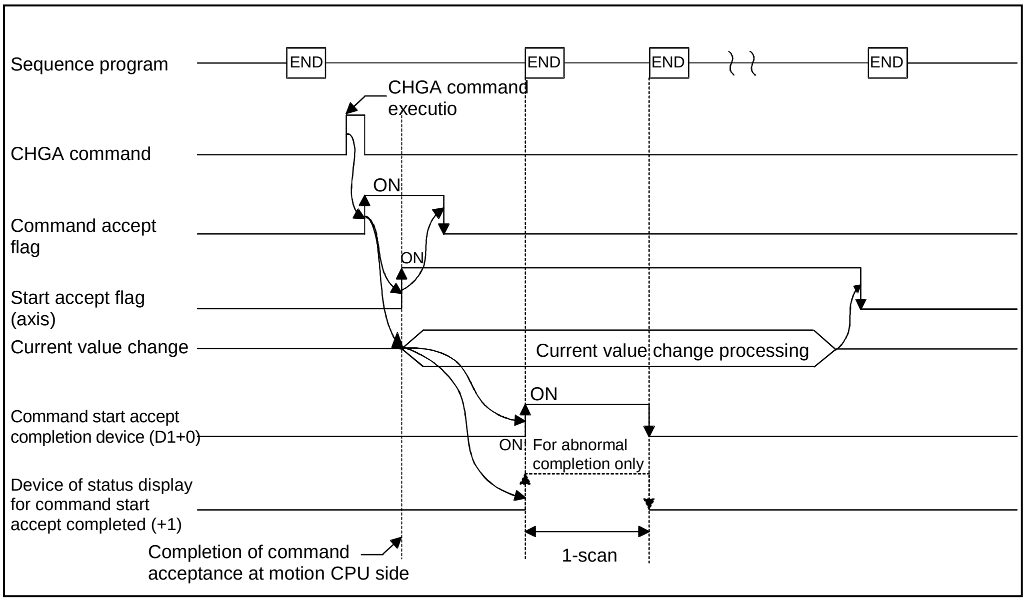
This program is used to change the current value of axis No. 1 (No. 2 machine motion CPU) from that of axis No. 1 (No. 1 machine PLC CPU) to "10".

Example



(2) Execution timing

The current value is changed for a designated axis at the rising edge (OFF → ON) of the CHGA command.



(3) Operation error conditions

An operation error occurs in the following cases and the SVST command is not executed.

(a) When the machine (set for reservation) is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(b) When the local machine is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

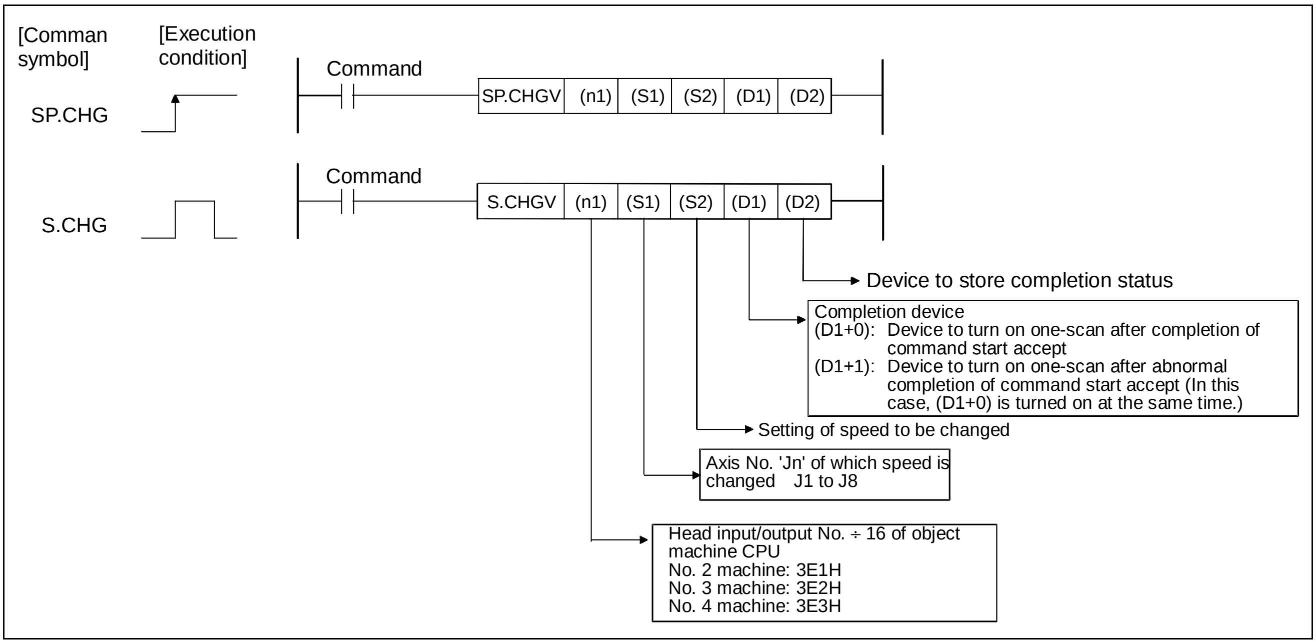
(c) When a CPU other than Q motion CPU is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(d) When the command is configured of devices other than usable device

(e) When 0 to 3DFH, 3E4H and following are designated by object machine CPU head No. input/output No. ÷ 16 [nl]

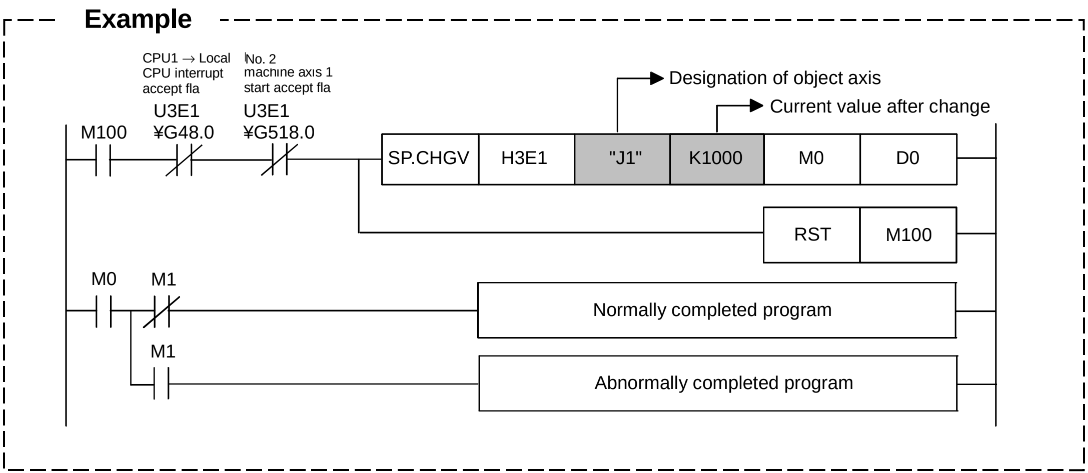
Appendix 5.3 CHGV speed change command

The CHGV speed change command is used to change the speed during positioning and JOG operation.



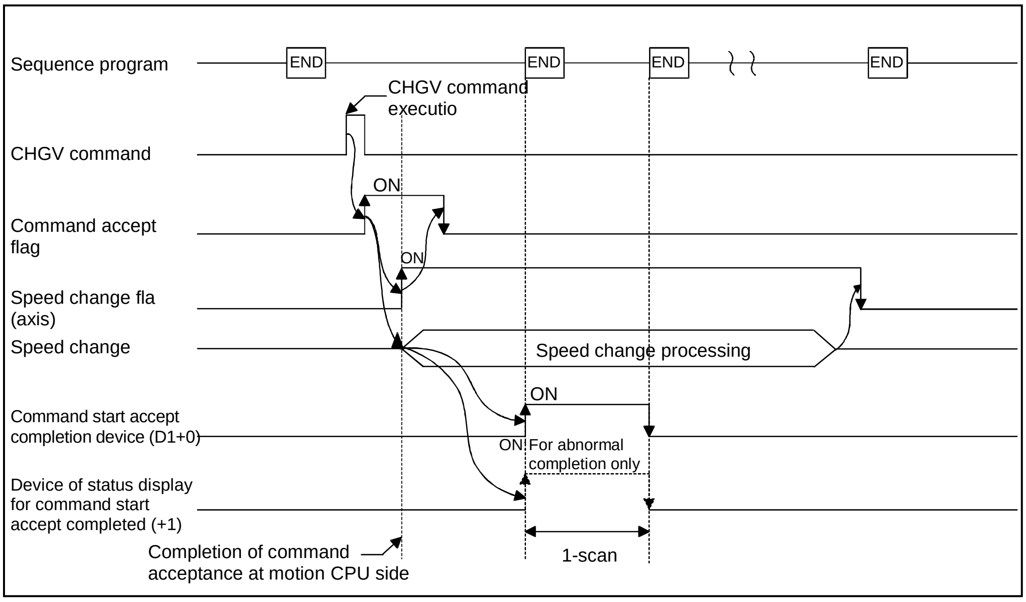
(1) CHGV command program example

This program is used to change the positioning speed of axis No. 1 (No. 2 machine motion CPU) from that of axis No. 1 (No. 1 machine PLC CPU) to "1000".



(2) Execution timing

The speed is changed for the designated axis at the rising edge (OFF → ON) of the CHGV command.



(3) Operation error conditions

An operation error occurs in the following cases and the SVST command is not executed.

(a) When the machine (set for reservation) is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(b) When the local machine is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

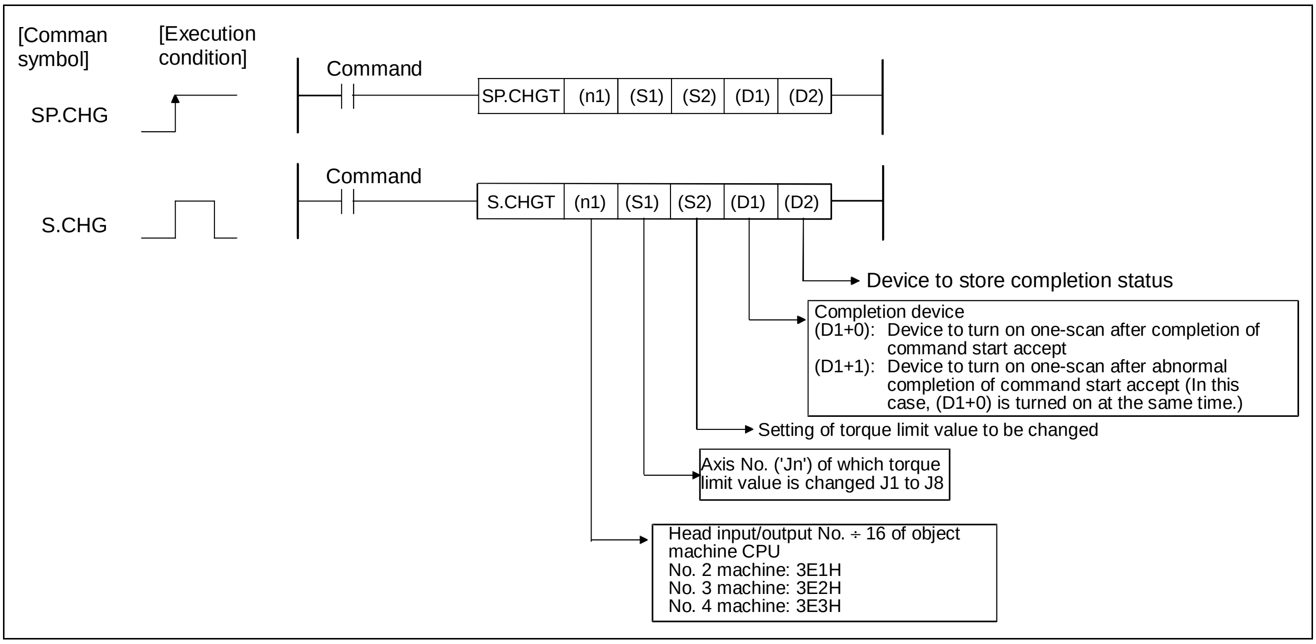
(c) When a CPU other than Q motion CPU is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(d) When the command is configured of devices other than usable device

(e) When 0 to 3DFH, 3E4H and following are designated by object machine CPU head No. input/output No. ÷ 16 [nl]

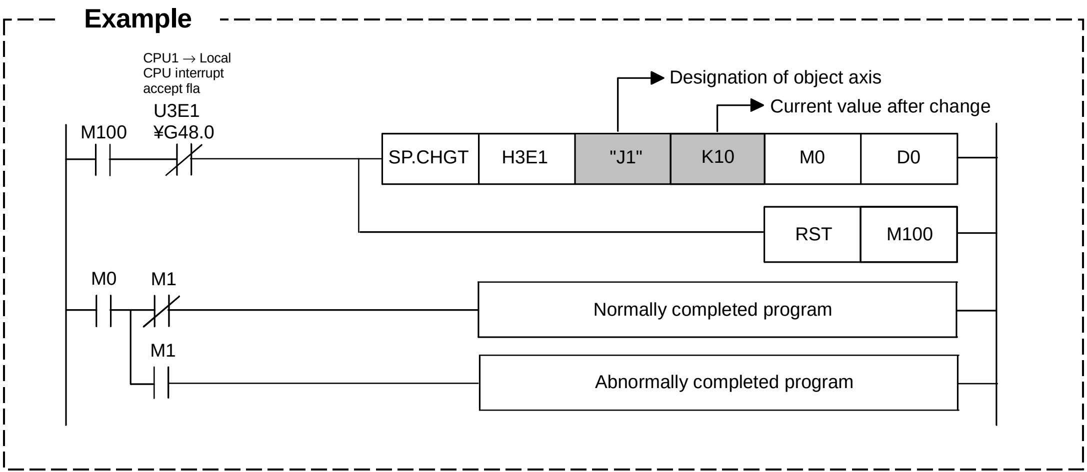
Appendix 5.4 CHGT torque limit value change request command

This command is used to change the torque limit value regardless of whether the operation is executing or stopping in the real mode.



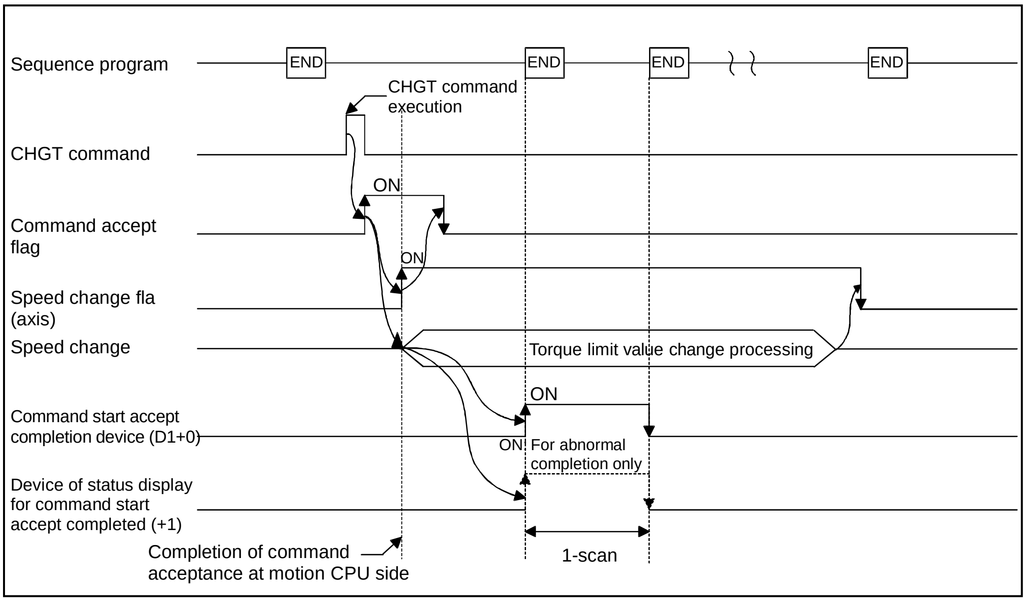
(1) CHGT command program example

This program is used to change the torque limit value of axis No. 1 (No. 2 machine motion CPU) from that of axis No. 1 (No. 1 machine PLC CPU) to "10%".



(2) Execution timing

The torque limit value is changed for the designated axis at the rising edge (OFF → ON) of the CHGT command.



(3) Operation error conditions

An operation error occurs in the following cases and the SVST command is not executed.

(a) When the machine (set for reservation) is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(b) When the local machine is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(c) When a CPU other than Q motion CPU is designated by object machine CPU head No. input/output No. ÷ 16 [nl]

(d) When the command is configured of devices other than usable device

(e) When 0 to 3DFH, 3E4H and following are designated by object machine CPU head No. input/output No. ÷ 16 [nl]

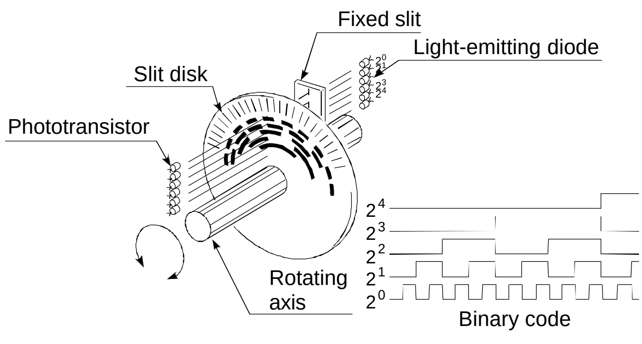
Appendix 6 Explanation of terms

A ACCELERATION

Refers to the cam's dimensionless acceleration rate.
The dimensionless acceleration rate is the dimensionless speed differentiated by the dimensionless time.
The maximum value is expressed as Am.
Refer to the term "Am".
Refer to the term "V".

ABSOLUTE ENCODER

This is a absolute position detector that enables the angle data within 1 motor rotation to be output to an external destination.
Absolute encoders are generally able to output 360° in 8 to 12 bits.
Incremental encoders have a disadvantage in that the axis position is lost when a power failure occurs. However, with absolute encoders, the axis position is not lost even when a power failure occurs.
Various codes such as a binary code and BCD code can be output.
Absolute encoders are more expensive, more accurate, and larger than incremental encoders. Refer to "ENCODER".

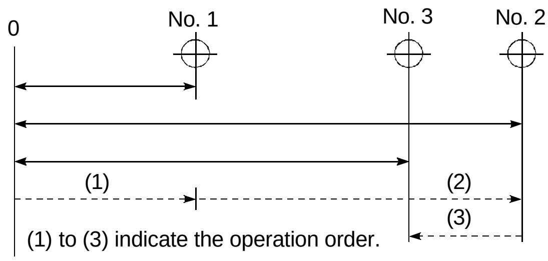


ABSOLUTE POSITION SYSTEM

If zero point return is carried out once after starting up the positioning control devices, the current value will be backed up with a battery when the power is turned OFF, and the machine deviation will be compensated even if it occurs.
Thus, zero point return is not required after the power is turned ON again.
A servomotor with absolute position detector and a compatible servo amplifier are required to structure this system.

ABSOLUTE SYSTEM

This is one system for expressing a positioning address.
This system uses 0 as a reference, and expresses the address as the distance from 0.
The positioning direction is automatically determined, even when it is not designated.
The other address system is the increment system.

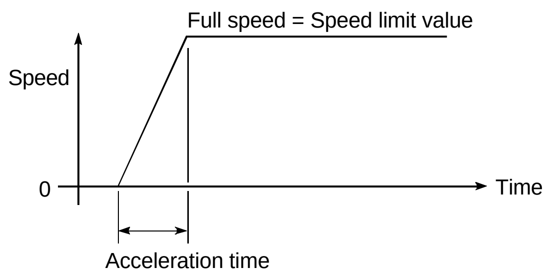


ACCELERATION

Acceleration is obtained from speed differentiated by time and refers to the change rate of speed. Acceleration is in proportion to stress. Refer to the term A

ACCELERATION TIME

Time for the full speed to be reached from the stopped state with the motion controller.
The parameter acceleration/deceleration refers to the time to reach the speed limit value, and if the set speed is low, the time will be proportionally shorter.
This is determined by the machine's inertia, motor torque and load's resistance torque, etc.



AC MOTOR DRIVE UNIT

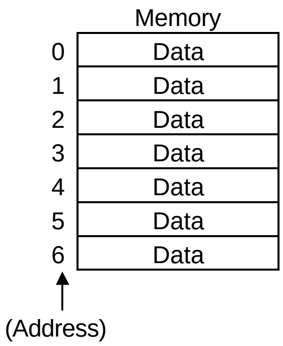
The AC motor drive unit is a built-in type servo amplifier that can drive one connected servomotor.

ACTUAL CURRENT VALUE

Number of pulses for real servo movement, calculated from the feedback pulses.

ADDRESS

1) The memory address. Many addresses are stored in the memory. An address is read or written after it is designated.



2) For positioning, this is a numerical value to indicate the target position, designated in mm, inch, angle or No. of pulse units.

Am ACCELERATION

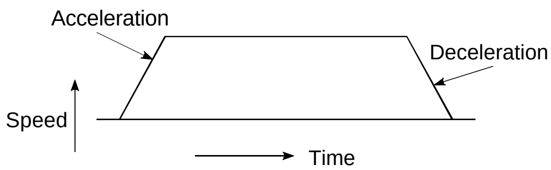
Refers to the maximum value of the cam's dimensionless acceleration rate.
Refer to the term "A".

ANALOG COMMAND

Converts command pulse to an analog voltage in positioning unit, and outputs converted analog voltage to servomotor drive unit.
The motion controller has no dedicated unit that can command this analog command.
This can be used independently with the MELSEC-A AD72 or AD70 type analog output positioning unit.

AUTOMATIC TRAPEZOIDAL ACCELERATION/DECELERATION

Operation that creates a trapezoidal time and speed graph during positioning.

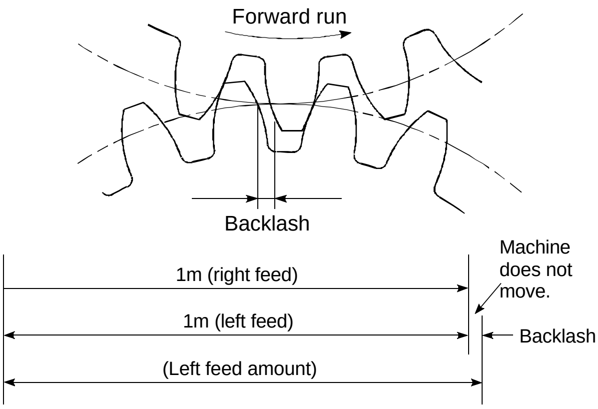


AUTO TUNING (Automatic Tuning)

Properties such as responsiveness and stability of machines driven with a servomotor are affected by changes in the inertia moment and rigidity due to changes in the machine load, etc.
This function automatically adjusts the speed loop gain and position loop gain to match the machine state, so the machine's performance can be maintained at its optimum state.

BACKLASH COMPENSATION

When a forward run operation changes to a reverse run operation, there is sometimes play (backlash) in the mesh of the toothed gears. This also occurs when using a worm gear. Because of this backlash, a left feed of 1m carried out after a right feed of 1m will not be sufficient to return the machine to its original position. The machine cannot be positioned to its original position without an extra feed equivalent to the backlash amount. This is similar to the "play" of a car's handle.

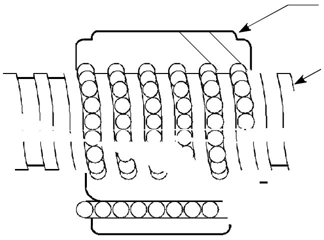


BACKUP FUNCTION

1) Function that saves the sequence programs and device states in the PLC CPU's RAM memory even when the power is OFF.
2) Function that saves the current value for the absolute position compatible system even when the power is OFF.
3) When the CPU unit is replaced, the CPU data (servo programs, servo parameters, absolute position compatible data, etc.) are read out, and reloaded after the CPU is replaced.

BALL SCREW

This is a type of screw, with balls lined up in the threads like ball bearings. This is used for positioning as the backlash is small, and rotation is possible with little force.
Refer to the term "FEED SCREW".



BASE SHUT-OFF

The servo amplifier supplies power to the servomotor by switching the power transistor. Thus, the base is shut off to stop the power supply to the servomotor when the servo is turned OFF or when an alarm occurs, etc. The servomotor will be free run at this time.

BLANK COVER MODULE

An empty unit used to improve the appearance of blank slots in the main base and extension base.

BOTTOM DEAD CENTER

Refers to the lower side of the machine installation path during reciprocating operation of the cam mechanism. Lowest section of the cam.
Refer to the terms "RECIPROCATING CAM" and "FEED CAM".

CAM

Machine element that conveys a specific operation by directly contacting a section with a simple-shape contactor such as a knife edge, roller or plane.

CAM CURVE

An operation curve of the follower member moved with the cam. this can be set with the software package (SW0SRX-CAMP). The cam curve names include uniform speed, uniform acceleration,
5th power, cycloidal, modified trapezoid, modified sine, modified constant speed, trapechloid, multi-chord and harmonic, etc.

CAMP

Refers to the software package (SW3RN-CAMP) used to create the cams for the virtual mode's cam output.

CHANGE signal

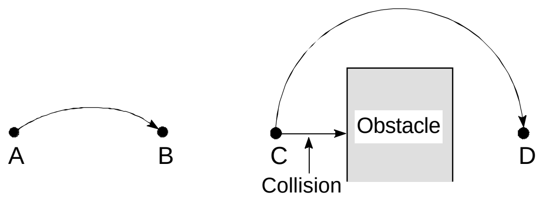
The CHANGE signal is an external signal used to change the speed/position control from the speed control being executed to position control.

CHARACTERISTICS OF CAM CURVES

Refers to the speed and acceleration of the cam curve.

CIRCULAR INTERPOLATION

Automatic operation, in which the CPU makes calculations so that an arc can be drawn when positioning is carried out by simultaneously operating both the longitudinal feed and latitudinal feed motors.
Round shapes can be created by designating the auxiliary point, radius and center point, and if there are obstacles, these can be avoided.
Refer to the term "LINEAR INTERPOLATION".



COMMAND IN-POSITION

Signal that positioning data's fixed parameters, which detects the difference of the positioning address (command position) and current feed value, and turns ON when the result matches the setting value.
This is used to detect the positioning end point address slightly ahead of its position to carry out work beforehand.

CONSTANT SPEED CONTROL (Uniform speed control)

With one start command, the positioning control to the preset pass point with linear or circular movement, and carries out positioning to the end positioning to the end point at a set speed.
The same control of the pass points can be repeated by using the FOR/NEXT command.

CONSTANT VELOCITY CURVE

Suitable when uniform speed movement is required.

CONTROL UNIT

This is one type of positioning reference data.
The unit to be used is designated as mm, inch, degree, or pulse.

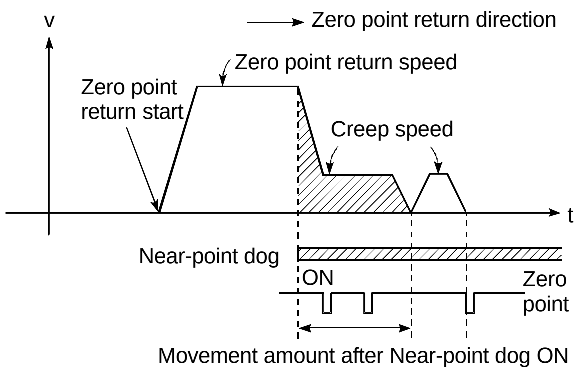
In Japan, it's either mm or degree. In the US, it's either inch or degree.	You can also use pulses!

COPY

COPY means to transfer a part of the EDIT screen to another location.

COUNT TYPE ZERO POINT RETURN

During zero point return, decelerates to the creep speed when the near-point dog turns ON. After moving the movement amount from after the near-point dog turns ON, sets the next zero point signal position as the zero point address.
The near-point dog length can be ignored with this method.
Refer to the term "ZERO POINT RETURN METHOD".



CP CONTROL (Continuous Path Control)

Continuous path is a control method in which a path is followed without interrupting such as in uniform speed control.

CREEP SPEED

Speed at which the axis moves at slightly before the zero point when returning to the zero point during positioning.
It is difficult for the machine to stop accurately when running at high speed, so the movement must first be changed to the creep speed before stopping.
Refer to the term "NEAR-POINT DOG TYPE ZERO POINT RETURN".

CURRENT FEED VALUE

The number of pulses calculated to correspond to the movement distance output from the motion controller.

CURRENT LOOP MODE

Also called the torque loop mode.
Refer to "POSITIONING LOOP MODE".

CURRENT VALUE

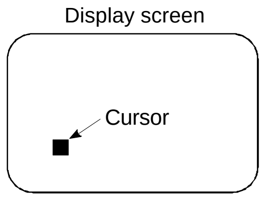
Current address in positioning control.

CURRENT VALUE CHANGE, CURRENT VALUE REWRITE

Refers to teaching the temporary approximate value as the current value for positioning when assembling the machine and connecting with the motion controller.
This function can also be used to write a temporary current value when the current value has been lost due to accidents, etc. If a zero point return is carried out after that, the motion controller will recognize the zero point.
The current value can be changed with the CHGA command while positioning is stopped.

CURSOR

This is the point on the display screen of a peripheral device, CRT, etc., which shows the operator where the next character will appear.



CUT

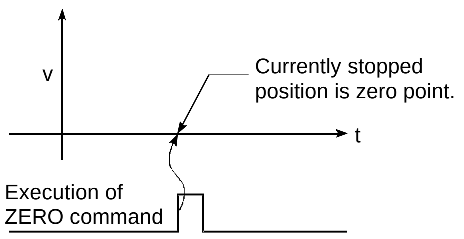
Cut refers to storing the parts on the Edit screen into the system buffer.
The parts stored in the system buffer with cut can be redisplayed on the Edit screen using paste.

CYCLOID CURVE

This is abbreviated as CY curve, and is known as a continuous curve. This curve has few acceleration frequency elements, and is suitable for high speeds. High characteristics values for speed, acceleration and inertia torque are a disadvantage.

DATA SET TYPE ZERO POINT RETURN

The currently stopped position is used as the zero point address.
A near-point dog switch is not required.
Refer to the term "ZERO POINT RETURN METHOD".

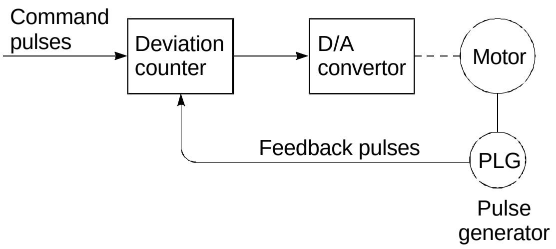


DELETE

Delete refers to deleting a part from the Edit screen.

DEVIATION COUNTER

Counter built into the drive unit for positioning. The feedback pulses are subtracted from the motion controller's command pulses, and the deviation value (droop pulses) of the commanded pulse sand feedback pulses are sent to the D/A converter to operate the motor. When there are no more command pulses, the motor is run until the droop pulses reaches 0.

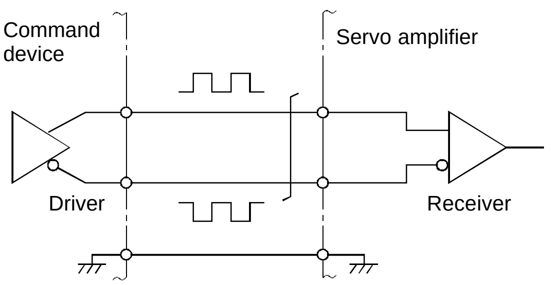


DIFFERENTIAL GEAR

One of the conveyance modules for the mechanism program in the virtual mode.
These gears make an auxiliary input according to the main shaft rotation.

DIFFERENTIAL OUTPUT

This is a type of encoder feedback pulse output.
When transmitting one signal, by transmitting a reversed-polarity signal as a pair, the receiver can make a judgment with the signal's logic.
This allows high-speed signals with a pulse train resistant to noise to be transmitted.



DIGITAL BUS CONNECTION

Generally, the commands output from the motion controller to the servo amplifier are pulse train or analog output commands.
However, a bus line can be connected to command with digital values. This allows a highly reliably, fast and high-accuracy system to be structured.

DIRECT CLUTCH

One of the mechanism programs for the virtual mode.
This is the conveyance module clutch, and is a clutch with zero setting time for which the smoothing time constant is not set.
Refer to the term "SMOOTHING CLUTCH".

DISCONTINUOUS CURVE

This refers to a uniform speed curve or uniform acceleration curve that does not have a continuous acceleration speed between the interval containing the start end and final end of a cam curve.

DOG SIGNAL

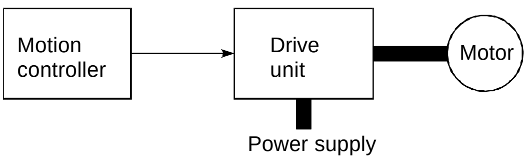
The near-point dog of the zero point return.

DRIVE MODULE

One of the mechanism programs for the virtual mode.
Refers to the virtual servomotor and synchronous encoder that rotate the main shaft and auxiliary input shaft.

DRIVE UNIT

The commands output from the motion controller are low-voltage, low-current commands with insufficient energy to run the motor. The drive unit increases the width of these commands so the motor can be run.

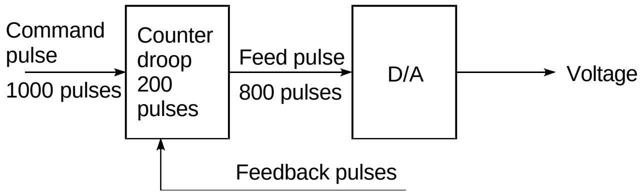


DRIVE UNIT READY

This signal is output when the drive unit for the motor is in a READY state.
This signal remains OFF when the drive unit power is OFF, or during faults, etc.

DROOP PULSE

Because of inertia (GD2) in the machine, it will lag behind and not be able to track if the positioning module speed commands are issued in their normal state.
Thus, for a servomotor, a method is used in which the speed command pulses are delayed by accumulation in a deviation counter. These accumulated pulses are called the droop pulse.
The deviation counter emits all pulses and returns to 0 when the machine stops.
In accurate terms, the difference of the feed pulse and feedback pulse is the droop pulse.



DWELL TIME

This is the time taken immediately after the positioning is completed to adjust for the droop pulses in the deviation counter. The positioning will not be accurate if this time is too short, so set a longer time as the dwell time.

DWELL

Dwell refers to the state that is temporarily stopped without the follower member's displacement changing as time elapses.

DWELL PERIOD

This is the rotation angle of the input axis when the output axis is in the dwell state. The sum with the assigned angle is 360°.

DYNAMIC BRAKE

When protection circuits operate due to power failures, emergency stops (EMG signal) etc., this function is used to short-circuit between servomotor terminals via a resistor, thermally consume the rotation energy, and cause a sudden stop without allowing coasting of the motor.
Braking power is generated by electromagnetic brakes only when running motors with which a large brake torque can be obtained. Because electromagnetic brakes have no holding power, they are used in combination with mechanical brakes to prevent dropping of the vertical axis.

EIA

Refers to the EIA codes (EIA Standards) punched into the paper punch paper to instruct machining to the NC unit.
In addition to NC language, ISO Codes (ISO Standards) and JIS Codes (JIS Standards) can be used.

EIA CODE

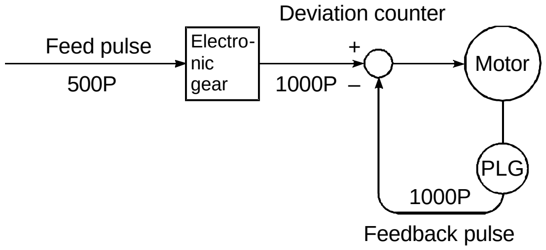
Tape code established by the Electronic Industries Association used for paper punching tape used with NC control machines.
Configured of eight tracks containing 6 bits indicating the information, an odd parity bit and EOB character (End Of Block).

ELECTROMAGNETIC BRAKE

This function is supplied on motors with electromagnetic brakes. Electromagnetic brakes are used to prevent slipping during power failures and faults when driving a vertical axis, or as a protective function when the machine is stopped.
These brakes are activated when not excited.

ELECTRONIC GEAR

Function that allows the feed amount per feed pulse to be freely changed during positioning.
The feed pulse and feedback pulse ratio, or P rate, is selected according to the machine, but this allows the setting to be made regardless of the machine system.



EMERGENCY STOP

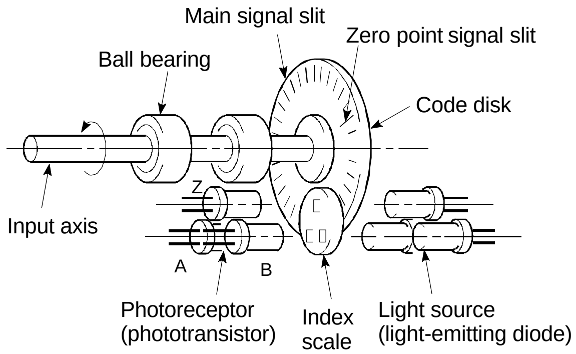
Emergency stop or a program to safety stop is placed in the PLC program. In addition, a circuit must be provided outside the PLC to ensure that the system stops.
This is provided because the emergency stop could be invalidated in the sequence program because of an unexpected PLC fault or because of the PLC power ON/OFF timing.
Disconnection and contact faults can be easily detected by using a b contact for the input device.
Using the EMG signal is recommended.

EMG SIGNAL

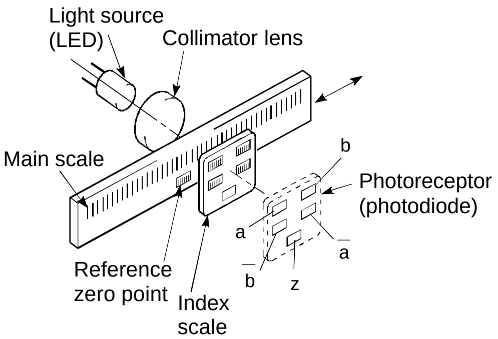
Emergency stop for all axes using a b contact for the external switch. This switch is always ON.
When operated, all axes will stop, the external emergency stop input flag (M9076) will turn OFF, and the motor will coast.
Caution is required during use as the address will deviate.

ENCODER

This refers to an encoding device, such as a pulse generator, that inputs the position information into the control unit.



Rotary encoder
(Incremental)



Linear encoder

A binary output method, including the incremental method and absolute method, is used.
Refer to the term "ABSOLUTE ENCODER".
Refer to the term "INCREMENTAL ENCODER".

ERROR COMPENSATION

If there is a dimension error in the machine, and that error is larger or smaller than 1m regardless of a 1m command being sent from the unit, the motion controller will compensate that error amount. For example, if the actual feed amount is smaller than 1m, extra pulses will be set to compensate the position to 1m.

EXTERNAL REGENERATIVE BRAKE RESISTOR

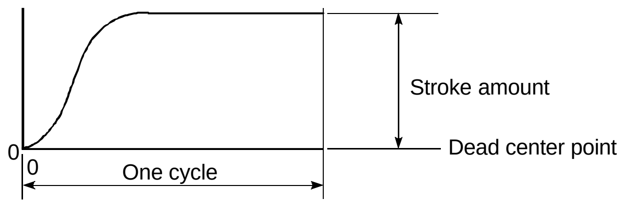
This is also called the regenerative brake.
When a machine is moved with a motor, power is normally supplied to the motor from an amplifier. However, the rotation energy in the motor and machine counterflows (regenerates) to the amplifier when the motor is decelerating or when driving a descending load.
The external regenerative resistor consumes this regeneration energy with resistance, obtains the regenerative brake torque, and enables the full capacity of the regeneration system during stopping.
It is used when carrying out highly repetitive acceleration/deceleration.

FEEDBACK PULSE

This is a pulse train returned to confirm that the machine moved according to the commands issued with automatic control. If the machine did not faithfully operate according to the commands, a correction command is issued. For example, if a command is issued for 10,000 pulses, and a feedback pulse of 10,000 pulses is returned, then the balance becomes 0 and it can be judged that the command was faithfully followed.
Refer to the term "DEVIATION COUNTER".

FEED CAM

Feed operation, equivalent to the stroke amount from the stroke lower limit position (bottom dead position), is continuously carried out, allowing conveyor feed and transfer machine feed.



FEED FORWARD CONTROL

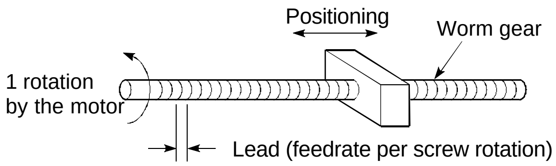
Used to reduce the motor delay and improve the servo tracking performance in respect to the position control command. (Invalid during automatic tuning.)
Set in the range of 0 to 150%.

FEED PULSE

Pulses issued to the servo unit or stepping motor from a command device such as a positioning unit. Also called the command pulse.

FEED SCREW

Machine that positions with the screw rotation, and functions as the basic screw. Ball screws are often used to reduce backlash and dimension error.



5TH POWER POLYNOMIAL CURVE

This curve with smooth and good characteristics has five boundary conditions.

FILE NAME

Name assigned when writing data or programs into the FD or HD.
Configured of the system name and machine name, using up to 8 characters. An index character is attached.
Refer to the term "MACHINE NAME".

FIXED-DIMENSION FEED

This is the feeding of a set dimension for cutting sheet and bar workpieces into the designated dimensions. Increment system positioning is often used. The current value is not incremented, even when the feed operation is repeated.
FEED-1, FEED-2 and FEED-3 are available.

FOLLOWER MEMBER

Generic term of the partner element (rod that moves back and forth) which contacts with the cam, or the load system following that element.

FORMATTING

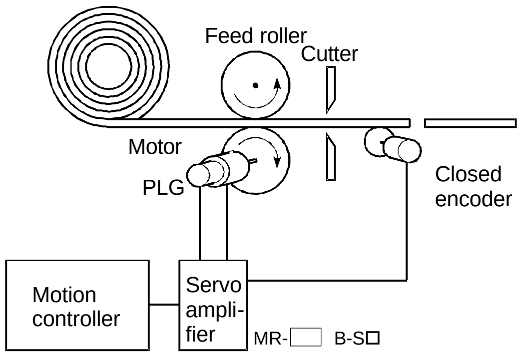
Refers to initializing the HD or FD disk.
Operation to write the personal computer rules and directly, etc., into the disk. The disk memory size will decrease according to the format.
The disk is for general-purpose use, so it must be formatted according to the personal computer. Formatting only needs to be carried out once.

FORWARD LIMIT SWITCH SIGNAL

This is the positioning control device input signal used to notify that the external upper limit switch (b contact configuration, normally ON) outside the movement range at which positioning control is carried out was activated.
The positioning operation stops when the external FLS signal (b contact) turns OFF (non-continuity status).

FULLY CLOSED CONTROL

The closed encoder is installed on the machine's movement mechanism to detect the direct movement distance and minimized the machine system error for the conveyance system mechanisms (gear, ball screw, timing belt, etc.) installed between the motor and machine.
Allows positioning control with a mechanism having slip to be optimized.



Using the closed encoder, the workpiece length can be directly detected, so the workpiece cutting length will be constant regardless of the feed roller slip.

The relation with GD2 is issued with 4gl using the gravitational acceleration as g.

GEAR

A conveyance module used in the mechanism program for the virtual mode, which branches the main shaft rotation to the output module. The gear ratio and rotation direction can be set.

GRID

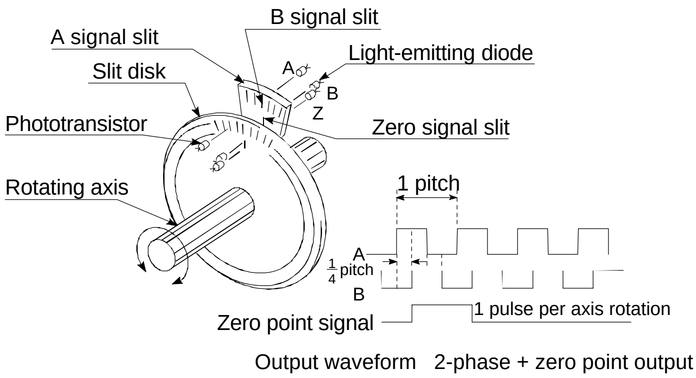
Reference horizontal lines and vertical lines handy for laying out parts on the MECHANISM EDIT screen.

HARMONIC MOTION

This is a type of cam curve.
With this movement, the displacement X can be expressed with the following expression for a right angle element in uniform speed circular movement:
$X = a^{\circ}\cos(\omega t + \Phi 0)$
Where, a is the vibration, ω is the amplitude, $(\omega t + \Phi 0)$ is the phase angle and $\Phi 0$ is the default phase angle

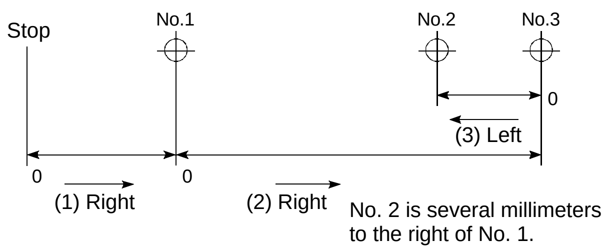
INCREMENTAL ENCODER

A device that simply outputs ON/OFF pulses by the rotation of the axis. 1-phase types output only A pulses, and do not indicate the axis rotation direction. 2-phase types output both A and B pulse trains, and can judge the rotation direction. The direction is judged to be forward if the B pulse train turns ON when A is ON, and judged to be reverse if A turns ON when B is ON. There is also another type of incremental encoder with a zero point signal. The most commonly used incremental encoders output between 100 and 10,000 pulses per axis rotation. Refer to "ENCODER".



INCREMENTAL MODE

With this method, the stop point during positioning is 0, and the position is indicated with the designated direction and distance. Also called the relative address system. This system is used in fixed-dimension feed, etc. Compare ABSOLUTE SYSTEM.



INERTIA

The property of an object, when not being affected by external forces, where it tries to maintain its current condition. The inertia moment.

IN POSITION

Signal that relies on the positioning data's servo parameters. The droop pulse amount in the deviation counter (difference of position feedback from position command value and servomotor) is detected, and if the result matches the setting value, this signal turns ON.
This can be used to disregard fractional droop pulses, and start the next positioning.

INSTALLATION FUNCTION

The OS (operating system) in the motion controller can be rewritten with a peripheral device. The OS includes SV13 for the transfer assembly machine, SVC22 for the automatic machine, SV43 for the machine tool peripheral device and SCV41 for the dedicated robot.
Usage that matches each machine can be realized by installing the OS.

INVERTER

This refers to a device to change a direct current (DC) to an alternating current (AC).
The device actually changes the motor speed by changing 50Hz or 60Hz of commercial frequency to direct current once, then changing it again to a 5 to 120Hz alternating current and controlling the motor speed.

JERK

The acceleration rate is differentiated by time to indicate the acceleration rate change rate.

JOG

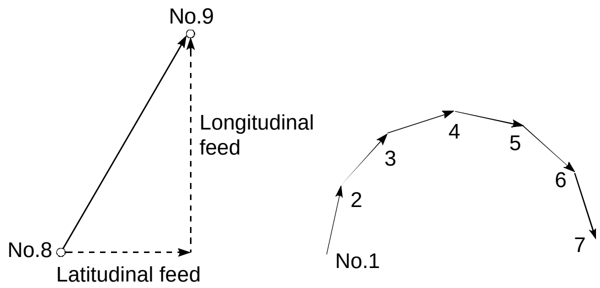
This refers to moving the tool in small steps at a time. Inching.
JOG operation can be carried out with test operation of a peripheral device, and from the sequence program by writing in the parameters and JOG speed.

kPPS

This is the abbreviation for "kilopulses per second". 80kPPS equals 80,000 pulses per second.

LINEAR INTERPOLATION

Automatic operation, in which the CPU makes calculations so that the axis moves along a straight line when positioning is carried out by simultaneously operating both the longitudinal feed and latitudinal feed motors.
The types include ABS-2, INC-2, ABS-4 and INC-4.
An example of 2-axis linear interpolation is shown on the right.



LINE MONITORING

Refers to monitoring the control status of the PLC and controller during operation.

LOAD INERTIA RATIO

GD_L^2/GD_M^2
Refer to "GD^2".

LOW-INERTIA MOTOR

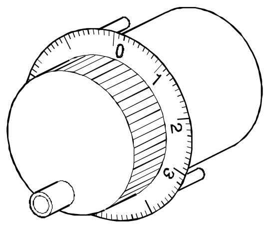
This is a motor used when frequent acceleration/deceleration is repeated. Low-inertia motors are longitudinally longer, to decrease the rotor diameter and cover the torque. This enables their inertia moment to be reduced up to 1/3 that of standard motors. The ideal load inertia ratio is 1 or less.

MACHINE NAME (System Name)

This is a sign of up to 8 characters that the user can choose and assign to a file name.
Alphabetic characters (uppercase), numerals, and minus (–) signs can be used. The first character is an alphabetic character.
Refer to the term "FILE NAME".

MANUAL PULSE GENERATOR

The handle of this device is manually rotated to generate pulses. This device is used when manually carrying out accurate positioning.



MARGIN

This refers to the cam and cam follower contact rate, and usually is 60% or more.

M CODE

Signal for auxiliary functions such as drill replacement, clamp tightening/loosening, welding electrode up/down and various displays carried out in sequence with positioning.
The user can assign (1: clamp, 2: loosen, etc.) the codes 1 to 255.
M is the abbreviation of M.

MASTER AXIS

When carrying out interpolation operations for positioning, this is the side on which the positioning data is executed in priority.
This refers to the interpolation control unit set in the parameter block.

MECHANISM PROGRAM

Refers to the program configured of the mechanism connection diagram containing drive modules (virtual servomotor, synchronous encoder), virtual main shaft, conveyance modules (gears, clutch, transmission, differential gears), output module (cam, roller, ball screw, rotary table), and the parameters for each module.

MECHANISM SUPPORT LANGUAGE

By using software to process the synchronous control that mechanically combines the mechanisms such as the conventional main shaft, gears and cam, etc., the positioning control is switched to control(roller output, ball screw output, rotary table output, cam output) by the servomotor.
Refer to the term "MECHANISM PROGRAM".

MODEL ADAPTIVE CONTROL

When operation is actually carried out, the status amount of the actual control will differ from the ideal control's status amount.
This enables optimum loop gain control based on this difference, and always carries out control with the best performance.

MODIFIED CONSTANT VELOCITY CURVE (Modified uniform speed)

This is abbreviated as MCV curve, and has a uniform speed interval at the center of the curve. This is used when the maximum speed needs to be dropped to reduce the pressure angle, or when a uniform speed section is required.
Suitable for medium-speed, heavy loads.

MODIFIED SINE CURVE

This is abbreviated as MS curve, and is a standard curve commonly used. This curve is often used when the maximum speed or cam axis torque coefficient is small, the acceleration is relatively small, and the load properties are unknown.
Suitable for high-speed, medium loads.

MODIFIED TRAPEZOID CURVE

This is abbreviated as MT curve, and is a standard curve developed to reduce the maximum acceleration value. This curve is suitable for high-speed light loads.

MONITORING TRACE GRAPH

This is a monitor function that traces (records) the values of the position command, position droop, monitor speed, motor current and speed command during positioning, and displays these as a waveform.

MOTION CONTROL

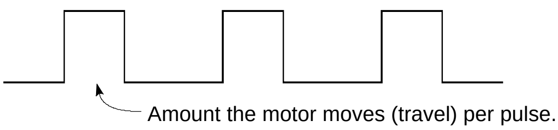
Refers to positioning control.

MOVEMENT AMOUNT PER PULSE

When using mm, inch, or angle units, the movement amount is calculated and output from the machine side showing how much the motor shaft moves per pulse. Equivalent to the positioning detection units. Positioning accuracy in smaller units is not possible.
On the motor side, the movement amount per axis rotation is normally designed as a reference, so it is calculated as follows.

Movement amount per pulse =

$$\frac{\text{P rate}}{\text{No. of pulses per encoder rotation}} \times \text{Movement amount per rotation}$$



MULTIPLICATION RATIO SETTING

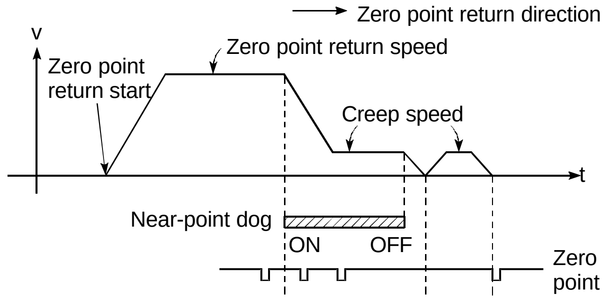
This refers to the P rate.
Refer to the term "P RATE".

NC LANGUAGE (Numerical Control Language)

This is the language punched into the paper tape that instructs the machining to the NC module.
The NC language consists of EIA codes (EIA language), ISO codes (ISO standards), and JIS codes (JIS standards).

NEAR-POINT DOG TYPE ZERO POINT RETURN

During zero point return, the axis starts to decelerate when the near-point dog turns ON. When the axis has moved to near-point dog OFF at the creep speed, the first zero point signal position is set as the zero point address. The length of the near-point dog is the point.
Refer to the term "ZERO POINT RETURN METHOD".



NO-DWELL MOTION

At the start end and final end of the operation, reciprocation is continued with the random acceleration value with no dwelling. This reduces the acceleration (A) value.

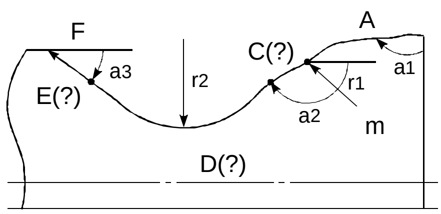
NOTCH FILTER

The notch frequency is set according to the machine system's resonance frequency.

Setting value	Notch frequency (Hz)
0	Not used
1	1125
2	750
3	562
4	450
5	375
6	321
7	281

NUMERICAL CONTROLLER

Advanced positioning carried out using a device called a numerical control module (NC module). This control can be used to carry out high-accuracy, high-speed control of 3 or more axes. It is possible to control movement for complex curved lines and curved surfaces.



ONE-DWELL MOTION, DWELL-RISE-DWELL MOTION

If the start end or final end of the stroke is stationary, and the same curve is returned for the rising stroke or lowering stroke, this can be used to reduce the acceleration and make the movement smooth.

OPTION SLOT

Slot for mounting motion unit or MELSEC-Q Series to match working purposes.

OUTPUT MODULE

Module that moves the servomotor in the virtual mode. Includes the roller, ball screw, rotary table and cam.

PANCAKE MOTOR

The axial dimension of this motor is approx. 100mm shorter than that of a standard motor. This type of motor is used when the servomotor installation space is narrow.

PARABOLIC CURVE

This is abbreviated as PB curve. This curve features the smallest dimensionless maximum acceleration, and enables short-time control under conditions that suppress the acceleration's maximum value.
Acceleration can be noncontinuous and vibration can occur easily.

PARAMETERS

This specifies the PLC functions. The user can set the memory size, relay or timer type, status latch selection and comment size, etc., in parameters. Default settings to set the basic functions are provided. Fixed parameters and servo parameters for positioning are also provided.

PARAMETER BLOCK

This is used to easily change the control conditions used for the positioning process, such as the acceleration/deceleration control data.

PASTE

Paste refers to redisplaying the parts cut from the Edit screen and stored in the system buffer.

PCPU

Positioning control CPU used in the motion controller's CPU configuration.
There is also a sequence control CPU called the SCPU.

PG0 (PG ZERO)

Refer to the term "ZERO POINT SIGNAL".

PLC READY

This signal indicates that the PLC CPU is ready. The special function unit's function can be used only in this state.

PLURAL HARMONIC MOTION

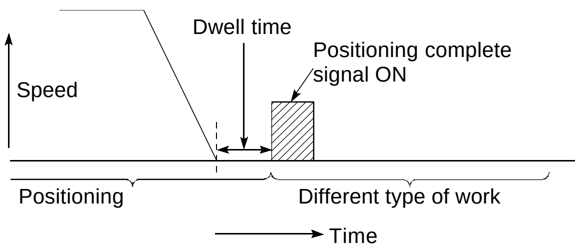
This is a type of cam curve.
With this movement, the acceleration pattern moves in the same manner as several right angle axis elements in uniform circular motion. "Single harmonic motion" has been improved so vibration does not occur as easily.

POSITIONING

Moving the machine from a point to the next determined point. For example, used to determine the length in a mm unit, or to find the hole drilling position.
Servomotor is used to drive the motion controller which issues the position command.

POSITIONING COMPLETION SIGNAL

Signal Xn1 which turns ON when the positioning dwell time is completed.
Used to start a separate work (clamping, etc.) after positioning with this signal.



POSITIONING DEVICES

This refers to the input/output signals, internal relays, data registers, special relays and special registers used to exchange signals between the SPCU (PLC CPU) and PCPU (positioning CPU).

POSITIONING PARAMETERS

This is basic data for carrying out positioning control. Types of data include the system settings set to match the servomotor and servo amplifier being used, the control unit, movement amount per pulse, speed limit value, upper and lower stroke limit values, and acceleration/deceleration time, etc.

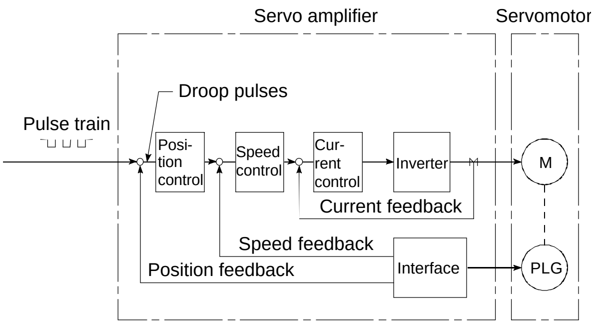
POSITION LOOP GAIN

This is an item in the positioning data's servo parameters, and indicates the speed of the control response during positioning control.
The No. of droop pulses in the deviation counter is specified during operation, and if the setting value is high, the droop pulses decrease allowing the settling time when stopping to be shortened. However, if too high, undershooting may occur during stopping, or there may be vibration when stopping. If the setting value is small, the droop pulses will increase, thereby causing the settling time when stopping to increase. This may allow smooth stopping, but will increase the stopping error.

$$\text{Position loop gain} = \frac{\text{Command pulse frequency}}{\text{Droop pulses}} \ (\text{sec}^{-1})$$

POSITION LOOP MODE

This is one servo control mode used in positioning. It is a mode for carrying out position control. The other servo control modes are the speed loop mode for carrying out speed control, and the torque loop mode for carrying out torque control (current control).

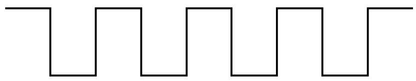


PTP Control (Point To Point Control)

This is a type of positioning control. With this control method, the points to be passed are designated at random locations on the path. Movement only to a given target positioning is requested. Path control is not required during movement from a given point to the next value.

PULSE

Refers to the current (voltage) turning ON and OFF within a short time. Similar to the human's pulse. A pulse train is a sequence of pulses.
The MELSEC AD71 is a unit that generates pulses.
AD61 is a unit that receives and counts the pulses.



PULSE COMMAND

Command that turns ON only one cycle (1 scan) of the program when the condition is ON. With the MELSEC-A, one scan time is the rising edge of when the signal turns ON. Commands include the PLS command that turns ON, and the PLF command that turns one scan time ON at the falling edge of the OFF command.

PULSE GENERATOR

This is a device that generates pulses. Examples include devices installed on the motor shaft that create pulses when the shaft rotates, and digital devices.
1-phase types output one pulse train. 2-phase types output two pulse trains with a phase difference. From 600 to 1,000,000 pulses can be output per shaft rotation. Generators with a ZERO POINT signal function to output 1 or 2 pulses per shaft rotation. Abbreviated as PLG.
Refer to the term "ENCODER".

PULSE RATE (P RATE)

Coefficient that doubles, triples, halves or thirds the feedback pulses per motor axis rotation during positioning.
Ratio of the feed pulse and feedback pulse.
For example, if the P rate is set to 2 when the pulses per rotation are 2400, this will be equivalent to 1200 pulses. The axis rotation per pulse for 2400 pulses is 0.15°, but with 1200 pulses, this becomes 0.3°. The positioning accuracy drops as the P rate increases.
Refer to the term "ELECTRONIC GEAR".

PULSE TRAIN COMMAND

Positioning control proportional to the number of pulses can be carried out by continuously outputting the number of pulses corresponding to the machine's movement distance from the motion controller to the servomotor's servo amplifier.

READY (M9074)

State in which the PCPU or servo amplifier power has been turned ON and is in the normal operation state.

REVERSE LIMIT SWITCH

This is the positioning control device input signal used to notify that the external lower limit switch (b contact configuration, normally ON) outside the movement range at which positioning control is carried out was activated.
The positioning operation stops when the external RLS signal (b contact) turns OFF (non-continuity status).

REAL MODE

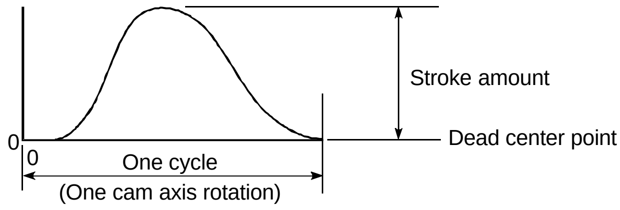
Refers to directly controlling the servomotor with the servo program.

REAL-TIME AUTO TUNING (Real-time Automatic Tuning)

Refer to "AUTO TUNING".

RECIPROCATING CAM

Reciprocating operation, equivalent to the stroke from the stroke lower limit position (bottom dead position), is continuously carried out, allowing extrusion/return operations, vertical operations, and left/right operations.

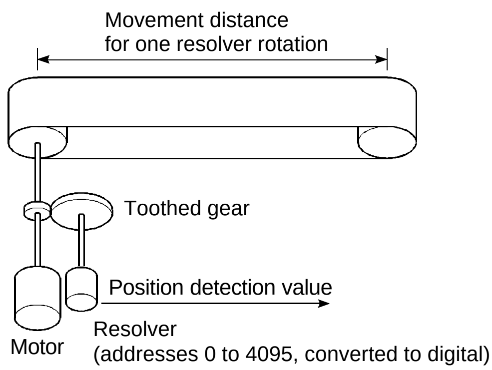


REGENERATIVE BRAKE OPTION

This function is an option. It is used when carrying out highly repetitive acceleration/deceleration.
Refer to "EXTERNAL REGENERATIVE RESISTOR".

RESOLVER

This is also called a 2-phase synchronizer. Compared to the 1-phase voltage input, one rotation of the axis rotation angle is converted into a right angle 2-phase voltage (analog) and output.



ROLLER

A cylindrical rotating item that feeds and rolls paper or steel plates, etc.
The roller output can be set as a virtual mode output module.

ROTARY TABLE

A round table on which the workpiece is placed. Positioning control is carried out while rotating the workpiece in a 360° range.

SCPU

Sequence CPU used in the motion controller CPU configuration.
There is also a positioning control CPU called the PCPU.

SCROLL

Refers to continuously moving up/down the CRT screen. The screen changes according to the movement of the machine being controlled or according to the keys being operated.

SEQUENCE CONTROL

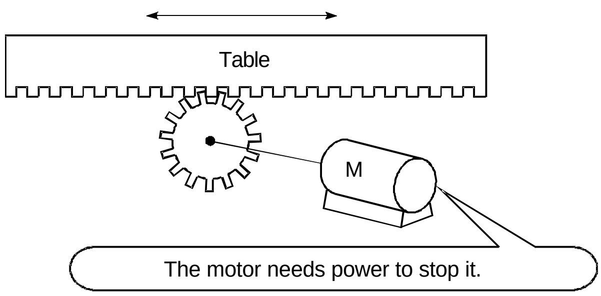
This means a sequence program in which the completion of one of a series of operations is detected by a switch, and operations such as the start of the next operation by that signal are carried out in order and controlled.

SERVO AMPLIFIER

The amplifier includes the type built-into the controller base and the external type. This amplifier outputs the rotation command to the servomotor, receives the feedback pulses and controls the servomotor.

SERVO LOCK

In positioning using a servomotor, stepping motor, etc., working power is required to hold the machine at the stop position.
(The position will be lost if the machine is moved by external power.)
This kind of state is called servo lock or servo lock torque.

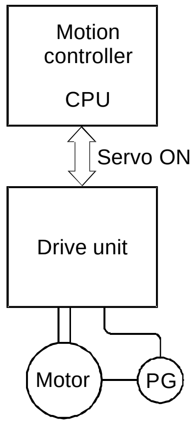


SERVOMOTOR

A motor that rotates true to the command. Servomotors are highly responsive, and can carry out frequent high-speed and high-accuracy starts and stops.
DC and AC types are available, as well as large-capacity motors. A pulse generator accessory for speed detection is common, and feedback control is often carried out.
In other words, this motor moves according to the command values while detecting the current value, and minimizes the difference between the command value and current value.

SERVO ON

Positioning cannot be started unless the drive unit is normal and servo ON is set to ON.



SERVO PARAMETERS

Refer to the term "POSITIONING PARAMETERS".

SERVO PROGRAM

Program to control the servo program.
The commands include independent linear control, linear interpolation control, circular interpolation control, fixed-dimension feed, speed control, uniform speed control and zero point return, etc.

SERVO RESPONSE

Set the responsiveness for automatic tuning. Optimum response corresponding to the machine's rigidity can be selected. The higher the machine's rigidity is, the higher the responsiveness can be set. This allows the tracking of the command to be improved, and the settling time to be shortened.

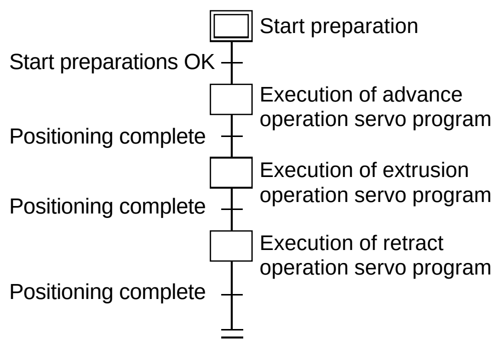
Machine type	Setting value	Details: Responsive-ness	Details: Guideline to applicable machine's rigidity	Details: Load inertia target GD_L^2/GD_M^2	Position settling time guideline = GD_L^2/GD_M^2 within 5-fold
Normal	1	Low response	Low rigidity	1 to 10-fold	5 to 300ms
	2		to		
	3	Medium response	Medium rigidity		1 to 70ms
	4		to		
	5	High response	High rigidity		1 to 30ms
Large friction	8	Low response	Low rigidity		70 to 400ms
	9		to		
	A	Medium response	Medium rigidity		10 to 100ms
	B		to		
	C	High response	High rigidity		10 to 50ms

SETTING TIME

This refers to the delay time from the time stop command is completed to when the servomotor stops (time for droop pulse to increment by one).

SFC (Sequential Function Chart)

A sequential function chart is a programming method optimally structured for running a machine's automatic control in sequence with the PLC.



SIMPLE HARMONIC MOTION

This is a type of cam curve.
With this movement, the acceleration pattern moves in the same manner as a right angle axis element in uniform circular motion.
This curve has smooth and good characteristics so is suitable for low-speed applications.
Acceleration can be noncontinuous and vibration can occur easily.

SIMULTANEOUS START CONTROL

A START command that simultaneously executes two to three types of servo programs, and starts several servomotors simultaneously.
Multiple axes designated in the special registers for JOG operation are simultaneously started by the special relay.

SKIP FUNCTION

Function that allows the next positioning to be started if the external STOP signal turns ON during positioning control or if the external STOP signal remains ON while stopped.
If the external STOP signal input invalid flag is turned ON during deceleration and the start accept flag is turned OFF, the next positioning will start with the SVST command.

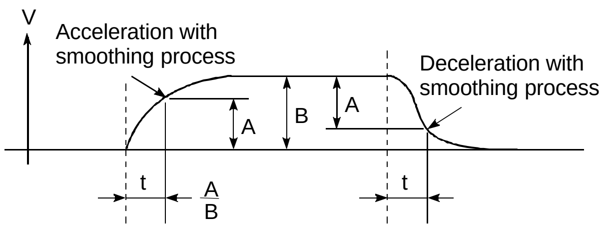
SLAVE AXIS

Refer to the term "MASTER AXIS".

SMOOTHING CLUTCH

Clutch used as a conveyance module in the virtual mode. The smoothing time constant is set for this clutch.
The rotation operation can be smoothly conveyed when the clutch is ON and OFF.
The direct clutch refers to when the smoothing time constant is set to zero.

SMOOTHING TIME CONSTANT t



Refer to the term "SMOOTHING CLUTCH".

SPEED CHANGE

Refer to the term "DSFLP COMMAND".

SPEED CHANGE GEAR

This is a transmission module in the mechanism program for the virtual mode. The main shaft's rotation speed is changed and conveyed to the roller output module.

SPEED CHANGEOVER CONTROL

With this control, positioning is carried out to the end point of the movement amount while changing the speed at the speed changeover point during positioning control.

SPEED CONTROL

This control is suitable for endless rotation in a single direction, such as for a conveyor. Commands include the VF forward run, VR reverse run commands (position loop) and VVF forward run, VVR reverse run commands (speed loop). The current feed value is cleared to zero simultaneously with starting. Rotation takes place at the preset speed, the current feed value is not incremented/decremented, and the axis decelerates to a stop when the stop command is input.
The upper and lower limit values for the stroke limit are ignored.

SPEED INTEGRAL COMPENSATION

This is one item in the servo parameters of the positioning data. It is used to raise the frequency response during speed control, and improve transient characteristics.
When adjusting the speed loop gain, raising this value is effective if the overshooting during acceleration/deceleration remains large.
This compensation is set in ms units.

SPEED LIMIT VALUE

This is the max. speed for positioning. Even if other data is mistakenly set to a higher speed than this, the positioning will be carried out at this speed limit value when it is set in the parameters. The acceleration time becomes the time to accelerate from a stopped state to the speed limit value, and the deceleration time becomes the time to decelerate from the speed limit value to a stopped state.

SPEED LOOP GAIN

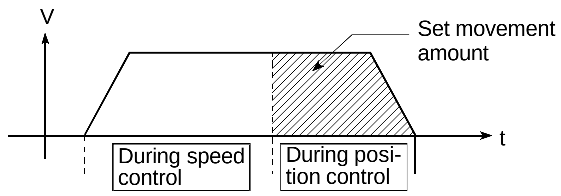
This is one item in the servo parameters of the positioning data. It expresses the speed of the control response during speed control. When the load inertia moment ratio increases, the control system speed response decreases and the operation may become unstable. If this happens, the operation can be improved by raising this setting value.
The overshoot will become larger if the speed loop gain is raised too far, and motor vibration noise will occur during operation and when stopped.

SPEED LOOP MODE

Refer to the term "POSITION LOOP MODE".

SPEED/POSITION CONTROL

Incremental positioning control is carried out when a changeover signal is input from an external device during speed control.

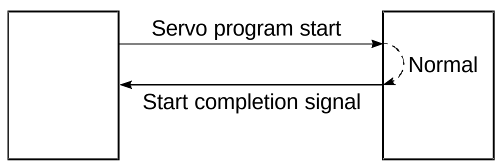


SSCNET

Abbreviation of Servo System Controller Network.
Connection method that improves the reliability through high-speed serial communication between the motion controller and servo amplifier.
Wiring work has been simplified with a one-touch connector.

START COMPLETE

This signal gives an immediate response notifying the user that the motion controller that was started is now in a normal state and can start positioning.



STARTING AXIS

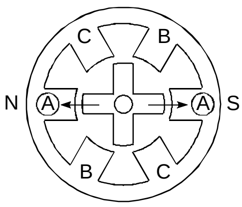
This is the axis to be started, and refers to axis 1 to axis 8/32.

STATUS

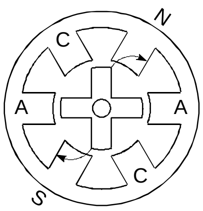
Device that indicates the status. Generic name of signal that turns ON (turns to 1) with the clutch status, virtual mode status or zero point return request, etc.

STEPPING MOTOR

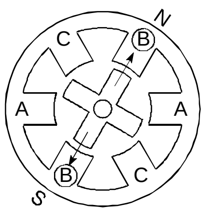
A motor that rotates a given angle (example: 0.15°) when 1 pulse is generated. For that reason, a rotation proportional to the No. of pulses can be obtained. 2-phase to 5-phase stepping motors are available. In the 3-phase type, the rotor rotates in order from A to C when a voltage is applied. Often found in compact motors, stepping motors rotate accurately without feedback. Be careful of step out when rotation is not accurate.



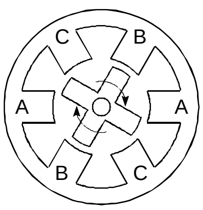
1) First, the A phase is excited by a pulse.



2) When the B phase is then excited, the force works in the direction shown by the arrows.



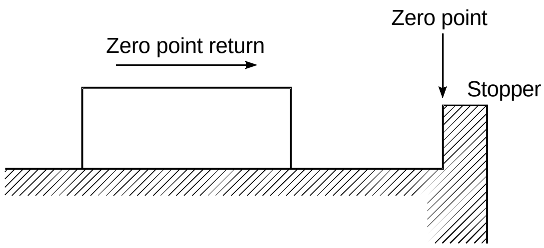
3) The nearest tooth to the B phase is attracted, and the rotation stops.



4) As the excitation phase is continuously changed, the rotor rotates in a clockwise direction.

STOPPER-FORCED STOP

Method of zero point return during positioning, which places a stopper at the zero on and presses against the stopper to stop.
If the axis is kept pressed against the stopper, the motor could burn or the stopper could be damaged. Thus, provide a timer and turn the motor OFF after a set time, or provide means to detect a sudden increase in the motor torque when pressing and turn the motor OFF, etc.

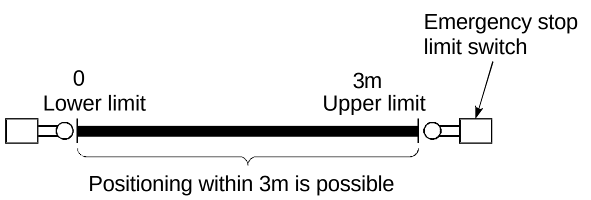


STROKE

The stroke is the variation in the operation by the distance from a stopped state to the next stopped state after a movement.

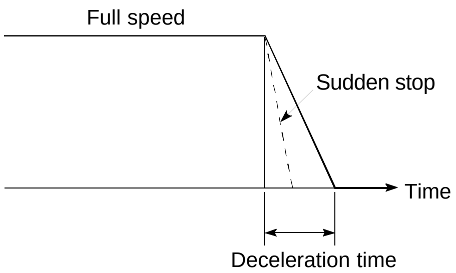
STROKE LIMIT

This is the range in which a positioning operation is possible, or the range in which the machine can be moved without damage occurring. For operations using a worm gear, the stroke limit is determined by the length of the screw. For operations using a fixed-dimension feed, it is determined by the max. dimension to be cut.
The upper and lower limits are set with the parameters. However, for safety purposes, a separate limit switch wired to the external signal input unit should be provided so that the operation can be stopped automatically.



SUDDEN STOP

A stop carried out in a shorter time than the deceleration time designated in the parameters.



STOP SIGNAL

In positioning control, this is the positioning control device input signal that directly stops the movement from an external source.
The operation stops when the external STOP signal (a contact) turns ON (continuity), and XnD turns ON.

SV12

Special specifications for the motion controller OS. Compared to SV13, linear interpolation and uniform speed control can be commanded for 1 to 8 axes.

SV13

Motion controller OS prepared for transfer and assembly. 1 to 4-axis linear interpolation, 2-axis circular interpolation, CPU control (uniform speed control) and speed control, etc., can be executed, making this suitable for applications such as transfer machines and assembly machines.
Order control is possible with SFC.

SV22

Motion controller OS prepared for automatic machines. Multiple servomotors can be synchronously controlled, and the cam can be controlled with the software, making this suitable for applications such as automatic machines.

SV43

Motion controller OS prepared for machine tool peripheral devices. Linear interpolation, 2-axis circular interpolation, CP control (uniform speed control) and speed control, etc., can be carried out with NC language (EIA), making this suitable applications such as machine tools.

SV51

Motion controller OS prepared for dedicated robots. 3D linear interpolation, 3D circular interpolation and 3D CP control, etc., can be carried out, making this suitable for dedicated robots (painting machines, transfer machines, etc.).

SYNCHRONIZED CONTROL

The virtual mode drive module that rotates the main shaft while synchronizing several output modules (servomotors) via the conveyance module.

SYNCHRONOUS ENCODER

One of the virtual mode drive modules which inputs the encoder pulses for the external machine and operates the output module in synchronization with these pulses.

TEACHING

Function for the operator to indicate and teach the position during positioning, when the address is unclear, or when the point is to be matched to the actual part.
For example, it is bothersome to write in all of the address for points in a complicated shape such as a drawing, so if the points are taught while following a model, positioning can be realized later.

TEACHING PLAYBACK FUNCTION

Using the teaching unit (A30TU/A31UT), the machine can be actually moved to set the positioning points with the address teach function, and to simultaneously create the servo program and set the positioning points with the program teach function.

TEACHING UNIT

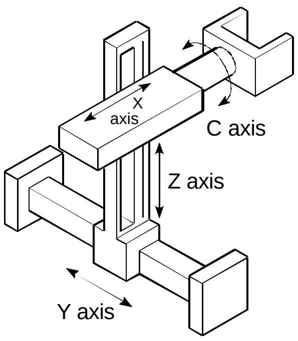
Device that allows teaching such as writing/reading data, operation and monitor during positioning.
The A30TU/A31TU type teaching unit is available.

THREE-DIMENSIONAL CAM

This cam uses three-dimensional movements allowing the generally occupied space to be reduced compared to the flat cam, and also allows a positive motion locked cam to be realized.

THREE-DIMENSIONAL INTERPOLATION CP CONTROL

This machine control servo command [CPSTART XYZ command] is used with the SV51 dedicated robot, and controls the XYZ axes (3-axis orthogonal) and C axis (1-axis rotation) at a uniform speed.



TOP DEAD CENTER

Refers to the upper side of the machine installation path during reciprocating operation of the cam mechanism.

TORQUE

Torque is obtained by multiplying the size of the force applied on the axis by the arm length to the function line of that force. The unit is N.m (kgf.m).

TORQUE LOOP MODE

Also called the current loop mode.
Refer to "POSITIONING LOOP MODE".

TORQUE RIPPLE

Torque width variations, deviations in the torque.

TRACKING

In this function, positioning is carried out at a speed relative to a moving target object by inputting the movement amount from an external encoder and adding it to the servo command value.

TRANSMISSION MODULE

One of the transmission modules for the mechanism program in the virtual mode.
Refers to a gear, clutch, speed change gear, or differential gear that conveys the drive module's rotation to the output module.

TRAPECLOID CURVE

This is abbreviated as TRP curve. This curve can suppress the residual vibration after input has stop, and has high shake-proof properties.

TRAVEL

Refer to the term "STROKE".

TWO-DWELL MOTION

Motion with dwelling at both ends of the stroke.

UNIT SETTING

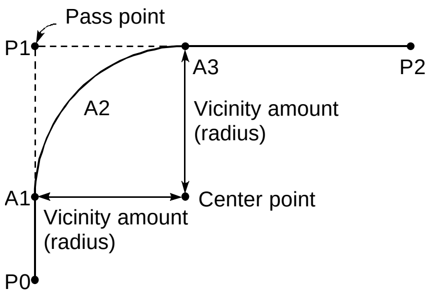
Refers to changing the actual address unit or movement amount unit to be positioned to.
Use mm, inch, degree or pulse unit.

UNSYMMETRICAL

With this curve, the proportion of the first and latter acceleration ranges is different, so it is manually used to improve the deceleration range characteristics for high-speed specifications.

VICINITY PASSAGE

This allows the pass points to be moved smoothly during 3D interpolation CP control of the SV51 dedicated robot.



When the vicinity passage is zero:
The path will follow P0 → P1 → P2.
When vicinity passage is designated:
The path will follow P0 → A1 → A2 → A3 → P2.

VIRTUAL AUXILIARY INPUT

One of the mechanism programs for the virtual mode.
Incremental or decremental rotation is applied from the auxiliary axis' virtual servomotor or synchronous encoder.

VIRTUAL MAIN SHAFT

One of the mechanism programs for the virtual mode.
The drive module's rotation is coupled to the conveyance module's gears with this shaft.

VIRTUAL MODE

A method of operating the mechanism program drive modules with the servo program or external encoder, and driving the servomotor.
Conversely, the method of directly driving the servomotor with a servo program is called the real mode.
Refer to the term "MECHANISM PROGRAM".

VIRTUAL MODE STATUS

Special relay M2044, a monitor that allows the virtual mode to be confirmed.

VIRTUAL SERVOMOTOR

A drive module used in the mechanism program for the virtual mode, which is started by the servo program. The main shaft is coupled to the virtual servomotor.

Vm VELOCITY

Refers to the cam's maximum dimensionless speed.
Refer to the term "V".

V VELOCITY

Refers to the cam's dimensionless speed. This is the dimensions displacement (movement displacement expressed as 0 to 1 from start to stop) differentiated by the dimensionless time (operation time expressed as 0 to 1 from start to stop).
Refer to the term "Vm".

WDT ERROR (Watchdog Timer Error)

This is read as a watchdog timer error. It means that an error occurred in the PCPU. M9073 turns ON when an error occurs.

WINDOW

This refers to the selection screen displayed on the SW6RN-GSV22P or CAMP screen by the peripheral device.

- Menu selection window
- Mode function selection window
- Sub-function selection window
- Execution/setting selection window

WORD

This refers to a data unit. With the MELSEC-A Series, one word has 16 bits, and the values handled are -32,768 to 32767 as a decimal or 0 to FFFF as a hexadecimal.
Note that there are 32-bit commands. In this case, one word will have 32 bits, and the values will be -2,147,483,648 to 2,147,483,647. As a hexadecimal, the values will be 0 to FFFFFFFF.

WORD DEVICES

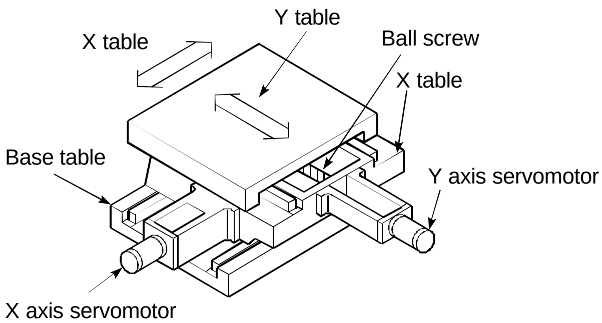
This is a device used in the PLC, and is an element having data. One point is a device configured of one word. Word devices include the timer (T), counter (C) and various registers (D, R, W, Z, V, A), etc.

X AXIS

Two-dimensional left/right horizontal direction.

XY TABLE

Device that can move the table in the X (horizontal direction) and Y (vertical direction) to simplify positioning work.
Used to drill holes in the plate, or to draw figures.



Y AXIS

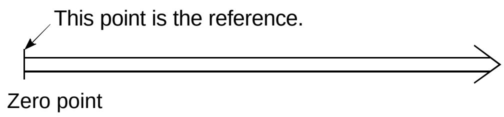
Two-dimensional forward/backward direction.

Z AXIS

Three-dimensional up/down movement.

ZERO POINT

This is the reference position for positioning. Positioning cannot start without a reference point.



ZERO POINT RETURN DATA

Data required by the motion controller to return to the zero point. This value is determined by the machine design, and requires the machine design to be changed to change to value later. This point is the reference for zero point positioning, so if a power failure occurs during positioning, or if the power is turned OFF and the axis is moved manually, the current value in the motion controller will deviate. In this case, carry out zero point return. When zero point return is started, it will search for the near-point dog, regardless of the current value, and will change to the creep speed.

ZERO POINT RETURN METHOD

The zero-point dog type, count type or data set type can be used.

ZERO POINT RETURN REQUEST

This signal turns ON in the following cases when using the increment method.

1) When the power is turned ON.
2) When the PLC READY signal turns OFF.
3) When the parameters or zero point return data are written from the peripheral device.
4) When the following is selected during the peripheral device's test mode
   Zero point return
   Positioning
   JOG operation
   Manual pulse generator

   Whether to carry out zero point return in these cases is up to the user.

ZERO POINT SIGNAL

1 pulse (or 2 pulses) generated per axis rotation of the pulse generator.
This signal is used in the positioning zero point return. Also called a Z phase, Z signal or PG0.
Refer to "PULSE GENERATOR".

Z PHASE

Also called PG zero.
Refer to the term "ZERO POINT SIGNAL".