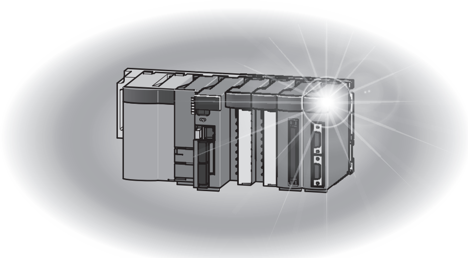
MITSUBISHI ELECTRIC

Motion Controller

MELSEC Q series

Q173D(S)CPU/Q172D(S)CPU Motion Controller (SV13/SV22) Programming Manual (Motion SFC)

-Q172DCPU
-Q173DCPU
-Q172DCPU-S1
-Q173DCPU-S1
-Q172DSCPU
-Q173DSCPU



● SAFETY PRECAUTIONS ●

(Please read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.
These precautions apply only to this product. Refer to the Q173D(S)CPU/Q172D(S)CPU Users manual for a description of the Motion controller safety precautions.
In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".

⚠ DANGER	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠ CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Depending on circumstances, procedures indicated by ⚠ CAUTION may also be linked to serious results.
In any case, it is important to follow the directions for usage.

Please save this manual to make it accessible when required and always forward it to the end user.

For Safe Operations

1. Prevention of electric shocks

⚠DANGER
● Never open the front case or terminal covers while the power is ON or the unit is running, as this may lead to electric shocks. ● Never run the unit with the front case or terminal cover removed. The high voltage terminal and charged sections will be exposed and may lead to electric shocks. ● Never open the front case or terminal cover at times other than wiring work or periodic inspections even if the power is OFF. The insides of the Motion controller and servo amplifier are charged and may lead to electric shocks. ● Completely turn off the externally supplied power used in the system before mounting or removing the module, performing wiring work, or inspections. Failing to do so may lead to electric shocks. ● When performing wiring work or inspections, turn the power OFF, wait at least ten minutes, and then check the voltage with a tester, etc. Failing to do so may lead to electric shocks. ● Be sure to ground the Motion controller, servo amplifier and servomotor. (Ground resistance : 100 Ω or less) Do not ground commonly with other devices. ● The wiring work and inspections must be done by a qualified technician. ● Wire the units after installing the Motion controller, servo amplifier and servomotor. Failing to do so may lead to electric shocks or damage. ● Never operate the switches with wet hands, as this may lead to electric shocks. ● Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to electric shocks. ● Do not touch the Motion controller, servo amplifier or servomotor terminal blocks while the power is ON, as this may lead to electric shocks. ● Do not touch the built-in power supply, built-in grounding or signal wires of the Motion controller and servo amplifier, as this may lead to electric shocks.

2. For fire prevention

⚠CAUTION
● Install the Motion controller, servo amplifier, servomotor and regenerative resistor on incombustible. Installing them directly or close to combustibles will lead to fire. ● If a fault occurs in the Motion controller or servo amplifier, shut the power OFF at the servo amplifier’s power source. If a large current continues to flow, fire may occur. ● When using a regenerative resistor, shut the power OFF with an error signal. The regenerative resistor may abnormally overheat due to a fault in the regenerative transistor, etc., and may lead to fire. ● Always take heat measures such as flame proofing for the inside of the control panel where the servo amplifier or regenerative resistor is installed and for the wires used. Failing to do so may lead to fire. ● Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to fire.

3. For injury prevention

⚠CAUTION

- Do not apply a voltage other than that specified in the instruction manual on any terminal. Doing so may lead to destruction or damage.
- Do not mistake the terminal connections, as this may lead to destruction or damage.
- Do not mistake the polarity (+ / -), as this may lead to destruction or damage.
- Do not touch the heat radiating fins of controller or servo amplifier, regenerative resistor and servomotor, etc., while the power is ON and for a short time after the power is turned OFF. In this timing, these parts become very hot and may lead to burns.
- Always turn the power OFF before touching the servomotor shaft or coupled machines, as these parts may lead to injuries.
- Do not go near the machine during test operations or during operations such as teaching. Doing so may lead to injuries.

4. Various precautions

Strictly observe the following precautions.

Mistaken handling of the unit may lead to faults, injuries or electric shocks.

(1) System structure

⚠CAUTION

- Always install a leakage breaker on the Motion controller and servo amplifier power source.
- If installation of an electromagnetic contactor for power shut off during an error, etc., is specified in the instruction manual for the servo amplifier, etc., always install the electromagnetic contactor.
- Install the emergency stop circuit externally so that the operation can be stopped immediately and the power shut off.
- Use the Motion controller, servo amplifier, servomotor and regenerative resistor with the correct combinations listed in the instruction manual. Other combinations may lead to fire or faults.
- Use the Motion controller, base unit and motion module with the correct combinations listed in the instruction manual. Other combinations may lead to faults.
- If safety standards (ex., robot safety rules, etc.,) apply to the system using the Motion controller, servo amplifier and servomotor, make sure that the safety standards are satisfied.
- Construct a safety circuit externally of the Motion controller or servo amplifier if the abnormal operation of the Motion controller or servo amplifier differ from the safety directive operation in the system.
- In systems where coasting of the servomotor will be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use dynamic brakes.
- Make sure that the system considers the coasting amount even when using dynamic brakes.
- In systems where perpendicular shaft dropping may be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use both dynamic brakes and electromagnetic brakes.

⚠CAUTION

- The dynamic brakes must be used only on errors that cause the forced stop, emergency stop, or servo OFF. These brakes must not be used for normal braking.
- The brakes (electromagnetic brakes) assembled into the servomotor are for holding applications, and must not be used for normal braking.
- The system must have a mechanical allowance so that the machine itself can stop even if the stroke limits switch is passed through at the max. speed.
- Use wires and cables that have a wire diameter, heat resistance and bending resistance compatible with the system.
- Use wires and cables within the length of the range described in the instruction manual.
- The ratings and characteristics of the parts (other than Motion controller, servo amplifier and servomotor) used in a system must be compatible with the Motion controller, servo amplifier and servomotor.
- Install a cover on the shaft so that the rotary parts of the servomotor are not touched during operation.
- There may be some cases where holding by the electromagnetic brakes is not possible due to the life or mechanical structure (when the ball screw and servomotor are connected with a timing belt, etc.). Install a stopping device to ensure safety on the machine side.
- To maintain the safety of the programmable controller system against unauthorized access from external devices via the network, take appropriate measures. To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.

(2) Parameter settings and programming

⚠CAUTION

- Set the parameter values to those that are compatible with the Motion controller, servo amplifier, servomotor and regenerative resistor model and the system application. The protective functions may not function if the settings are incorrect.
- The regenerative resistor model and capacity parameters must be set to values that conform to the operation mode, servo amplifier and servo power supply module. The protective functions may not function if the settings are incorrect.
- Set the mechanical brake output and dynamic brake output validity parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Set the stroke limit input validity parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servomotor encoder type (increment, absolute position type, etc.) parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servomotor capacity and type (standard, low-inertia, flat, etc.) parameter to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.

⚠CAUTION

- Set the servo amplifier capacity and type parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Use the program commands for the program with the conditions specified in the instruction manual.
- Set the sequence function program capacity setting, device capacity, latch validity range, I/O assignment setting, and validity of continuous operation during error detection to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Some devices used in the program have fixed applications, so use these with the conditions specified in the instruction manual.
- The input devices and data registers assigned to the link will hold the data previous to when communication is terminated by an error, etc. Thus, an error correspondence interlock program specified in the instruction manual must be used.
- Use the interlock program specified in the intelligent function module's instruction manual for the program corresponding to the intelligent function module.

(3) Transportation and installation

⚠CAUTION

- Transport the product with the correct method according to the mass.
- Use the servomotor suspension bolts only for the transportation of the servomotor. Do not transport the servomotor with machine installed on it.
- Do not stack products past the limit.
- When transporting the Motion controller or servo amplifier, never hold the connected wires or cables.
- When transporting the servomotor, never hold the cables, shaft or detector.
- When transporting the Motion controller or servo amplifier, never hold the front case as it may fall off.
- When transporting, installing or removing the Motion controller or servo amplifier, never hold the edges.
- Install the unit according to the instruction manual in a place where the mass can be withstood.
- Do not get on or place heavy objects on the product.
- Always observe the installation direction.
- Keep the designated clearance between the Motion controller or servo amplifier and control panel inner surface or the Motion controller and servo amplifier, Motion controller or servo amplifier and other devices.
- Do not install or operate Motion controller, servo amplifiers or servomotors that are damaged or that have missing parts.
- Do not block the intake/outtake ports of the Motion controller, servo amplifier and servomotor with cooling fan.
- Do not allow conductive matter such as screw or cutting chips or combustible matter such as oil enter the Motion controller, servo amplifier or servomotor.

⚠CAUTION

- The Motion controller, servo amplifier and servomotor are precision machines, so do not drop or apply strong impacts on them.
- Securely fix the Motion controller, servo amplifier and servomotor to the machine according to the instruction manual. If the fixing is insufficient, these may come off during operation.
- Always install the servomotor with reduction gears in the designated direction. Failing to do so may lead to oil leaks.
- Store and use the unit in the following environmental conditions.

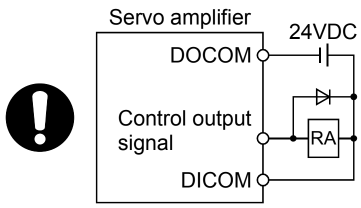
Environment	Conditions	
	Motion controller/Servo amplifier	Servomotor
Ambient temperature	According to each instruction manual.	0°C to +40°C (With no freezing) (32°F to +104°F)
Ambient humidity	According to each instruction manual.	80% RH or less (With no dew condensation)
Storage temperature	According to each instruction manual.	-20°C to +65°C (-4°F to +149°F)
Atmosphere	Indoors (where not subject to direct sunlight). No corrosive gases, flammable gases, oil mist or dust must exist	
Altitude	According to each instruction manual	
Vibration	According to each instruction manual	

- When coupling with the synchronous encoder or servomotor shaft end, do not apply impact such as by hitting with a hammer. Doing so may lead to detector damage.
- Do not apply a load larger than the tolerable load onto the synchronous encoder and servomotor shaft. Doing so may lead to shaft breakage.
- When not using the module for a long time, disconnect the power line from the Motion controller or servo amplifier.
- Place the Motion controller and servo amplifier in static electricity preventing vinyl bags and store.
- When storing for a long time, please contact with our sales representative.
  Also, execute a trial operation.
- When fumigants that contain halogen materials such as fluorine, chlorine, bromine, and iodine are used for disinfecting and protecting wooden packaging from insects, they cause malfunction when entering our products.
  Please take necessary precautions to ensure that remaining materials from fumigant do not enter our products, or treat packaging with methods other than fumigation (heat method).
  Additionally, disinfect and protect wood from insects before packing products.

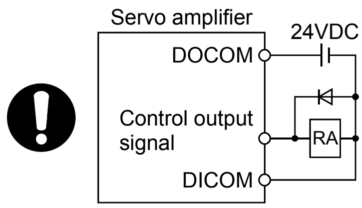
(4) Wiring

⚠CAUTION

- Correctly and securely wire the wires. Reconfirm the connections for mistakes and the terminal screws for tightness after wiring. Failing to do so may lead to run away of the servomotor.
- After wiring, install the protective covers such as the terminal covers to the original positions.
- Do not install a phase advancing capacitor, surge absorber or radio noise filter (option FR-BIF) on the output side of the servo amplifier.
- Correctly connect the output side (terminal U, V, W) and ground. Incorrect connections will lead the servomotor to operate abnormally.
- Do not connect a commercial power supply to the servomotor, as this may lead to trouble.
- Do not mistake the direction of the surge absorbing diode installed on the DC relay for the control signal output of brake signals, etc. Incorrect installation may lead to signals not being output when trouble occurs or the protective functions not functioning.



For the sink output interface



For the source output interface

- Do not connect or disconnect the connection cables between each unit, the encoder cable or PLC expansion cable while the power is ON.
- Securely tighten the cable connector fixing screws and fixing mechanisms. Insufficient fixing may lead to the cables coming off during operation.
- Do not bundle the power line or cables.

(5) Trial operation and adjustment

⚠CAUTION

- Confirm and adjust the program and each parameter before operation. Unpredictable movements may occur depending on the machine.
- Extreme adjustments and changes may lead to unstable operation, so never make them.
- When using the absolute position system function, on starting up, and when the Motion controller or absolute position motor has been replaced, always perform a home position return.
- Before starting test operation, set the parameter speed limit value to the slowest value, and make sure that operation can be stopped immediately by the forced stop, etc. if a hazardous state occurs.

(6) Usage methods

⚠CAUTION

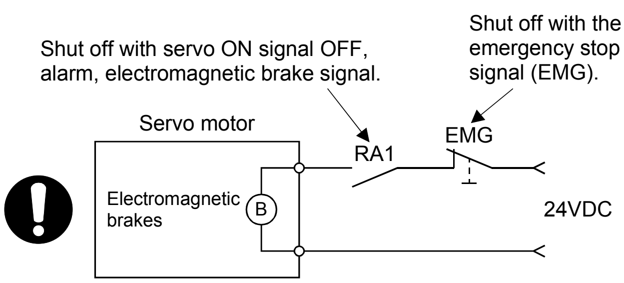
- Immediately turn OFF the power if smoke, abnormal sounds or odors are emitted from the Motion controller, servo amplifier or servomotor.
- Always execute a test operation before starting actual operations after the program or parameters have been changed or after maintenance and inspection.
- Do not attempt to disassemble and repair the units excluding a qualified technician whom our company recognized.
- Do not make any modifications to the unit.
- Keep the effect or electromagnetic obstacles to a minimum by installing a noise filter or by using wire shields, etc. Electromagnetic obstacles may affect the electronic devices used near the Motion controller or servo amplifier.
- When using the CE Mark-compliant equipment, refer to the User's manual for the Motion controllers and refer to the corresponding EMC guideline information for the servo amplifiers, inverters and other equipment.
- Use the units with the following conditions.

Item	Conditions
Input power	According to each instruction manual.
Input frequency	According to each instruction manual.
Tolerable momentary power failure	According to each instruction manual.

(7) Corrective actions for errors

⚠CAUTION

- If an error occurs in the self diagnosis of the Motion controller or servo amplifier, confirm the check details according to the instruction manual, and restore the operation.
- If a dangerous state is predicted in case of a power failure or product failure, use a servomotor with electromagnetic brakes or install a brake mechanism externally.
- Use a double circuit construction so that the electromagnetic brake operation circuit can be operated by emergency stop signals set externally.



- If an error occurs, remove the cause, secure the safety and then resume operation after alarm release.
- The unit may suddenly resume operation after a power failure is restored, so do not go near the machine. (Design the machine so that personal safety can be ensured even if the machine restarts suddenly.)

(8) Maintenance, inspection and part replacement

⚠CAUTION

- Perform the daily and periodic inspections according to the instruction manual.
- Perform maintenance and inspection after backing up the program and parameters for the Motion controller and servo amplifier.
- Do not place fingers or hands in the clearance when opening or closing any opening.
- Periodically replace consumable parts such as batteries according to the instruction manual.
- Do not touch the lead sections such as ICs or the connector contacts.
- Before touching the module, always touch grounded metal, etc. to discharge static electricity from human body. Failure to do so may cause the module to fail or malfunction.
- Do not directly touch the module's conductive parts and electronic components. Touching them could cause an operation failure or give damage to the module.
- Do not place the Motion controller or servo amplifier on metal that may cause a power leakage or wood, plastic or vinyl that may cause static electricity buildup.
- Do not perform a megger test (insulation resistance measurement) during inspection.
- When replacing the Motion controller or servo amplifier, always set the new module settings correctly.
- When the Motion controller or absolute value motor has been replaced, carry out a home position return operation using one of the following methods, otherwise position displacement could occur.
  1) After writing the servo data to the Motion controller using programming software, switch on the power again, then perform a home position return operation.
  2) Using the backup function of the programming software, load the data backed up before replacement.
- After maintenance and inspections are completed, confirm that the position detection of the absolute position detector function is correct.
- Do not drop or impact the battery installed to the module. Doing so may damage the battery, causing battery liquid to leak in the battery. Do not use the dropped or impacted battery, but dispose of it.
- Do not short circuit, charge, overheat, incinerate or disassemble the batteries.
- The electrolytic capacitor will generate gas during a fault, so do not place your face near the Motion controller or servo amplifier.
- The electrolytic capacitor and fan will deteriorate. Periodically replace these to prevent secondary damage from faults. Replacements can be made by our sales representative.
- Lock the control panel and prevent access to those who are not certified to handle or install electric equipment.
- Do not burn or break a module and servo amplifier. Doing so may cause a toxic gas.

(9) About processing of waste

When you discard Motion controller, servo amplifier, a battery (primary battery) and other option articles, please follow the law of each country (area).

⚠CAUTION
● This product is not designed or manufactured to be used in equipment or systems in situations that can affect or endanger human life. ● When considering this product for operation in special applications such as machinery or systems used in passenger transportation, medical, aerospace, atomic power, electric power, or submarine repeating applications, please contact your nearest Mitsubishi Electric sales representative. ● Although this product was manufactured under conditions of strict quality control, you are strongly advised to install safety devices to forestall serious accidents when it is used in facilities where a breakdown in the product is likely to cause a serious accident.

(10) General cautions

- All drawings provided in the instruction manual show the state with the covers and safety partitions removed to explain detailed sections. When operating the product, always return the covers and partitions to the designated positions, and operate according to the instruction manual.

REVISIONS

＊The manual number is given on the bottom left of the back cover.

Print Date	＊Manual Number	Revision
Sep., 2007	IB(NA)-0300135-A	First edition
Nov., 2009	IB(NA)-0300135-B	[Additional model] MR-J3W-☐B, MR-J3-☐B-RJ080W, MR-J3-☐BS [Additional correction/partial correction] Safety precautions, About Manuals, Restrictions by the software's version or serial number, Advanced S-curve acceleration/deceleration, Error code list, Warranty
Sep., 2011	IB(NA)-0300135-C	[Additional model] Q173DCPU-S1, Q172DCPU-S1, GX Works2, MR Configurator2 [Additional function] Operation control program (Type conversion (DFLT, SFLT)), Vision system dedicated function (MVOPEN, MVLOAD, MVTRG, MVPST, MVIN, MVFIN, MVCLOSE, MVCOM), Vision system connection function [Additional correction/partial correction] Safety precautions, About Manuals, Restrictions by the software's version, Error code list
May., 2012	IB(NA)-0300135-D	[Additional model] Q173DSCPU, Q172DSCPU, MR-J4-☐B, MR-J4W-☐B [Additional function] Motion dedicated PLC instruction (Torque limit value individual change request instruction (D(P).CHGT2)), Operation control program (Motion-dedicated function (CHGT2, CHGP), Vision system dedicated function (MVOUT), Data control (SCL, DSCL), Program control (IF - ELSE - IEND, SELECT - CASE - SEND, FOR - NEXT, BREAK)) [Additional correction/partial correction] About Manuals, Manual Page Organization, Restrictions by the software's version, Programming software version, PI-PID switching command (M3217+20n), Parameter error No. (#8009+20n), Servo status1 (#8010+20n), Servo status2 (#8011+20n), Servo status3 (#8012+20n), Product information list device (#8736 to #8751), Motion error history device (#8640 to #8735), Limited count for repeat control of task parameters, Error code list, Processing times
Sep., 2012	IB(NA)-0300135-E	[Additional function] Advanced synchronous control, Motion dedicated PLC instruction (Current value change request instruction of command generation axis (D(P).CHGAS), Speed change instruction of command generation axis (D(P).CHGVS)), Operation control program (Synchronous control dedicated function (CAMRD, CAMWR, CAMWR2, CAMMK, CAMPSCL)) [Additional correction/partial correction] About Manuals, Restrictions by the software's version, Programming software version, Positioning dedicated devices (Internal relays (M8192 to M12063), Data registers (D8192 to D19823)), Error code list, Processing times

Print Date	＊Manual Number	Revision
Apr., 2013	IB(NA)-0300135-F	[Additional function] Operation control program (Others (RTO, RFROM)) [Additional correction/partial correction] About Manuals, Restrictions by the software's version, [Rq.324] Connection command of synchronous encoder via device/master CPU (M11602+4n), [Md.412] Execute cam axis length per cycle (D13622+30n, D13623+30n), [Pr.422] Cam axis length per cycle change setting (D115059+150n), Operation control program (Synchronous control dedicated function (CAMMK)), Error code list, Processing times
Nov., 2013	IB(NA)-0300135-G	[Additional correction/partial correction] Safety precautions, Restrictions by the software's version, Motion error history device error information
Dec., 2015	IB(NA)-0300135-H	[Additional correction/partial correction] Restrictions by the software's version, Structure of the Motion CPU program, Operation control programs (Synchronous control dedicated function(CAMRD, CAMWR)), Servo status7 (#8018+20n), Motion error history device error information, Motion SFC error code list, Warranty
Mar., 2017	IB(NA)-0300135-J	[Additional correction/partial correction] Safety precautions, Restrictions by the software's version, Internal operation data types of operation control programs, Task and interrupt processing, Motion SFC error code list, Warranty

Japanese Manual Number IB(NA)-0300127

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2007 MITSUBISHI ELECTRIC CORPORATION

INTRODUCTION

Thank you for choosing the Mitsubishi Electric Motion controller Q173D(S)CPU/Q172D(S)CPU.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the Motion controller you have purchased, so as to ensure correct use.

CONTENTS

About Manuals

The following manuals are also related to this product.
In necessary, order them by quoting the details in the tables below.

Related Manuals

(1) Motion controller

Manual Name	Manual Number (Model Code)
Q173D(S)CPU/Q172D(S)CPU Motion controller User's Manual This manual explains specifications of the Motion CPU modules, Q172DLX Servo external signal interface module, Q172DEX Synchronous encoder interface module, Q173DPX Manual pulse generator interface module, Power supply modules, Servo amplifiers, SSCNETⅢ cables and Synchronous encoder, and the maintenance/inspection for the system, trouble shooting and others.	IB-0300133 (1XB927)
Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON) This manual explains the Multiple CPU system configuration, performance specifications, common parameters, auxiliary/applied functions, error lists and others.	IB-0300134 (1XB928)
Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (Motion SFC) This manual explains the functions, programming, debugging, error lists for Motion SFC and others.	IB-0300135 (1XB929)
Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE) This manual explains the servo parameters, positioning instructions, device lists, error lists and others.	IB-0300136 (1XB930)
Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE) This manual explains the dedicated instructions to use the synchronous control by virtual main shaft, mechanical system program create mechanical module, servo parameters, positioning instructions, device lists, error lists and others.	IB-0300137 (1XB931)
Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control) This manual explains the dedicated instructions to use the synchronous control by synchronous control parameters, device lists, error lists and others.	IB-0300198 (1XB953)
Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (Safety Observation) This manual explains the details, safety parameters, safety sequence program instructions, device lists and error lists and others for safety observation function by Motion controller.	IB-0300183 (1XB945)
Motion controller Setup Guidance (MT Developer2 Version1) This manual explains the items related to the setup of the Motion controller programming software MT Developer2.	IB-0300142 (—)

(2) PLC

Manual Name	Manual Number (Model Code)
QCPU User's Manual (Hardware Design, Maintenance and Inspection) This manual explains the specifications of the QCPU modules, power supply modules, base units, extension cables, memory card battery, and the maintenance/inspection for the system, trouble shooting, error codes and others.	SH-080483ENG (13JR73)
QnUCPU User's Manual (Function Explanation, Program Fundamentals) This manual explains the functions, programming methods and devices and others to create programs with the QCPU.	SH-080807ENG (13JZ27)
QCPU User's Manual (Multiple CPU System) This manual explains the Multiple CPU system overview, system configuration, I/O modules, communication between CPU modules and communication with the I/O modules or intelligent function modules.	SH-080485ENG (13JR75)
QnUCPU User's Manual (Communication via Built-in Ethernet Port) This manual explains functions for the communication via built-in Ethernet port of the CPU module.	SH-080811ENG (13JZ29)
MELSEC-Q/L Programming Manual (Common Instruction) This manual explains how to use the sequence instructions, basic instructions, application instructions and micro computer program.	SH-080809ENG (13JW10)
MELSEC-Q/L/QnA Programming Manual (PID Control Instructions) This manual explains the dedicated instructions used to exercise PID control.	SH-080040 (13JF59)
MELSEC-Q/L/QnA Programming Manual (SFC) This manual explains the system configuration, performance specifications, functions, programming, debugging, error codes and others of MELSAP3.	SH-080041 (13JF60)
I/O Module Type Building Block User's Manual This manual explains the specifications of the I/O modules, connector, connector/terminal block conversion modules and others.	SH-080042 (13JL99)
MELSEC-L SSCNETⅢ/H Head Module User's Manual This manual explains specifications of the head module, procedures before operation, system configuration, installation, wiring, settings, and troubleshooting.	SH-081152ENG (13JZ78)

(3) Servo amplifier

Manual Name	Manual Number (Model Code)
SSCNETⅢ/H Interface AC Servo MR-J4-_B_(-RJ) Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for AC Servo MR-J4-_B_(-RJ) Servo amplifier.	SH-030106 (1CW805)
SSCNETⅢ/H Interface Multi-axis AC Servo MR-J4W2-_B/MR-J4W3-_B/MR-J4W2-0303B6 Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Multi-axis AC Servo MR-J4W2-_B/MR-J4W3-_B/MR-J4W2-0303B6 Servo amplifier.	SH-030105 (1CW806)
SSCNETⅢ interface MR-J3-□B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for MR-J3-□B Servo amplifier.	SH-030051 (1CW202)
SSCNETⅢ interface 2-axis AC Servo Amplifier MR-J3W-0303BN6/MR-J3W-□B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for 2-axis AC Servo Amplifier MR-J3W-0303BN6/MR-J3W-□B Servo amplifier.	SH-030073 (1CW604)
SSCNETⅢ Compatible Linear Servo MR-J3-□B-RJ004U□ Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Linear Servo MR-J3-□B-RJ004U□ Servo amplifier.	SH-030054 (1CW943)
SSCNETⅢ Compatible Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier.	SH-030056 (1CW304)
SSCNETⅢ Interface Direct Drive Servo MR-J3-□B-RJ080W Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Direct Drive Servo MR-J3-□B-RJ080W Servo amplifier.	SH-030079 (1CW601)
SSCNETⅢ interface Drive Safety integrated MR-J3-□B Safety Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for safety integrated MR-J3-□B Safety Servo amplifier.	SH-030084 (1CW205)

Manual Page Organization

The symbols used in this manual are shown below.

Symbol	Description
QDS	Symbol that indicates correspondence to only Q173DSCPU/Q172DSCPU.
QD	Symbol that indicates correspondence to only Q173DCPU(-S1)/Q172DCPU(-S1).

MEMO

1. OVERVIEW

1.1 Overview

This programming manual describes the Motion SFC program of the operating system software "SW8DNC-SV13Q□", "SW8DNC-SV22Q□" for Motion CPU module (Q173D(S)CPU/Q172D(S)CPU).

In this manual, the following abbreviations are used.

Generic term/Abbreviation	Description
Q173D(S)CPU/Q172D(S)CPU or Motion CPU (module)	Q173DSCPU/Q172DSCPU/Q173DCPU/Q172DCPU/Q173DCPU-S1/ Q172DCPU-S1 Motion CPU module
Q172DLX/Q172DEX/Q173DPX/ Q173DSXY or Motion module	Q172DLX Servo external signals interface module/ Q172DEX Synchronous encoder interface module(Note-1)/ Q173DPX Manual pulse generator interface module/ Q173DSXY Safety signal module
MR-J4(W)-□B	Servo amplifier model MR-J4-□B/MR-J4W-□B
MR-J3(W)-□B	Servo amplifier model MR-J3-□B/MR-J3W-□B
AMP or Servo amplifier	General name for "Servo amplifier model MR-J4-□B/MR-J4W-□B/MR-J3-□B/ MR-J3W-□B"
QCPU, PLC CPU or PLC CPU module	QnUD(E)(H)CPU/QnUDVCPU
Multiple CPU system or Motion system	Abbreviation for "Multiple PLC system of the Q series"
CPUn	Abbreviation for "CPU No.n (n= 1 to 4) of the CPU module for the Multiple CPU system"
Operating system software	General name for "SW7DNC-SV□Q□/SW8DNC-SV□Q□"
SV13	Operating system software for conveyor assembly use (Motion SFC) : SW8DNC-SV13Q□
SV22	Operating system software for automatic machinery use (Motion SFC) : SW8DNC-SV22Q□
Programming software package	General name for MT Developer2/GX Works2/GX Developer/MR Configurator□
MELSOFT MT Works2	Abbreviation for "Motion controller engineering environment MELSOFT MT Works2"
MT Developer2(Note-2)	Abbreviation for "Motion controller programming software MT Developer2 (Version 1.00A or later)"
GX Works2	Abbreviation for "Programmable controller engineering software MELSOFT GX Works2 (Version 1.15R or later)"
GX Developer	Abbreviation for "MELSEC PLC programming software package GX Developer (Version 8.48A or later)"
MR Configurator□(Note-2)	General name for "MR Configurator/MR Configurator2"
MR Configurator	Abbreviation for "Servo setup software package MR Configurator (Version C0 or later)"
MR Configurator2	Abbreviation for "Servo setup software package MR Configurator2 (Version 1.01B or later)"
Manual pulse generator or MR-HDP01	Abbreviation for "Manual pulse generator (MR-HDP01)"
Serial absolute synchronous encoder or Q171ENC-W8/Q170ENC	Abbreviation for "Serial absolute synchronous encoder (Q171ENC-W8/ Q170ENC)"
SSCNETⅢ/H(Note-3)	High speed synchronous network between Motion controller and servo amplifier
SSCNETⅢ(Note-3)	

Generic term/Abbreviation	Description
SSCNETⅢ(/H)(Note-3)	General name for SSCNETⅢ/H, SSCNETⅢ
Absolute position system	General name for "system using the servomotor and servo amplifier for absolute position"
Battery holder unit	Battery holder unit (Q170DBATC)
Intelligent function module	General name for module that has a function other than input or output such as A/D converter module and D/A converter module.
SSCNETⅢ/H head module	Abbreviation for "MELSEC-L series SSCNETⅢ/H head module (LJ72MS15)"
Optical hub unit or MR-MV200	Abbreviation for "SSCNETⅢ/H compatible optical hub unit (MR-MV200)"

(Note-1): Q172DEX can be used in SV22.

(Note-2): This software is included in Motion controller engineering environment "MELSOFT MT Works2".

(Note-3): SSCNET: Servo System Controller NETwork

REMARK

For information about each module, design method for program and parameter, refer to the following manuals relevant to each module.

Item		Reference Manual
Motion CPU module/Motion unit		Q173D(S)CPU/Q172D(S)CPU Motion controller User's Manual
PLC CPU, peripheral devices for sequence program design, I/O modules and intelligent function module		Manual relevant to each module
Operation method for MT Developer2		Help of each software
SV13/SV22	• Multiple CPU system configuration • Performance specification • Design method for common parameter • Auxiliary and applied functions (common)	Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)
	• Design method for positioning control program in the real mode • Design method for positioning control parameter	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)
	• Design method for safety observation parameter • Design method for user made safety sequence program	Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (Safety Observation)
SV22 (Virtual mode)	• Design method for mechanical system program	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)
SV22 (Advanced synchronous control)	• Design method for synchronous control parameter	Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)

1.2 Features

The Motion CPU and Motion SFC program have the following features.

1.2.1 Features of Motion SFC programs

(1) Since a program intelligible for anyone can be created in flow chart form by making a sequence of machine operation correspond to each operation step, maintenance nature improves.

(2) Since transition conditions are judged with Motion CPU side and positioning starts, there is not dispersion in the response time influenced by PLC scan time.

(3) High speed and high response processing is realizable with the step processing method (only active steps) of Motion SFC.

(4) Not only positioning control but also numerical operations, device SET/RST, etc. can be processed with Motion CPU side, making via PLC CPU is unnecessary and a tact time can be shortened.

(5) By transition condition description peculiar to Motion SFC, the instructions to servo amplifier is possible at completion of starting condition.

(6) By transition condition description peculiar to Motion SFC, after starting, transition to next step is possible without waiting for positioning completion.

(7) Motion SFC program that responds it at high speed for interrupt input from external source can be executed.

(8) Motion SFC program can be executed in the fixed cycle (Min. 0.22ms: Q17□DSCPU use) by synchronizing to the Motion operation cycle.

1.2.2 Performance specifications

(1) Basic specifications of Q173DCPU/Q172DCPU

(a) Motion control specifications

Item		Q173DSCPU	Q172DSCPU	Q173DCPU(-S1)	Q172DCPU(-S1)
Number of control axes		Up to 32 axes	Up to 16 axes	Up to 32 axes	Up to 8 axes
Operation cycle (default)	SV13	0.22ms/ 1 to 4 axes 0.44ms/ 5 to 10 axes 0.88ms/ 11 to 24 axes 1.77ms/25 to 32 axes	0.22ms/ 1 to 4 axes 0.44ms/ 5 to 10 axes 0.88ms/ 11 to 16 axes	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 18 axes 1.77ms/19 to 32 axes	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 8 axes
	SV22	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 16 axes 1.77ms/17 to 32 axes	0.44ms/ 1 to 6 axes 0.88ms/ 7 to 16 axes	0.44ms/ 1 to 4 axes 0.88ms/ 5 to 12 axes 1.77ms/13 to 28 axes 3.55ms/29 to 32 axes	0.44ms/ 1 to 4 axes 0.88ms/ 5 to 8 axes
Interpolation functions		Linear interpolation (Up to 4 axes), Circular interpolation (2 axes), Helical interpolation (3 axes)			
Control modes		PTP(Point to Point) control, Speed control, Speed-position switching control, Fixed-pitch feed, Constant speed control, Position follow-up control, Speed control with fixed position stop, Speed switching control, High-speed oscillation control, Speed-torque control, Synchronous control (SV22 (Virtual mode switching method/Advanced synchronous control method))		PTP(Point to Point) control, Speed control, Speed-position switching control, Fixed-pitch feed, Constant speed control, Position follow-up control, Speed control with fixed position stop, Speed switching control, High-speed oscillation control, Synchronous control (SV22)	
Acceleration/deceleration control		Trapezoidal acceleration/deceleration, S-curve acceleration/deceleration, Advanced S-curve acceleration/deceleration			
Compensation		Backlash compensation, Electronic gear, Phase compensation (SV22)			
Programming language		Motion SFC, Dedicated instruction, Mechanical support language (SV22) (Note-1)		Motion SFC, Dedicated instruction, Mechanical support language (SV22)	
Servo program capacity		16k steps			
Number of positioning points		3200 points (Positioning data can be designated indirectly)			
Peripheral I/F		USB/RS-232/Ethernet (Via PLC CPU) PERIPHERAL I/F (Motion CPU)		USB/RS-232/Ethernet (Via PLC CPU) PERIPHERAL I/F (Motion CPU) (Note-2)	
Home position return function		Proximity dog method (2 types), Count method (3 types), Data set method (2 types), Dog cradle method, Stopper method (2 types), Limit switch combined method, Scale home position signal detection method, Dogless home position signal reference method, Driver home position return method		Proximity dog method (2 types), Count method (3 types), Data set method (2 types), Dog cradle method, Stopper method (2 types), Limit switch combined method, Scale home position signal detection method	
		Home position return re-try function provided, home position shift function provided			
JOG operation function		Provided			
Manual pulse generator operation function		Possible to connect 3 modules (Q173DPX use) Possible to connect 1 module (Built-in interface in Motion CPU use) (Note-3)		Possible to connect 3 modules (Q173DPX use)	
Synchronous encoder operation function (Note-4)		Possible to connect 12 module (SV22 use) (Q172DEX + Q173DPX + Built-in interface in Motion CPU + Via device (Note-5) + Via servo amplifier (Note-5), (Note-6) + Multiple CPU synchronous control (Note-5))		Possible to connect 12 modules (SV22 use) (Q172DEX + Q173DPX)	Possible to connect 8 modules (SV22 use) (Q172DEX + Q173DPX)
M-code function		M-code output function provided, M-code completion wait function provided			

Motion control specifications (continued)

<table>
<tr><th colspan="2">Item</th><th>Q173DSCPU</th><th>Q172DSCPU</th><th>Q173DCPU(-S1)</th><th>Q172DCPU(-S1)</th></tr>
<tr><td rowspan="2">Limit switch output function</td><td>SV13</td><td colspan="4">Number of output points 32 points
Watch data: Motion control data/Word device</td></tr>
<tr><td>SV22</td><td colspan="2">Virtual mode switching method:
Number of output points 32 points
Advanced synchronous control method:
Number of output points 64 points × 2 settings
Output timing compensation
Watch data: Motion control data/Word device</td><td colspan="2">Number of output points 32 points
Watch data: Motion control data/Word device</td></tr>
<tr><td colspan="2">ROM operation function</td><td colspan="4">Provided</td></tr>
<tr><td colspan="2">Multiple CPU synchronous control (Note-5)</td><td colspan="2">Provided</td><td colspan="2">None</td></tr>
<tr><td colspan="2">External input signal</td><td colspan="2">Q172DLX, External input signals (FLS/RLS/DOG) of servo amplifier,
Built-in interface in Motion CPU (DI), Bit device</td><td colspan="2">Q172DLX or External input signals (FLS/RLS/DOG) of servo amplifier</td></tr>
<tr><td colspan="2">High-speed reading function (Note-7)</td><td colspan="2">Provided
(Via built-in interface in Motion CPU,
Via input module,
Via tracking of Q172DEX/Q173DPX)</td><td colspan="2">Provided
(Via input module, Via tracking of Q172DEX/Q173DPX)</td></tr>
<tr><td colspan="2">Forced stop</td><td colspan="4">Motion controller forced stop (EMI connector, System setting),
Forced stop terminal of servo amplifier</td></tr>
<tr><td colspan="2">Number of I/O points</td><td colspan="2">Total 256 points
(Built-in interface in Motion CPU (Input 4 points) + I/O module + Intelligent function module)</td><td colspan="2">Total 256 points
(I/O module)</td></tr>
<tr><td rowspan="3">Mark detection function</td><td>Mark detection mode setting</td><td colspan="2">Continuous detection mode,
Specified number of detection mode,
Ring buffer mode</td><td colspan="2" rowspan="3">None</td></tr>
<tr><td>Mark detection signal</td><td colspan="2">Built-in interface in Motion CPU (4 points),
Bit device, DOG/CHANGE signal of Q172DLX</td></tr>
<tr><td>Mark detection setting</td><td colspan="2">32 settings</td></tr>
<tr><td colspan="2">Clock function</td><td colspan="4">Provided</td></tr>
<tr><td colspan="2">Security function</td><td colspan="2">Provided
(Protection by software security key or password)</td><td colspan="2">Provided
(Protection by password)</td></tr>
<tr><td colspan="2">All clear function</td><td colspan="4">Provided</td></tr>
<tr><td colspan="2">Remote operation</td><td colspan="4">Remote RUN/STOP, Remote latch clear</td></tr>
<tr><td rowspan="2">Optional data monitor function</td><td>SSCNETⅢ/H</td><td colspan="2">Up to 6 data/axis
(Communication data: Up to 6 points/axis)</td><td colspan="2">None</td></tr>
<tr><td>SSCNETⅢ</td><td colspan="4">Up to 3 data/axis
(Communication data: Up to 3 points/axis)</td></tr>
<tr><td colspan="2">Digital oscilloscope function</td><td colspan="2">Motion buffering method
(Real-time waveform can be displayed)
Sampling data: Word 16CH, Bit 16CH</td><td colspan="2">Motion buffering method
(Real-time waveform can be displayed)
Sampling data: Word 4CH, Bit 8CH</td></tr>
<tr><td colspan="2">Absolute position system</td><td colspan="4">Made compatible by setting battery to servo amplifier.
(Possible to select the absolute data method or incremental method for each axis)</td></tr>
<tr><td rowspan="2">SSCNET communication (Note-8)</td><td>Communication type</td><td colspan="2">SSCNETⅢ/H, SSCNETⅢ</td><td colspan="2">SSCNETⅢ</td></tr>
<tr><td>Number of lines</td><td>2 lines (Note-9)</td><td>1 line (Note-9)</td><td>2 lines</td><td>1 line</td></tr>
<tr><td colspan="2">Driver communication function (Note-10)</td><td colspan="2">Provided</td><td colspan="2">None</td></tr>
</table>

Motion control specifications (continued)

<table>
<tr><th colspan="2">Item</th><th>Q173DSCPU</th><th>Q172DSCPU</th><th>Q173DCPU(-S1)</th><th>Q172DCPU(-S1)</th></tr>
<tr><td rowspan="3">Number of Motion related modules</td><td>Q172DLX</td><td>4 modules usable</td><td>2 modules usable</td><td>4 modules usable</td><td>1 module usable</td></tr>
<tr><td>Q172DEX</td><td colspan="3">6 modules usable</td><td>4 modules usable</td></tr>
<tr><td>Q173DPX</td><td colspan="3">4 modules usable (Note-11)</td><td>3 modules usable (Note-11)</td></tr>
<tr><td colspan="2">Number of SSCNETⅢ/H head module connection stations</td><td>Up to 8 stations usable (Up to 4 stations/line)</td><td>Up to 4 stations usable</td><td colspan="2">Unusable</td></tr>
<tr><td colspan="2">Number of optical hub unit connections</td><td>Up to 32 units usable (Up to 16 units/line)</td><td>Up to 16 units usable</td><td colspan="2">Unusable</td></tr>
</table>

(Note-1): SV22 virtual mode only
(Note-2): Q173DCPU-S1/Q172DCPU-S1 only
(Note-3): When the manual pulse generator is used via the built-in interface in Motion CPU, the Q173DPX cannot be used.
(Note-4): Any incremental synchronous encoder connected to the built-in interface in Motion CPU will automatically be assigned an Axis No. one integer greater than the number of encoders connected to any Q172DEX modules and Q173DPX modules.
(Note-5): SV22 advanced synchronous control only
(Note-6): Servo amplifier (MR-J4-□B-RJ) only
(Note-7): This cannot be used in SV22 advanced synchronous control.
(Note-8): The servo amplifiers for SSCNET cannot be used.
(Note-9): SSCNETⅢ and SSCNETⅢ/H cannot be combined in the same line.
For Q173DSCPU, SSCNETⅢ or SSCNETⅢ/H can be set every line.
(Note-10): Servo amplifier (MR-J3-□B/MR-J4-□B) only.
(Note-11): When using the incremental synchronous encoder (SV22 use), you can use above number of modules.
When connecting the manual pulse generator, you can use only 1 module.

(b) Motion SFC Performance Specifications

Item				Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)
Motion SFC program capacity	Code total (Motion SFC chart + Operation control + Transition)			652k bytes	543k bytes
	Text total (Operation control + Transition)			668k bytes	484k bytes
Motion SFC program	Number of Motion SFC programs			256 (No.0 to 255)	
	Motion SFC chart size/program			Up to 64k bytes (Included Motion SFC chart comments)	
	Number of Motion SFC steps/program			Up to 4094 steps	
	Number of selective branches/branch			255	
	Number of parallel branches/branch			255	
	Parallel branch nesting			Up to 4 levels	
Operation control program (F/FS) / Transition program (G)	Number of operation control programs			4096 with F(Once execution type) and FS(Scan execution type) combined. (F/FS0 to F/FS4095)	
	Number of transition programs			4096(G0 to G4095)	
	Code size/program			Up to approx. 64k bytes (32766 steps)	
	Number of blocks(line)/program			Up to 8192 blocks (in the case of 4 steps(min)/blocks)	
	Number of characters/block			Up to 128 (comment included)	
	Number of operand/block			Up to 64 (operand: constants, word device, bit devices)	
	( ) nesting/block			Up to 32 levels	
	Descriptive expression	Operation control program		Calculation expression, bit conditional expression, branch/repetition processing	Calculation expression, bit conditional expression
		Transition program		Calculation expression/bit conditional expression/ comparison conditional expression	
Execute specification	Number of multi execute programs			Up to 256	
	Number of multi active steps			Up to 256 steps/all programs	
	Executed task	Normal task		Execute in main cycle of Motion CPU	
		Event task (Execution can be masked.)	Fixed cycle	Execute in fixed cycle (0.22ms, 0.44ms, 0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)	Execute in fixed cycle (0.44ms, 0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)
			External interrupt	Executes when the input set to the event task factor in the input module controlled by the Motion CPU (16 points) turns ON.	
			PLC interrupt	Execute with interrupt instruction (D(P).GINT) from PLC CPU.	
		NMI task		Executes when the input set to the NMI task factor in the input module controlled by the Motion CPU (16 points) turns ON.	
Number of I/O points (X/Y)				8192 points	
Number of real I/O points (PX/PY)				256 points (Built-in interface in Motion CPU (Input 4 points) + I/O module + Intelligent function module)	256 points (I/O module)

1.2.3 Operation control/transition control specifications

(1) Table of the operation control/transition control specifications

<table>
<tr><th>Item</th><th colspan="3">Specifications</th><th>Remark</th></tr>
<tr><td rowspan="3">Expression</td><td colspan="2">Calculation expression</td><td>Returns a numeric result.
Expressions for calculating indirectly specified data using constants and word devices.</td><td>D100+1,SIN(D100), etc.</td></tr>
<tr><td rowspan="2">Conditional expression</td><td>Bit conditional expression</td><td>Returns a true or false result.
Expression for judging ON or OFF of bit device.</td><td>M0, !M0, M1*M0,
(M1+M2)*(!M3+M4), etc.</td></tr>
<tr><td>Comparison conditional expression</td><td>Expressions for comparing indirectly specified data and calculation expressions using constants and word devices.</td><td>D100==100
D10<D102+D10, etc.</td></tr>
<tr><td>Bit devices</td><td colspan="3">
<table>
<tr><th colspan="2" rowspan="2">Device</th><th rowspan="2">Symbol</th><th colspan="2">Accessibility</th><th colspan="3">Usable tasks</th><th rowspan="2">Description example</th></tr>
<tr><th>Read</th><th>Write</th><th>Normal</th><th>Event</th><th>NMI</th></tr>
<tr><td rowspan="2">Input</td><td>Input module non-loaded range</td><td>X</td><td>○</td><td>○</td><td rowspan="10">○</td><td rowspan="10">○</td><td rowspan="10">○</td><td>X100</td></tr>
<tr><td>Input module loaded range</td><td>PX</td><td>○</td><td>×</td><td>PX180</td></tr>
<tr><td rowspan="2">Output</td><td>Output module non-loaded range</td><td>Y</td><td>○</td><td>○</td><td>Y100</td></tr>
<tr><td>Output module loaded range</td><td>PY</td><td>○</td><td>○</td><td>PY1E0</td></tr>
<tr><td colspan="2">Internal relay</td><td>M</td><td>○</td><td>○</td><td>M20</td></tr>
<tr><td rowspan="2">Multiple CPU area device</td><td>Self CPU</td><td rowspan="2">U□\G□.□</td><td>○</td><td>○</td><td rowspan="2">U3E0\G10200.A</td></tr>
<tr><td>Other CPU</td><td>○</td><td>×</td></tr>
<tr><td colspan="2">Link relay</td><td>B</td><td>○</td><td>○</td><td>B3FF</td></tr>
<tr><td colspan="2">Annunciator</td><td>F</td><td>○</td><td>○</td><td>F0</td></tr>
<tr><td colspan="2">Special relay</td><td>SM</td><td>○</td><td>○</td><td>SM0</td></tr>
</table>
○: Usable
×: Unusable

CAUTION
<Restrictions on write-enabled bit devices>
1) Write to device X is allowed only within the input module non-installed range.
2) Special relay has predetermined applications in the system.
Do not perform write to other than the user setting device.
</td><td>The input X/output Y are written with the actual input PX/actual output PY.
It does the layout of the I/O numbers of PX, PY by a set up of as system.
(In the operation control program/transition program, automatically represented as PX/PY according to the system setting information.)</td></tr>
<tr><td>Word devices</td><td colspan="3">
<table>
<tr><th colspan="2" rowspan="2">Devices</th><th rowspan="2">Symbol</th><th colspan="2">Accessibility</th><th colspan="3">Usable tasks</th><th rowspan="2">Description example</th></tr>
<tr><th>Read</th><th>Write</th><th>Normal</th><th>Event</th><th>NMI</th></tr>
<tr><td colspan="2">Data register</td><td>D</td><td>○</td><td>○</td><td rowspan="7">○</td><td rowspan="7">○</td><td rowspan="7">○</td><td>DOL</td></tr>
<tr><td colspan="2">Link register</td><td>W</td><td>○</td><td>○</td><td>W1F:F</td></tr>
<tr><td rowspan="2">Multiple CPU area device</td><td>Self CPU</td><td rowspan="2">U□\G□</td><td>○</td><td>○</td><td rowspan="2">U3E0\G10100</td></tr>
<tr><td>Other CPU</td><td>○</td><td>×</td></tr>
<tr><td colspan="2">Special register</td><td>SD</td><td>○</td><td>○</td><td>SD0</td></tr>
<tr><td colspan="2">Motion register</td><td>#</td><td>○</td><td>○</td><td>#0F</td></tr>
<tr><td colspan="2">Coasting timer</td><td>FT</td><td>○</td><td>×</td><td>FT</td></tr>
</table>
○: Usable
×: Unusable

CAUTION
<Restrictions on write-enabled word devices>
1) Special register has predetermined applications in the system.
Do not perform write to other than the user-set device.
</td><td></td></tr>
</table>

Table of the operation control/transition control specification (continued)

Item	Specifications			Remark
Data type	(None)	16-bit integer type (signed)	-32768 to 32767	K10, D100, etc.
		16-bit integer type (unsigned)	0 to 65535	
	L	32-bit integer type (signed)	-2147483648 to 2147483647	2000000000, W100L, etc.
		32-bit integer type (unsigned)	0 to 4294967295	
	F	64-bit floating-point type (double precision real number type)	IEEE format	1.23, #10F, etc.
Constant	K	Decimal constant	The above data type symbol 'L' or '. (decimal point)' provided at the end indicates the data type. The constant without the data type is regarded as the applicable minimum type.	K-100, H0FFL, etc. 'K' may be omitted.
	H	Hexadecimal constant		
Number of instructions	Binary operation	6	90 in total	
	Bit operation	6		
	Sign	1		
	Standard function	15		
	Type conversion	8		
	Bit device status	2		
	Bit device control	5		
	Logical operation	4		
	Comparison operation	6		
	Motion dedicated function	5		
	Others	12		
	Vision system dedicated function	9		
	Data control	2		
	Program control	4		
	Synchronous control dedicated function	5		
Read/write response of input PX, output PY	Input response	Direct read control at instruction execution.		
	Output response	Direct write control at instruction execution.		

(2) Table of the operation control/transition instruction

Classification	Symbol	Function	Format	Basic steps	Usable step F/FS	Usable step G	Y/N transition's conditional expression	Section of reference
Binary operation	=	Substitution	(D)=(S)	4	○	○	—	5.4.1
	+	Addition	(S1)+(S2)	4	○	○	—	5.4.2
	-	Subtraction	(S1)-(S2)	4	○	○	—	5.4.3
	*	Multiplication	(S1)*(S2)	4	○	○	—	5.4.4
	/	Division	(S1)/(S2)	4	○	○	—	5.4.5
	%	Remainder	(S1)%(S2)	4	○	○	—	5.4.6
Bit operation	~	Bit inversion (complement)	~(S)	2	○	○	—	5.5.1
	&	Bit logical AND	(S1)&(S2)	4	○	○	—	5.5.2
	|	Bit logical OR	(S1)|(S2)	4	○	○	—	5.5.3
	^	Bit exclusive logical OR	(S1)^(S2)	4	○	○	—	5.5.4
	>>	Bit right shift	(S1)>>(S2)	4	○	○	—	5.5.5
	<<	Bit left shift	(S1)<<(S2)	4	○	○	—	5.5.6
Sign	-	Sign inversion (complement of 2)	-(S)	2	○	○	—	5.5.7
Standard function	SIN	Sine	SIN(S)	2	○	○	—	5.6.1
	COS	Cosine	COS(S)	2	○	○	—	5.6.2
	TAN	Tangent	TAN(S)	2	○	○	—	5.6.3
	ASIN	Arcsine	ASIN(S)	2	○	○	—	5.6.4
	ACOS	Arccosine	ACOS(S)	2	○	○	—	5.6.5
	ATAN	Arctangent	ATAN(S)	2	○	○	—	5.6.6
	SQRT	Square root	SQRT(S)	2	○	○	—	5.6.7
	LN	Natural logarithm	LN(S)	2	○	○	—	5.6.8
	EXP	Exponential operation	EXP(S)	2	○	○	—	5.6.9
	ABS	Absolute value	ABS(S)	2	○	○	—	5.6.10
	RND	Round-off	RND(S)	2	○	○	—	5.6.11
	FIX	Round-down	FIX(S)	2	○	○	—	5.6.12
	FUP	Round-up	FUP(S)	2	○	○	—	5.6.13
	BIN	BCD ⟶ BIN conversion	BIN(S)	2	○	○	—	5.6.14
	BCD	BIN ⟶ BCD conversion	BCD(S)	2	○	○	—	5.6.15
Type conversion	SHORT	Convert into 16-bit integer type (signed)	SHORT(S)	2	○	○	—	5.7.1
	USHORT	Convert into 16-bit integer type (unsigned)	USHORT(S)	2	○	○	—	5.7.2
	LONG	Convert into 32-bit integer type (signed)	LONG(S)	2	○	○	—	5.7.3
	ULONG	Convert into 32-bit integer type (unsigned)	ULONG(S)	2	○	○	—	5.7.4
	FLOAT	Regard as signed data and convert into 64-bit floating point type	FLOAT(S)	2	○	○	—	5.7.5
	UFLOAT	Regard as unsigned data and convert into 64-bit floating point type	UFLOAT(S)	2	○	○	—	5.7.6
	DFLT	Floating-point value conversion 32-bit into 64-bit	DFLT(S)	2	○	○	—	5.7.7
	SFLT	Floating-point value conversion 64-bit into 32-bit	SFLT(S)	2	○	○	—	5.7.8
Bit device status	(None)	ON (normally open contact)	(S)	2	○	○	○	5.8.1
	!	OFF (normally closed contact)	!(S)	2	○	○	○	5.8.2
Bit device control	SET	Device set	SET(D)	3	○	○	—	5.9.1
			SET(D)= (conditional expression)	4	○	○	—	
	RST	Device reset	RST(D)	3	○	○	—	5.9.2
			RST(D)=(conditional expression)	4	○	○	—	
	DOUT	Device output	DOUT(D),(S)	4	○	○	—	5.9.3
	DIN	Device input	DIN(D),(S)	4	○	○	—	5.9.4
	OUT	Bit device output	OUT(D)=(conditional expression)	4	○	○	—	5.9.5

Table of the operation control/transition instruction (continued)

Classification	Symbol	Function	Format	Basic steps	Usable step		Y/N transition's conditional expression	Section of reference
					F/FS	G		
Logical operation	(None)	Logical acknowledgment	(Conditional expression)	0	○	○	○	5.10.1
	!	Logical negation	!(Conditional expression)	2	○	○	○	5.10.2
	*	Logical AND	(Conditional expression) * (conditional expression)	4	○	○	○	5.10.3
	+	Logical OR	(Conditional expression) + (conditional expression)	4	○	○	○	5.10.4
Comparison operation	==	Equal to	(Conditional expression) == (conditional expression)	4	○	○	○	5.11.1
	!=	Not equal to	(Conditional expression) != (conditional expression)	4	○	○	○	5.11.2
	<	Less than	(Conditional expression) < (conditional expression)	4	○	○	○	5.11.3
	<=	Less than or equal to	(Conditional expression) <= (conditional expression)	4	○	○	○	5.11.4
	>	More than	(Conditional expression) > (conditional expression)	4	○	○	○	5.11.5
	>=	More than or equal to	(Conditional expression) >= (conditional expression)	4	○	○	○	5.11.6
Motion dedicated function	CHGV	Speed change request	CHGV((S1),(S2))	4	○	○	—	5.12.1
	CHGVS	Command generation axis speed change request	CHGVS((S1),(S2))	4	○	○	—	5.12.2
	CHGT	Torque limit value change request	CHGT((S1),(S2))	4	○	○	—	5.12.3
	CHGT2	Torque limit value individual change request	CHGT2((S1),(S2),(S3))	5	○	○	—	5.12.4
	CHGP	Target position change request	CHGP((S1),(S2),(S3))	6	○	○	—	5.12.5
Others	EI	Event task enable	EI	1	○	○	—	5.13.1
	DI	Event task disable	DI	1	○	○	—	5.13.2
	NOP	No operation	NOP	1	○	○	—	5.13.3
	BMOV	Block transfer	BMOV(D),(S),(n)	6	○	○	—	5.13.4
	FMOV	Same data block transfer	FMOV(D),(S),(n)	6	○	○	—	5.13.5
	MULTW	Write device data to CPU shared memory of the self CPU	MULTW(D),(S),(n),(D1)	8	○	○	—	5.13.6
	MULTR	Read device data from CPU shared memory	MULTR(D),(S1),(S2),(n)	7	○	○	—	5.13.7
	TO	Write device data to intelligent function module	TO(D1),(D2),(S),(n)	7	○	○	—	5.13.8
	FROM	Read device data from intelligent function module	FROM(D),(S1),(S2),(n)	7	○	○	—	5.13.9
	RTO	Write buffer memory data to head module	RTO(D1),(D2),(D3),(S),(n),(D4)	11	○	○	—	5.13.10
	RFROM	Read buffer memory data from head module	RFROM(D),(S1),(S2),(S3),(n),(D1)	11	○	○	—	5.13.11
	TIME	Time to wait	TIME(S)	7	—	○	—	5.13.12
Vision system dedicated function	MVOPEN	Open line	MVOPEN(S1),(S2)	4	○	○	—	5.15.1
	MVLOAD	Load a program	MVLOAD(S1),(S2)	4	○	○	—	5.15.2
	MVTRG	Send an image acquisition trigger	MVTRG(S1),(S2)	4	○	○	—	5.15.3
	MVPST	Start a program	MVPST(S1),(S2)	4	○	○	—	5.15.4
	MVIN	Input data	MVIN(S1),(S2),(D),(S3)	8 or more	○	○	—	5.15.5
	MVOUT	Output data	MVOUT(S1),(S2),(S3),(S4)	8 or more	○	○	—	5.15.6
	MVFIN	Reset a status storage device	MVFIN(S)	2	○	○	—	5.15.7
	MVCLOSE	Close line	MVCLOSE(S)	2	○	○	—	5.15.8
	MVCOM	Send a command for native mode	MVCOM(S1),(S2),(D),(S3),(S4)	9 or more	○	○	—	5.15.9
Data control	SCL	16-bit integer type scaling	SLC(S1),(S2),(S3),(D)	8	○	○	—	5.16.1
	DSCL	32-bit integer type scaling	DSCL(S1),(S2),(S3),(D)	8	○	○	—	5.16.2

○: Usable, —: Unusable

Table of the operation control/transition instruction (continued)

Classification	Symbol	Function	Format	Basic steps	Usable step F/FS	Usable step G	Y/N transition's conditional expression	Section of reference

Program control	IF - ELSE - IEND	Conditional branch control	IF(S)
			:
			ELSE
			:
			IEND
				IF : 4
				ELSE : 3
				IEND : 1
					○	○	—	5.17.1

	SELECT - CASE - SEND	Selective branch control	SELECT CASE(S1) : CEND CASE(Sn) : CEND CLELSE : CEND SEND	SELECT : 1 CASE : 4 CEND : 3 CLELSE : 1 SEND : 1	○	○	—	5.17.2
	FOR -NEXT	Repeat control with specified count	FOR(D) = (S1) TO (S2) STEP (S3) : NEXT	FOR : 9 NEXT : 8	○	○	—	5.17.3
	BREAK	Force termination of repeat control	BREAK	3	○	○	—	5.17.4
Synchronous control dedicated function	CAMRD	Cam data read	CAMRD(S1),(S2),(n),(D)	7	○	○	—	5.18.1
	CAMWR	Cam data write	CAMWR(S1),(S2),(n),(S3)	7	○	○	—	5.18.2
	CAMWR2	Cam data write (Cam open area)	CAMWR2(S1),(S2),(n),(S3)	7	○	○	—	5.18.3
	CAMMK	Cam auto-generation	CAMMK(S1),(S2),(S3)	6	○	○	—	5.18.4
	CAMPSCL	Cam position calculation	CAMPSCL(S1),(S2),(D)	6	○	○	—	5.18.5

○: Usable, —: Unusable

(3) Rough calculation expression of single program for operation control/transition program

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

1.2.4 Positioning dedicated devices

(1) Positioning dedicated devices

The following section describes the positioning dedicated devices.
The following device range is valid in the Motion.

Item	Q173DSCPU	Q172DSCPU	Q173DCPU(-S1)	Q172DCPU(-S1)
Number of control axes	32 axes	16 axes	32 axes	8 axes

Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)", "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)", "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for details of the positioning dedicated devices.

(a) Table of the internal relays

• Overall configuration

<table>
<tr><th colspan="2">SV13</th><th colspan="4">SV22</th></tr>
<tr><th rowspan="2">Device No.</th><th rowspan="2">Purpose</th><th colspan="2">Virtual mode switching method</th><th colspan="2">Advanced synchronous control method</th></tr>
<tr><th>Device No.</th><th>Purpose</th><th>Device No.</th><th>Purpose</th></tr>
<tr><td>M0
to</td><td>User device
(2000 points)</td><td>M0
to</td><td>User device
(2000 points)</td><td>M0
to</td><td>User device
(2000 points)</td></tr>
<tr><td>M2000
to</td><td>Common device
(320 points)</td><td>M2000
to</td><td>Common device
(320 points)</td><td>M2000
to</td><td>Common device
(320 points)</td></tr>
<tr><td>M2320
to</td><td>Unusable
(80 points)</td><td>M2320
to</td><td>Unusable
(80 points)</td><td>M2320
to</td><td>Unusable
(80 points)</td></tr>
<tr><td>M2400
to</td><td>Axis status
(20 points × 32 axes)</td><td>M2400
to</td><td>Axis status
(20 points × 32 axes)
Real mode : Each axis
Virtual mode : Output module</td><td>M2400
to</td><td>Axis status
(20 points × 32 axes)</td></tr>
<tr><td>M3040
to</td><td>Unusable
(32 points)</td><td>M3040
to</td><td>Unusable
(32 points)</td><td>M3040
to</td><td>Unusable
(32 points)</td></tr>
<tr><td>M3072
to</td><td>Common device
(Command signal)
(64 points)</td><td>M3072
to</td><td>Common device
(Command signal)
(64 points)</td><td>M3072
to</td><td>Common device
(Command signal)
(64 points)</td></tr>
<tr><td>M3136
to</td><td>Unusable
(64 points)</td><td>M3136
to</td><td>Unusable
(64 points)</td><td>M3136
to</td><td>Unusable
(64 points)</td></tr>
<tr><td>M3200
to</td><td>Axis command signal
(20 points × 32 axes)</td><td>M3200
to</td><td>Axis command signal
(20 points × 32 axes)
Real mode : Each axis
Virtual mode : Output module</td><td>M3200
to</td><td>Axis command signal
(20 points × 32 axes)</td></tr>
<tr><td rowspan="3">M3840
to
M4687</td><td rowspan="3">User device
(848 points)</td><td>M3840
to</td><td>Unusable
(160 points)</td><td rowspan="3">M3840
to
M4687</td><td rowspan="3">User device
(848 points)</td></tr>
<tr><td>M4000
to</td><td>Virtual servomotor axis status (Note-1,2)
(20 points × 32 axes)
(Mechanical system setting axis only)</td></tr>
<tr><td>M4640
to
M4687</td><td>Synchronous encoder axis status (Note-2)
(4 points × 12 axes)</td></tr>
</table>

• Overall configuration (Continued)

<table>
<tr><th colspan="2">SV13</th><th colspan="4">SV22</th></tr>
<tr><th rowspan="2">Device No.</th><th rowspan="2">Purpose</th><th colspan="2">Virtual mode switching method</th><th colspan="2">Advanced synchronous control method</th></tr>
<tr><th>Device No.</th><th>Purpose</th><th>Device No.</th><th>Purpose</th></tr>
<tr><td rowspan="4">M4688
to</td><td rowspan="4">User device
(3504 points)</td><td>M4688
to</td><td>Unusable (Note-1)
(112 points)</td><td rowspan="4">M4688
to</td><td rowspan="4">User device
(3504 points)</td></tr>
<tr><td>M4800
to</td><td>Virtual servomotor axis command signal (Note-1,2)
(20 points × 32 axes)
(Mechanical system setting axis only)</td></tr>
<tr><td>M5440
to</td><td>Synchronous encoder axis command signal (Note-2)
(4 points × 12 axes)</td></tr>
<tr><td>M5488
to</td><td>User device (Note-3)
(2704 points)</td></tr>
<tr><td rowspan="13">M8192
to
M12063</td><td rowspan="13">System area
(3872 points)</td><td rowspan="13">M8192
to
M12063</td><td rowspan="13">System area
(3872 points)</td><td>M8192
to</td><td>System area
(1608 points) QDS Ver.!</td></tr>
<tr><td>M9800
to</td><td>Command generation axis status
(20 points × 32 axes) QDS Ver.!</td></tr>
<tr><td>M10440
to</td><td>Synchronous encoder axis status
(10 points × 12 axes) QDS Ver.!</td></tr>
<tr><td>M10560
to</td><td>Output axis status
(10 points × 32 axes) QDS Ver.!</td></tr>
<tr><td>M10880
to</td><td>Synchronous control signal [St.380]
(32 points) QDS Ver.!</td></tr>
<tr><td>M10912
to</td><td>Synchronous analysis complete signal [St.381]
(32 points) QDS Ver.!</td></tr>
<tr><td>M10944
to</td><td>Unusable
(16 points)</td></tr>
<tr><td>M10960
to</td><td>Command generation axis command signal
(20 points × 32 axes) QDS Ver.!</td></tr>
<tr><td>M11600
to</td><td>Synchronous endcoer axis command signal
(4 points × 12 axes) QDS Ver.!</td></tr>
<tr><td>M11648
to</td><td>Unusable
(32 points)</td></tr>
<tr><td>M11680
to</td><td>Output axis command signal
(10 points × 32 axes) QDS Ver.!</td></tr>
<tr><td>M12000
to</td><td>Synchronous control start signal [Rq.380]
(32 points) QDS Ver.!</td></tr>
<tr><td>M12032
to
M12063</td><td>Synchronous analysis request signal [Rq.381]
(32 points) QDS Ver.!</td></tr>
</table>

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Overall configuration (Continued)

SV13		SV22			
		Virtual mode switching method		Advanced synchronous control method	
Device No.	Purpose	Device No.	Purpose	Device No.	Purpose
M12064 to M12287	System area (224 points)	M12064 to M12287	System area (224 points)	M12064 to M12287	Unusable (224 points)

(Note-1): It can be used as a user device in the SV22 real mode only.
(Note-2): Do not set the M4000 to M5487 as a latch range in the virtual mode.
(Note-3): The cam axis command signal and smoothing clutch complete signal can be set as the optional device at the parameter.

1) Table of the axis statuses (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
M2400 to M2419	Axis 1 status	M2720 to M2739	Axis 17 status
M2420 to M2439	Axis 2 status	M2740 to M2759	Axis 18 status
M2440 to M2459	Axis 3 status	M2760 to M2779	Axis 19 status
M2460 to M2479	Axis 4 status	M2780 to M2799	Axis 20 status
M2480 to M2499	Axis 5 status	M2800 to M2819	Axis 21 status
M2500 to M2519	Axis 6 status	M2820 to M2839	Axis 22 status
M2520 to M2539	Axis 7 status	M2840 to M2859	Axis 23 status
M2540 to M2559	Axis 8 status	M2860 to M2879	Axis 24 status
M2560 to M2579	Axis 9 status	M2880 to M2899	Axis 25 status
M2580 to M2599	Axis 10 status	M2900 to M2919	Axis 26 status
M2600 to M2619	Axis 11 status	M2920 to M2939	Axis 27 status
M2620 to M2639	Axis 12 status	M2940 to M2959	Axis 28 status
M2640 to M2659	Axis 13 status	M2960 to M2979	Axis 29 status
M2660 to M2679	Axis 14 status	M2980 to M2999	Axis 30 status
M2680 to M2699	Axis 15 status	M3000 to M3019	Axis 31 status
M2700 to M2719	Axis 16 status	M3020 to M3039	Axis 32 status

• Details of each axis

Device No.	Signal name	
M2400+20n	Positioning start complete	
M2401+20n	Positioning complete	
M2402+20n	In-position	
M2403+20n	Command in-position	
M2404+20n	Speed controlling	
M2405+20n	Speed/position switching latch	
M2406+20n	Zero pass	
M2407+20n	Error detection	
M2408+20n	Servo error detection	
M2409+20n	Home position return request	
M2410+20n	Home position return complete	
M2411+20n	External signals	FLS
M2412+20n		RLS
M2413+20n		STOP
M2414+20n		DOG/CHANGE
M2415+20n	Servo ready	
M2416+20n	Torque limiting	
M2417+20n	Unusable	
M2418+20n	Virtual mode continuation operation disable warning (SV22) (Note-1)	
M2419+20n	M-code outputting	

(Note-1): It is unusable in the SV13/SV22 real mode and SV22 advanced synchronous control.

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The following device area can be used as a user device.
- Q172DSCPU : 17 axes or more
- Q172DCPU(-S1) : 9 axes or more

However, when the project of Q172DSCPU/Q172DCPU(-S1) is replaced with Q173DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

2) Table of the axis command signals (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
M3200 to M3219	Axis 1 command signal	M3520 to M3539	Axis 17 command signal
M3220 to M3239	Axis 2 command signal	M3540 to M3559	Axis 18 command signal
M3240 to M3259	Axis 3 command signal	M3560 to M3579	Axis 19 command signal
M3260 to M3279	Axis 4 command signal	M3580 to M3599	Axis 20 command signal
M3280 to M3299	Axis 5 command signal	M3600 to M3619	Axis 21 command signal
M3300 to M3319	Axis 6 command signal	M3620 to M3639	Axis 22 command signal
M3320 to M3339	Axis 7 command signal	M3640 to M3659	Axis 23 command signal
M3340 to M3359	Axis 8 command signal	M3660 to M3679	Axis 24 command signal
M3360 to M3379	Axis 9 command signal	M3680 to M3699	Axis 25 command signal
M3380 to M3399	Axis 10 command signal	M3700 to M3719	Axis 26 command signal
M3400 to M3419	Axis 11 command signal	M3720 to M3739	Axis 27 command signal
M3420 to M3439	Axis 12 command signal	M3740 to M3759	Axis 28 command signal
M3440 to M3459	Axis 13 command signal	M3760 to M3779	Axis 29 command signal
M3460 to M3479	Axis 14 command signal	M3780 to M3799	Axis 30 command signal
M3480 to M3499	Axis 15 command signal	M3800 to M3819	Axis 31 command signal
M3500 to M3519	Axis 16 command signal	M3820 to M3839	Axis 32 command signal

• Details of each axis

Device No.	SV13	SV22
M3200+20n	Stop command	Stop command
M3201+20n	Rapid stop command	Rapid stop command
M3202+20n	Forward rotation JOG start command	Forward rotation JOG start command
M3203+20n	Reverse rotation JOG start command	Reverse rotation JOG start command
M3204+20n	Complete signal OFF command	Complete signal OFF command
M3205+20n	Speed/position switching enable command	Speed/position switching enable command
M3206+20n	Unusable	Unusable
M3207+20n	Error reset command	Error reset command
M3208+20n	Servo error reset command	Servo error reset command
M3209+20n	External stop input disable at start command	External stop input disable at start command
M3210+20n	Unusable	Unusable
M3211+20n		
M3212+20n	Feed current value update command	Feed current value update command
M3213+20n	Unusable	Address clutch reference setting command (Note-1)
M3214+20n		Cam reference position setting command (Note-1)
M3215+20n	Servo OFF command	Servo OFF command
M3216+20n	Gain changing command	Gain changing command
M3217+20n	PI-PID switching command QDS	PI-PID switching command QDS
M3218+20n	Control loop changing command	Control loop changing command
M3219+20n	FIN signal	FIN signal

(Note-1): It is unusable in the SV22 real mode and SV22 advanced synchronous control.

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The following device area can be used as a user device.
- Q172DSCPU : 17 axes or more
- Q172DCPU(-S1) : 9 axes or more

However, when the project of Q172DSCPU/Q172DCPU(-S1) is replaced with Q173DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

3) Table of the virtual servomotor axis statuses (SV22 virtual mode only)

Device No.	Signal name	Device No.	Signal name
M4000 to M4019	Axis 1 status	M4320 to M4339	Axis 17 status
M4020 to M4039	Axis 2 status	M4340 to M4359	Axis 18 status
M4040 to M4059	Axis 3 status	M4360 to M4379	Axis 19 status
M4060 to M4079	Axis 4 status	M4380 to M4399	Axis 20 status
M4080 to M4099	Axis 5 status	M4400 to M4419	Axis 21 status
M4100 to M4119	Axis 6 status	M4420 to M4439	Axis 22 status
M4120 to M4139	Axis 7 status	M4440 to M4459	Axis 23 status
M4140 to M4159	Axis 8 status	M4460 to M4479	Axis 24 status
M4160 to M4179	Axis 9 status	M4480 to M4499	Axis 25 status
M4180 to M4199	Axis 10 status	M4500 to M4519	Axis 26 status
M4200 to M4219	Axis 11 status	M4520 to M4539	Axis 27 status
M4220 to M4239	Axis 12 status	M4540 to M4559	Axis 28 status
M4240 to M4259	Axis 13 status	M4560 to M4579	Axis 29 status
M4260 to M4279	Axis 14 status	M4580 to M4599	Axis 30 status
M4280 to M4299	Axis 15 status	M4600 to M4619	Axis 31 status
M4300 to M4319	Axis 16 status	M4620 to M4639	Axis 32 status

• Details of each axis

Device No.	Signal name
M4000+20n	Positioning start complete
M4001+20n	Positioning complete
M4002+20n	Unusable
M4003+20n	Command in-position
M4004+20n	Speed controlling
M4005+20n	Unusable
M4006+20n	
M4007+20n	Error detection
M4008+20n	Unusable
M4009+20n	
M4010+20n	
M4011+20n	
M4012+20n	
M4013+20n	
M4014+20n	
M4015+20n	
M4016+20n	
M4017+20n	
M4018+20n	
M4019+20n	M-code outputting

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The unused axis areas in the mechanical system program can be used as a user device.

4) Table of the virtual servomotor axis command signals (SV22 virtual mode only)

Device No.	Signal name	Device No.	Signal name
M4800 to M4819	Axis 1 command signal	M5120 to M5139	Axis 17 command signal
M4820 to M4839	Axis 2 command signal	M5140 to M5159	Axis 18 command signal
M4840 to M4859	Axis 3 command signal	M5160 to M5179	Axis 19 command signal
M4860 to M4879	Axis 4 command signal	M5180 to M5199	Axis 20 command signal
M4880 to M4899	Axis 5 command signal	M5200 to M5219	Axis 21 command signal
M4900 to M4919	Axis 6 command signal	M5220 to M5239	Axis 22 command signal
M4920 to M4939	Axis 7 command signal	M5240 to M5259	Axis 23 command signal
M4940 to M4959	Axis 8 command signal	M5260 to M5279	Axis 24 command signal
M4960 to M4979	Axis 9 command signal	M5280 to M5299	Axis 25 command signal
M4980 to M4999	Axis 10 command signal	M5300 to M5319	Axis 26 command signal
M5000 to M5019	Axis 11 command signal	M5320 to M5339	Axis 27 command signal
M5020 to M5039	Axis 12 command signal	M5340 to M5359	Axis 28 command signal
M5040 to M5059	Axis 13 command signal	M5360 to M5379	Axis 29 command signal
M5060 to M5079	Axis 14 command signal	M5380 to M5399	Axis 30 command signal
M5080 to M5099	Axis 15 command signal	M5400 to M5419	Axis 31 command signal
M5100 to M5119	Axis 16 command signal	M5420 to M5439	Axis 32 command signal

• Details of each axis

Device No.	Signal name
M4800+20n	Stop command
M4801+20n	Rapid stop command
M4802+20n	Forward rotation JOG start command
M4803+20n	Reverse rotation JOG start command
M4804+20n	Complete signal OFF command
M4805+20n	Unusable
M4806+20n	
M4807+20n	Error reset command
M4808+20n	Unusable
M4809+20n	External stop input disable at start command
M4810+20n	Unusable
M4811+20n	
M4812+20n	
M4813+20n	
M4814+20n	
M4815+20n	
M4816+20n	
M4817+20n	
M4818+20n	
M4819+20n	FIN signal

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The unused axis areas in the mechanical system program can be used as a user device.

5) Table of the synchronous encoder axis statuses (SV22 virtual mode only)

Device No.	Signal name		Device No.	Signal name	
M4640	Axis 1	Error detection	M4664	Axis 7	Error detection
M4641		External signal TREN	M4665		External signal TREN
M4642		Virtual mode continuation operation disable warning	M4666		Virtual mode continuation operation disable warning
M4643		Unusable	M4667		Unusable
M4644	Axis 2	Error detection	M4668	Axis 8	Error detection
M4645		External signal TREN	M4669		External signal TREN
M4646		Virtual mode continuation operation disable warning	M4670		Virtual mode continuation operation disable warning
M4647		Unusable	M4671		Unusable
M4648	Axis 3	Error detection	M4672	Axis 9	Error detection
M4649		External signal TREN	M4673		External signal TREN
M4650		Virtual mode continuation operation disable warning	M4674		Virtual mode continuation operation disable warning
M4651		Unusable	M4675		Unusable
M4652	Axis 4	Error detection	M4676	Axis 10	Error detection
M4653		External signal TREN	M4677		External signal TREN
M4654		Virtual mode continuation operation disable warning	M4678		Virtual mode continuation operation disable warning
M4655		Unusable	M4679		Unusable
M4656	Axis 5	Error detection	M4680	Axis 11	Error detection
M4657		External signal TREN	M4681		External signal TREN
M4658		Virtual mode continuation operation disable warning	M4682		Virtual mode continuation operation disable warning
M4659		Unusable	M4683		Unusable
M4660	Axis 6	Error detection	M4684	Axis 12	Error detection
M4661		External signal TREN	M4685		External signal TREN
M4662		Virtual mode continuation operation disable warning	M4686		Virtual mode continuation operation disable warning
M4663		Unusable	M4687		Unusable

POINT

(1) The range of axis No.1 to 8 is valid in the Q172DCPU(-S1).

(2) The device area more than 9 axes in the Q172DCPU(-S1) can be used as a user device.
However, when the project of Q172DCPU(-S1) is replaced with Q173DSCPU/ Q172DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

6) Table of the synchronous encoder axis command signals (SV22 virtual mode only)

Device No.	Signal name		Device No.	Signal name	
M5440	Axis 1	Error reset	M5464	Axis 7	Error reset
M5441		Unusable	M5465		Unusable
M5442			M5466		
M5443			M5467		
M5444	Axis 2	Error reset	M5468	Axis 8	Error reset
M5445		Unusable	M5469		Unusable
M5446			M5470		
M5447			M5471		
M5448	Axis 3	Error reset	M5472	Axis 9	Error reset
M5449		Unusable	M5473		Unusable
M5450			M5474		
M5451			M5475		
M5452	Axis 4	Error reset	M5476	Axis 10	Error reset
M5453		Unusable	M5477		Unusable
M5454			M5478		
M5455			M5479		
M5456	Axis 5	Error reset	M5480	Axis 11	Error reset
M5457		Unusable	M5481		Unusable
M5458			M5482		
M5459			M5483		
M5460	Axis 6	Error reset	M5484	Axis 12	Error reset
M5461		Unusable	M5485		Unusable
M5462			M5486		
M5463			M5487		

POINT

(1) The range of axis No.1 to 8 is valid in the Q172DCPU(-S1).
(2) The device area more than 9 axes in the Q172DCPU(-S1) can be used as a user device.
However, when the project of Q172DCPU(-S1) is replaced with Q173DSCPU/Q172DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

7) Table of the command generation axis statuses (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
M9800 to M9819	Axis 1 status	M10120 to M10139	Axis 17 status
M9820 to M9839	Axis 2 status	M10140 to M10159	Axis 18 status
M9840 to M9859	Axis 3 status	M10160 to M10179	Axis 19 status
M9860 to M9879	Axis 4 status	M10180 to M10199	Axis 20 status
M9880 to M9899	Axis 5 status	M10200 to M10219	Axis 21 status
M9900 to M9919	Axis 6 status	M10220 to M10239	Axis 22 status
M9920 to M9939	Axis 7 status	M10240 to M10259	Axis 23 status
M9940 to M9959	Axis 8 status	M10260 to M10279	Axis 24 status
M9960 to M9979	Axis 9 status	M10280 to M10299	Axis 25 status
M9980 to M9999	Axis 10 status	M10300 to M10319	Axis 26 status
M10000 to M10019	Axis 11 status	M10320 to M10339	Axis 27 status
M10020 to M10039	Axis 12 status	M10340 to M10359	Axis 28 status
M10040 to M10059	Axis 13 status	M10360 to M10379	Axis 29 status
M10060 to M10079	Axis 14 status	M10380 to M10399	Axis 30 status
M10080 to M10099	Axis 15 status	M10400 to M10419	Axis 31 status
M10100 to M10119	Axis 16 status	M10420 to M10439	Axis 32 status

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
M9800+20n	St.340	Command generation axis positioning start complete
M9801+20n	St.341	Command generation axis positioning complete
M9802+20n	—	Unusable
M9803+20n	St.342	Command generation axis command in-position
M9804+20n	St.343	Command generation axis speed controlling
M9805+20n M9806+20n	—	Unusable
M9807+20n	St.344	Command generation axis error detection
M9808+20n M9809+20n	—	Unusable
M9810+20n	St.345	Command generation axis start accept flag
M9811+20n	St.346	Command generation axis speed change accepting flag
M9812+20n	St.347	Command generation axis speed change "0" accepting flag
M9813+20n	St.348	Command generation axis automatic decelerating flag
M9814+20n M9815+20n M9816+20n M9817+20n M9818+20n	—	Unusable
M9819+20n	St.349	Command generation axis M-code outputting

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

8) Table of the command generation axis command signals (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
M10960 to M10979	Axis 1 command signal	M11280 to M11299	Axis 17 command signal
M10980 to M10999	Axis 2 command signal	M11300 to M11319	Axis 18 command signal
M11000 to M11019	Axis 3 command signal	M11320 to M11339	Axis 19 command signal
M11020 to M11039	Axis 4 command signal	M11340 to M11359	Axis 20 command signal
M11040 to M11059	Axis 5 command signal	M11360 to M11379	Axis 21 command signal
M11060 to M11079	Axis 6 command signal	M11380 to M11399	Axis 22 command signal
M11080 to M11099	Axis 7 command signal	M11400 to M11419	Axis 23 command signal
M11100 to M11119	Axis 8 command signal	M11420 to M11439	Axis 24 command signal
M11120 to M11139	Axis 9 command signal	M11440 to M11459	Axis 25 command signal
M11140 to M11159	Axis 10 command signal	M11460 to M11479	Axis 26 command signal
M11160 to M11179	Axis 11 command signal	M11480 to M11499	Axis 27 command signal
M11180 to M11199	Axis 12 command signal	M11500 to M11519	Axis 28 command signal
M11200 to M11219	Axis 13 command signal	M11520 to M11539	Axis 29 command signal
M11220 to M11239	Axis 14 command signal	M11540 to M11559	Axis 30 command signal
M11240 to M11259	Axis 15 command signal	M11560 to M11579	Axis 31 command signal
M11260 to M11279	Axis 16 command signal	M11580 to M11599	Axis 32 command signal

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
M10960+20n	Rq.341	Command generation axis stop command
M10961+20n	Rq.342	Command generation axis rapid stop command
M10962+20n	Rq.343	Command generation axis forward rotation JOG start command
M10963+20n	Rq.344	Command generation axis reverse rotation JOG start command
M10964+20n	Rq.345	Command generation axis complete signal OFF command
M10965+20n M10966+20n	—	Unusable
M10967+20n	Rq.346	Command generation axis error reset command
M10968+20n M10969+20n M10970+20n M10971+20n	—	Unusable
M10972+20n	Rq.347	Feed current value update request command
M10973+20n M10974+20n M10975+20n M10976+20n M10977+20n M10978+20n	—	Unusable
M10979+20n	Rq.348	Command generation axis FIN signal

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

9) Table of the synchronous encoder axis statuses (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name
M10440 to M10449	Axis 1 status
M10450 to M10459	Axis 2 status
M10460 to M10469	Axis 3 status
M10470 to M10479	Axis 4 status
M10480 to M10489	Axis 5 status
M10490 to M10499	Axis 6 status
M10500 to M10509	Axis 7 status
M10510 to M10519	Axis 8 status
M10520 to M10529	Axis 9 status
M10530 to M10539	Axis 10 status
M10540 to M10549	Axis 11 status
M10550 to M10559	Axis 12 status

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
M10440+10n	St.320	Synchronous encoder axis setting valid flag
M10441+10n	St.321	Synchronous encoder axis connecting valid flag
M10442+10n	St.322	Synchronous encoder axis counter enable flag
M10443+10n	St.323	Synchronous encoder axis current value setting request flag
M10444+10n	St.324	Synchronous encoder axis error detection flag
M10445+10n	—	Unusable
M10446+10n	St.325	Synchronous encoder axis control complete flag
M10447+10n	—	Unusable
M10448+10n		
M10449+10n		

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
 • Q173DSCPU/Q172DSCPU: Axis No.1 to 12 (n=0 to 11)

10) Table of the synchronous encoder axis command signal (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol	Signal name		Device No.	Symbol	Signal name	
M11600	Rq.323	Axis 1	Synchronous encoder axis error reset	M11624	Rq.323	Axis 7	Synchronous encoder axis error reset
M11601	Rq.320		Synchronous encoder axis control request	M11625	Rq.320		Synchronous encoder axis control request
M11602	Rq.324		Connection command of synchronous encoder via device/master CPU	M11626	Rq.324		Connection command of synchronous encoder via device/master CPU
M11603	—		Unusable	M11627	—		Unusable
M11604	Rq.323	Axis 2	Synchronous encoder axis error reset	M11628	Rq.323	Axis 8	Synchronous encoder axis error reset
M11605	Rq.320		Synchronous encoder axis control request	M11629	Rq.320		Synchronous encoder axis control request
M11606	Rq.324		Connection command of synchronous encoder via device/master CPU	M11630	Rq.324		Connection command of synchronous encoder via device/master CPU
M11607	—		Unusable	M11631	—		Unusable
M11608	Rq.323	Axis 3	Synchronous encoder axis error reset	M11632	Rq.323	Axis 9	Synchronous encoder axis error reset
M11609	Rq.320		Synchronous encoder axis control request	M11633	Rq.320		Synchronous encoder axis control request
M11610	Rq.324		Connection command of synchronous encoder via device/master CPU	M11634	Rq.324		Connection command of synchronous encoder via device/master CPU
M11611	—		Unusable	M11635	—		Unusable
M11612	Rq.323	Axis 4	Synchronous encoder axis error reset	M11636	Rq.323	Axis 10	Synchronous encoder axis error reset
M11613	Rq.320		Synchronous encoder axis control request	M11637	Rq.320		Synchronous encoder axis control request
M11614	Rq.324		Connection command of synchronous encoder via device/master CPU	M11638	Rq.324		Connection command of synchronous encoder via device/master CPU
M11615	—		Unusable	M11639	—		Unusable
M11616	Rq.323	Axis 5	Synchronous encoder axis error reset	M11640	Rq.323	Axis 11	Synchronous encoder axis error reset
M11617	Rq.320		Synchronous encoder axis control request	M11641	Rq.320		Synchronous encoder axis control request
M11618	Rq.324		Connection command of synchronous encoder via device/master CPU	M11642	Rq.324		Connection command of synchronous encoder via device/master CPU
M11619	—		Unusable	M11643	—		Unusable
M11620	Rq.323	Axis 6	Synchronous encoder axis error reset	M11644	Rq.323	Axis 12	Synchronous encoder axis error reset
M11621	Rq.320		Synchronous encoder axis control request	M11645	Rq.320		Synchronous encoder axis control request
M11622	Rq.324		Connection command of synchronous encoder via device/master CPU	M11646	Rq.324		Connection command of synchronous encoder via device/master CPU
M11623	—		Unusable	M11647	—		Unusable

Ver.! : Refer to Section 1.3 for the software version that supports this function.

MEMO

11) Table of the output axis statuses
(SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
M10560 to M10569	Axis 1 status	M10720 to M10729	Axis 17 status
M10570 to M10579	Axis 2 status	M10730 to M10739	Axis 18 status
M10580 to M10589	Axis 3 status	M10740 to M10749	Axis 19 status
M10590 to M10599	Axis 4 status	M10750 to M10759	Axis 20 status
M10600 to M10609	Axis 5 status	M10760 to M10769	Axis 21 status
M10610 to M10619	Axis 6 status	M10770 to M10779	Axis 22 status
M10620 to M10629	Axis 7 status	M10780 to M10789	Axis 23 status
M10630 to M10639	Axis 8 status	M10790 to M10799	Axis 24 status
M10640 to M10649	Axis 9 status	M10800 to M10809	Axis 25 status
M10650 to M10659	Axis 10 status	M10810 to M10819	Axis 26 status
M10660 to M10669	Axis 11 status	M10820 to M10829	Axis 27 status
M10670 to M10679	Axis 12 status	M10830 to M10839	Axis 28 status
M10680 to M10689	Axis 13 status	M10840 to M10849	Axis 29 status
M10690 to M10699	Axis 14 status	M10850 to M10859	Axis 30 status
M10700 to M10709	Axis 15 status	M10860 to M10869	Axis 31 status
M10710 to M10719	Axis 16 status	M10870 to M10879	Axis 32 status

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
M10560+10n	St.420	Main shaft clutch ON/OFF status
M10561+10n	St.421	Main shaft clutch smoothing status
M10562+10n	St.423	Auxiliary shaft clutch ON/OFF status
M10563+10n	St.424	Auxiliary shaft clutch smoothing status
M10564+10n	—	Unusable
M10565+10n		
M10566+10n	St.426	Control change complete
M10567+10n	—	Unusable
M10568+10n		
M10569+10n		

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
  • Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
  • Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
  However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

12) Table of the output axis command signals (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
M11680 to M11689	Axis 1 command signal	M11840 to M11849	Axis 17 command signal
M11690 to M11699	Axis 2 command signal	M11850 to M11859	Axis 18 command signal
M11700 to M11709	Axis 3 command signal	M11860 to M11869	Axis 19 command signal
M11710 to M11719	Axis 4 command signal	M11870 to M11879	Axis 20 command signal
M11720 to M11729	Axis 5 command signal	M11880 to M11889	Axis 21 command signal
M11730 to M11739	Axis 6 command signal	M11890 to M11899	Axis 22 command signal
M11740 to M11749	Axis 7 command signal	M11900 to M11909	Axis 23 command signal
M11750 to M11759	Axis 8 command signal	M11910 to M11919	Axis 24 command signal
M11760 to M11769	Axis 9 command signal	M11920 to M11929	Axis 25 command signal
M11770 to M11779	Axis 10 command signal	M11930 to M11939	Axis 26 command signal
M11780 to M11789	Axis 11 command signal	M11940 to M11949	Axis 27 command signal
M11790 to M11799	Axis 12 command signal	M11950 to M11959	Axis 28 command signal
M11800 to M11809	Axis 13 command signal	M11960 to M11969	Axis 29 command signal
M11810 to M11819	Axis 14 command signal	M11970 to M11979	Axis 30 command signal
M11820 to M11829	Axis 15 command signal	M11980 to M11989	Axis 31 command signal
M11830 to M11839	Axis 16 command signal	M11990 to M11999	Axis 32 command signal

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
M11680+10n	Rq.400	Main shaft clutch command
M11681+10n	Rq.401	Main shaft clutch control invalid command
M11682+10n	Rq.402	Main shaft clutch forced OFF command
M11683+10n	—	Unusable
M11684+10n	Rq.403	Auxiliary shaft clutch command
M11685+10n	Rq.404	Auxiliary shaft clutch control invalid command
M11686+10n	Rq.405	Auxiliary shaft clutch forced OFF command
M11687+10n	—	Unusable
M11688+10n	Rq.406	Control change request command
M11689+10n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

13) Table of the synchronous control signals
(SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol	Signal name	
M10880	St.380	Axis 1	Synchronous control
M10881		Axis 2	
M10882		Axis 3	
M10883		Axis 4	
M10884		Axis 5	
M10885		Axis 6	
M10886		Axis 7	
M10887		Axis 8	
M10888		Axis 9	
M10889		Axis 10	
M10890		Axis 11	
M10891		Axis 12	
M10892		Axis 13	
M10893		Axis 14	
M10894		Axis 15	
M10895		Axis 16	
M10896		Axis 17	
M10897		Axis 18	
M10898		Axis 19	
M10899		Axis 20	
M10900		Axis 21	
M10901		Axis 22	
M10902		Axis 23	
M10903		Axis 24	
M10904		Axis 25	
M10905		Axis 26	
M10906		Axis 27	
M10907		Axis 28	
M10908		Axis 29	
M10909		Axis 30	
M10910		Axis 31	
M10911		Axis 32	

POINT

(1) The range of axis No.1 to 16 is valid in the Q172DSCPU.
(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

14) Table of the synchronous analysis complete signals (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol	Signal name	
M10912	St.381	Axis 1	Synchronous analysis complete
M10913		Axis 2	
M10914		Axis 3	
M10915		Axis 4	
M10916		Axis 5	
M10917		Axis 6	
M10918		Axis 7	
M10919		Axis 8	
M10920		Axis 9	
M10921		Axis 10	
M10922		Axis 11	
M10923		Axis 12	
M10924		Axis 13	
M10925		Axis 14	
M10926		Axis 15	
M10927		Axis 16	
M10928		Axis 17	
M10929		Axis 18	
M10930		Axis 19	
M10931		Axis 20	
M10932		Axis 21	
M10933		Axis 22	
M10934		Axis 23	
M10935		Axis 24	
M10936		Axis 25	
M10937		Axis 26	
M10938		Axis 27	
M10939		Axis 28	
M10940		Axis 29	
M10941		Axis 30	
M10942		Axis 31	
M10943		Axis 32	

POINT

(1) The range of axis No.1 to 16 is valid in the Q172DSCPU.
(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

15) Table of the synchronous control start signals (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol		Signal name
M12000	Rq.380	Axis 1	Synchronous control start
M12001		Axis 2	
M12002		Axis 3	
M12003		Axis 4	
M12004		Axis 5	
M12005		Axis 6	
M12006		Axis 7	
M12007		Axis 8	
M12008		Axis 9	
M12009		Axis 10	
M12010		Axis 11	
M12011		Axis 12	
M12012		Axis 13	
M12013		Axis 14	
M12014		Axis 15	
M12015		Axis 16	
M12016		Axis 17	
M12017		Axis 18	
M12018		Axis 19	
M12019		Axis 20	
M12020		Axis 21	
M12021		Axis 22	
M12022		Axis 23	
M12023		Axis 24	
M12024		Axis 25	
M12025		Axis 26	
M12026		Axis 27	
M12027		Axis 28	
M12028		Axis 29	
M12029		Axis 30	
M12030		Axis 31	
M12031		Axis 32	

POINT

(1) The range of axis No.1 to 16 is valid in the Q172DSCPU.
(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

16) Table of the synchronous analysis request signals (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol	Signal name	
M12032	Rq.381	Axis 1	Synchronous analysis request
M12033		Axis 2	
M12034		Axis 3	
M12035		Axis 4	
M12036		Axis 5	
M12037		Axis 6	
M12038		Axis 7	
M12039		Axis 8	
M12040		Axis 9	
M12041		Axis 10	
M12042		Axis 11	
M12043		Axis 12	
M12044		Axis 13	
M12045		Axis 14	
M12046		Axis 15	
M12047		Axis 16	
M12048		Axis 17	
M12049		Axis 18	
M12050		Axis 19	
M12051		Axis 20	
M12052		Axis 21	
M12053		Axis 22	
M12054		Axis 23	
M12055		Axis 24	
M12056		Axis 25	
M12057		Axis 26	
M12058		Axis 27	
M12059		Axis 28	
M12060		Axis 29	
M12061		Axis 30	
M12062		Axis 31	
M12063		Axis 32	

POINT

(1) The range of axis No.1 to 16 is valid in the Q172DSCPU.
(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

17) Table of the common devices (SV13/SV22)

<table>
<tr><th colspan="3">SV13</th><th colspan="3">SV22</th><th rowspan="2">Refresh cycle</th><th rowspan="2">Fetch cycle</th><th rowspan="2">Signal direction</th><th rowspan="2">Remark (Note-7)</th></tr>
<tr><th>Device No.</th><th colspan="2">Signal name</th><th>Device No.</th><th colspan="2">Signal name</th></tr>
<tr><td>M2000</td><td colspan="2">PLC ready flag</td><td>M2000</td><td colspan="2">PLC ready flag</td><td></td><td>Main cycle</td><td>Command signal</td><td>M3072</td></tr>
<tr><td>M2001
to
M2032</td><td>Axis1
to
Axis32</td><td>Start accept flag (32 points)</td><td>M2001
to
M2032</td><td>Axis1
to
Axis32</td><td>Start accept flag (32 points)</td><td>Operation cycle</td><td></td><td>Status signal (Note-1, 2, 3, 4)</td><td></td></tr>
<tr><td>M2033
M2034</td><td colspan="2">Unusable (2 points)</td><td>M2033
M2034</td><td colspan="2">Unusable (2 points)</td><td>—</td><td>—</td><td>—</td><td></td></tr>
<tr><td>M2035</td><td colspan="2">Motion error history clear request flag</td><td>M2035</td><td colspan="2">Motion error history clear request flag</td><td></td><td>Main cycle</td><td>Command signal</td><td>M3080</td></tr>
<tr><td>M2036
M2037</td><td colspan="2">Unusable (2 points)</td><td>M2036
M2037</td><td colspan="2">Unusable (2 points)</td><td>—</td><td>—</td><td>—</td><td></td></tr>
<tr><td>M2038</td><td colspan="2">Motion SFC debugging flag</td><td>M2038</td><td colspan="2">Motion SFC debugging flag</td><td>At debug mode transition</td><td rowspan="2"></td><td rowspan="2">Status signal</td><td></td></tr>
<tr><td>M2039</td><td colspan="2">Motion error detection flag</td><td>M2039</td><td colspan="2">Motion error detection flag</td><td>Immediate</td><td></td></tr>
<tr><td>M2040</td><td colspan="2">Speed switching point specified flag</td><td>M2040</td><td colspan="2">Speed switching point specified flag</td><td></td><td>At start</td><td>Command signal</td><td>M3073</td></tr>
<tr><td>M2041</td><td colspan="2">System setting error flag</td><td>M2041</td><td colspan="2">System setting error flag</td><td>Operation cycle</td><td></td><td>Status signal</td><td></td></tr>
<tr><td>M2042</td><td colspan="2">All axes servo ON command</td><td>M2042</td><td colspan="2">All axes servo ON command</td><td rowspan="2"></td><td>Operation cycle</td><td rowspan="2">Command signal</td><td>M3074</td></tr>
<tr><td>M2043</td><td colspan="2" rowspan="4">Unusable (4 points)</td><td>M2043</td><td colspan="2">Real mode /virtual mode switching request (Note-5)</td><td>At virtual mode transition</td><td>M3075</td></tr>
<tr><td>M2044</td><td>M2044</td><td colspan="2">Real mode/virtual mode switching status (Note-5)</td><td rowspan="3">At virtual mode transition</td><td rowspan="4"></td><td rowspan="4">Status signal</td><td rowspan="4"></td></tr>
<tr><td>M2045</td><td>M2045</td><td colspan="2">Real mode/virtual mode switching error detection flag (Note-5)</td></tr>
<tr><td>M2046</td><td>M2046</td><td colspan="2">Out-of-sync warning (Note-5)</td></tr>
<tr><td>M2047</td><td colspan="2">Motion slot fault detection flag</td><td>M2047</td><td colspan="2">Motion slot fault detection flag</td><td>Operation cycle</td></tr>
<tr><td>M2048</td><td colspan="2">JOG operation simultaneous start command</td><td>M2048</td><td colspan="2">JOG operation simultaneous start command</td><td></td><td>Main cycle</td><td>Command signal</td><td>M3076</td></tr>
<tr><td>M2049</td><td colspan="2">All axes servo ON accept flag</td><td>M2049</td><td colspan="2">All axes servo ON accept flag</td><td>Operation cycle</td><td></td><td>Status signal</td><td></td></tr>
<tr><td>M2050</td><td colspan="2">Unusable</td><td>M2050</td><td colspan="2">Unusable</td><td>—</td><td>—</td><td>—</td><td></td></tr>
<tr><td>M2051</td><td colspan="2">Manual pulse generator 1 enable flag</td><td>M2051</td><td colspan="2">Manual pulse generator 1 enable flag</td><td rowspan="3"></td><td rowspan="3">Main cycle</td><td rowspan="3">Command signal</td><td>M3077</td></tr>
<tr><td>M2052</td><td colspan="2">Manual pulse generator 2 enable flag</td><td>M2052</td><td colspan="2">Manual pulse generator 2 enable flag</td><td>M3078</td></tr>
<tr><td>M2053</td><td colspan="2">Manual pulse generator 3 enable flag</td><td>M2053</td><td colspan="2">Manual pulse generator 3 enable flag</td><td>M3079</td></tr>
<tr><td>M2054</td><td colspan="2">Operation cycle over flag</td><td>M2054</td><td colspan="2">Operation cycle over flag</td><td>Operation cycle</td><td></td><td>Status signal</td><td rowspan="3"></td></tr>
<tr><td>M2055
to
M2060</td><td colspan="2">Unusable (6 points)</td><td>M2055
to
M2060</td><td colspan="2">Unusable (6 points)</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>M2061
to
M2092</td><td>Axis 1
to
Axis 32</td><td>Speed change accepting flag (32 axes)</td><td>M2061
to
M2092</td><td>Axis 1
to
Axis 32</td><td>Speed change accepting flag (32 axes)</td><td>Operation cycle</td><td></td><td>Status signal (Note-1, 2, 3, 4)</td></tr>
</table>

Table of the common devices (SV13/SV22) (continued)

SV13 Device No.	SV13 Signal name	SV22 Device No.	SV22 Signal name	Refresh cycle	Fetch cycle	Signal direction	Remark (Note-7)
M2093 to M2127	Unusable (35 points)	M2093 to M2100	Unusable (8 points)	—	—	—	
		M2101 to M2112	Axis 1 to Axis 12 Synchronous encoder current value changing flag (Note-5, 6) (12 axes)	Operation cycle		Status signal (Note-2, 4)	
		M2113 to M2127	Unusable (15 points)	—	—	—	
M2128 to M2159	Axis 1 to Axis 32 Automatic decelerating flag (32 axes)	M2128 to M2159	Axis 1 to Axis 32 Automatic decelerating flag (32 axes)	Operation cycle		Status signal (Note-1, 2, 3, 4)	
M2160 to M2239	Unusable (80 points)	M2160 to M2223	Unusable (Note-8) (64 points)	—	—	—	
		M2224 to M2239	Unusable (16 points)				
M2240 to M2271	Axis 1 to Axis 32 Speed change "0" accepting flag (32 axes)	M2240 to M2271	Axis 1 to Axis 32 Speed change "0" accepting flag (32 axes)	Operation cycle		Status signal (Note-1, 2, 3, 4)	
M2272 to M2303	Axis 1 to Axis 32 Control loop monitor status (32 axes)	M2272 to M2303	Axis 1 to Axis 32 Control loop monitor status (32 axes)				
M2304 to M2319	Unusable (16 points)	M2304 to M2319	Unusable (16 points)	—	—	—	

(Note-1): The range of axis No.1 to 16 is valid in the Q172DSCPU.
(Note-2): The range of axis No.1 to 8 is valid in the Q172DCPU(-S1).
(Note-3): Device area of 17 axes or more is unusable in the Q172DSCPU.
(Note-4): Device area of 9 axes or more is unusable in the Q172DCPU(-S1).
(Note-5): It is unusable in the SV22 advanced synchronous control.
(Note-6): It is unusable in the real mode. (It can be used in the real mode for the version (Refer to Section 1.3) that supports "synchronous encoder current value in real mode".)
(Note-7): It can also be ordered the device of a remark column.
(Note-8): These devices can be used as the clutch statuses.
The clutch status can also be set as the optional device at the clutch parameter.
Refer to Section 7.2.2 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for details.

18) Table of the common devices (Command signal) (SV13/SV22)

Device No.	Signal name	Refresh cycle	Fetch cycle	Signal direction	Remark (Note-1), (Note-2)
M3072	PLC ready flag		Main cycle	Command signal	M2000
M3073	Speed switching point specified flag		At start		M2040
M3074	All axes servo ON command		Operation cycle		M2042
M3075	Real mode/virtual mode change request (SV22) (Note-3)		At virtual mode transition		M2043
M3076	JOG operation simultaneous start command		Main cycle		M2048
M3077	Manual pulse generator 1 enable flag				M2051
M3078	Manual pulse generator 2 enable flag				M2052
M3079	Manual pulse generator 3 enable flag				M2053
M3080	Motion error history clear request flag				M2035
M3081 to M3135	Unusable (Note-4) (55 points)	—	—	—	—

(Note-1): The state of a device is not in agreement when the device of a remark column is turned ON/OFF directly. In addition, when the request from a data register and the request from the above device are performed simultaneously, the request from the above device becomes effective.

(Note-2): It can also be ordered the device of a remark column.

(Note-3): It is unusable in the SV22 advanced synchronous control.

(Note-4): Do not use it as a user device. It can be used as a device that performs automatic refresh because of area for the reserve of command signal.

POINT

The device of a remark column turns ON by OFF to ON of the above device, and turns OFF by ON to OFF of the above device.
The command signal cannot be turned ON/OFF by the PLC CPU in the automatic refresh because the statuses and commands are mixed together in M2000 to M2053. Use the above devices in the case.
And, it can also be turned ON/OFF by the data register.
Refer to Section 3.2.3 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" or Section 4.2.8 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for details.

MEMO

(b) Table of the data registers
• Overall configuration

SV13		SV22			
		Virtual mode switching method		Advanced synchronous control method	
Device No.	Purpose	Device No.	Purpose	Device No.	Purpose
D0 to	Axis monitor device (20 points × 32 axes)	D0 to	Axis monitor device (20 points × 32 axes) Real mode : Each axis Virtual mode : Output module	D0 to	Axis monitor device (20 points × 32 axes)
D640 to	Control change register (2 points × 32 axes)	D640 to	Control change register (2 points × 32 axes)	D640 to	Control change register (2 points × 32 axes)
D704 to	Common device (Command signal) (54 points)	D704 to	Common device (Command signal) (54 points)	D704 to	Common device (Command signal) (54 points)
D758 to	Unusable (42 points)	D758 to	Unusable (42 points)	D758 to	Unusable (42 points)
D800 to D8191	User device (7392 points)	D800 to	Virtual servomotor axis monitor device (Note-1) (10 points × 32 axes) (Mechanical system setting axis only)	D800 to D8191	User device (7392 points)
		D1120 to	Synchronous encoder axis monitor device (10 points × 12 axes)		
		D1240 to	Cam axis monitor device (Note-1) (10 points × 32 axes)		
		D1560 to D8191	User device (6632 points)		
				D8192 to	User device (2048 points) QDS Ver.!
				D10240 to	System area (2040 points) QDS Ver.!
				D12280 to	Servo input axis monitor device (10 points × 32 axes) QDS Ver.!
				D12600 to	Command generation axis monitor device (20 points × 32 axes) QDS Ver.!
				D13240 to	Synchronous encoder axis monitor device (20 points × 12 axes) QDS Ver.!
				D13480 to	Unusable (120 points) QDS Ver.!
				D13600 to	Output axis monitor device (30 points × 32 axes) QDS Ver.!
				D14560 to D14599	Unusable (40 points) QDS Ver.!

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Overall configuration (Continued)

SV13		SV22			
		Virtual mode switching method		Advanced synchronous control method	
Device No.	Purpose	Device No.	Purpose	Device No.	Purpose
				D14600 to	Servo input axis control device (2 points × 32 axes) QDS Ver.!
				D14664 to	Unusable (16 points) QDS Ver.!
				D14680 to	Command generation axis control device (4 points × 32 axes) QDS Ver.!
				D14808 to	Unusable (12 points) QDS Ver.!
				D14820 to	Synchronous encoder axis control device (10 points × 12 axes) QDS Ver.!
				D14940 to	Unusable (60 points) QDS Ver.!
				D15000 to	Output axis control device (150 points × 32 axes) QDS Ver.!
				D19800 to D19823	Unusable (24 points) QDS Ver.!

(Note-1): It can be used as a user device in the SV22 real mode only.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

1) Table of each axis monitor devices (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
D0 to D19	Axis 1 monitor device	D320 to D339	Axis 17 monitor device
D20 to D39	Axis 2 monitor device	D340 to D359	Axis 18 monitor device
D40 to D59	Axis 3 monitor device	D360 to D379	Axis 19 monitor device
D60 to D79	Axis 4 monitor device	D380 to D399	Axis 20 monitor device
D80 to D99	Axis 5 monitor device	D400 to D419	Axis 21 monitor device
D100 to D119	Axis 6 monitor device	D420 to D439	Axis 22 monitor device
D120 to D139	Axis 7 monitor device	D440 to D459	Axis 23 monitor device
D140 to D159	Axis 8 monitor device	D460 to D479	Axis 24 monitor device
D160 to D179	Axis 9 monitor device	D480 to D499	Axis 25 monitor device
D180 to D199	Axis 10 monitor device	D500 to D519	Axis 26 monitor device
D200 to D219	Axis 11 monitor device	D520 to D539	Axis 27 monitor device
D220 to D239	Axis 12 monitor device	D540 to D559	Axis 28 monitor device
D240 to D259	Axis 13 monitor device	D560 to D579	Axis 29 monitor device
D260 to D279	Axis 14 monitor device	D580 to D599	Axis 30 monitor device
D280 to D299	Axis 15 monitor device	D600 to D619	Axis 31 monitor device
D300 to D319	Axis 16 monitor device	D620 to D639	Axis 32 monitor device

• Details of each axis

Device No.	Signal name		Signal direction
	SV13/SV22 real mode/ SV22 advanced synchronous control	SV22 virtual mode	
D0+20n D1+20n	Feed current value	Feed current value/roller cycle speed	Monitor device
D2+20n D3+20n	Real current value	Real current value	
D4+20n D5+20n	Deviation counter value	Deviation counter value	
D6+20n	Minor error code	Minor error code	
D7+20n	Major error code	Major error code	
D8+20n	Servo error code	Servo error code	
D9+20n	Home position return re-travel value	—	
D10+20n D11+20n	Travel value after proximity dog ON	—	
D12+20n	Execute program No.	—	
D13+20n	M-code	—	
D14+20n	Torque limit value	Torque limit value	
D15+20n	Data set pointer for constant-speed control	—	
D16+20n D17+20n	Unusable (Note-1)	Unusable (Note-1)	—
D18+20n D19+20n	Real current value at stop input	—	Monitor device

(Note-1): It can be used as the travel value change register.
The travel value change register can be set to the device optionally in the servo program.
Refer to Section 6.15 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for details.

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The following device area can be used as a user device.
- Q172DSCPU : 17 axes or more
- Q172DCPU(-S1) : 9 axes or more

However, when the project of Q172DSCPU/Q172DCPU(-S1) is replaced with Q173DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

2) Table of the control change registers (SV13/SV22)

Device No.	Signal name	Device No.	Signal name
D640 D641	Axis 1 JOG speed setting register	D672 D673	Axis 17 JOG speed setting register
D642 D643	Axis 2 JOG speed setting register	D674 D675	Axis 18 JOG speed setting register
D644 D645	Axis 3 JOG speed setting register	D676 D677	Axis 19 JOG speed setting register
D646 D647	Axis 4 JOG speed setting register	D678 D679	Axis 20 JOG speed setting register
D648 D649	Axis 5 JOG speed setting register	D680 D681	Axis 21 JOG speed setting register
D650 D651	Axis 6 JOG speed setting register	D682 D683	Axis 22 JOG speed setting register
D652 D653	Axis 7 JOG speed setting register	D684 D685	Axis 23 JOG speed setting register
D654 D655	Axis 8 JOG speed setting register	D686 D687	Axis 24 JOG speed setting register
D656 D657	Axis 9 JOG speed setting register	D688 D689	Axis 25 JOG speed setting register
D658 D659	Axis 10 JOG speed setting register	D690 D691	Axis 26 JOG speed setting register
D660 D661	Axis 11 JOG speed setting register	D692 D693	Axis 27 JOG speed setting register
D662 D663	Axis 12 JOG speed setting register	D694 D695	Axis 28 JOG speed setting register
D664 D665	Axis 13 JOG speed setting register	D696 D697	Axis 29 JOG speed setting register
D666 D667	Axis 14 JOG speed setting register	D698 D699	Axis 30 JOG speed setting register
D668 D669	Axis 15 JOG speed setting register	D700 D701	Axis 31 JOG speed setting register
D670 D671	Axis 16 JOG speed setting register	D702 D703	Axis 32 JOG speed setting register

POINT

(1) The following range is valid.
- Q172DSCPU : Axis No.1 to 16
- Q172DCPU(-S1) : Axis No.1 to 8

(2) The following device area can be used as a user device.
- Q172DSCPU : 17 axes or more
- Q172DCPU(-S1) : 9 axes or more

However, when the project of Q172DSCPU/Q172DCPU(-S1) is replaced with Q173DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

MEMO

3) Table of the virtual servomotor axis monitor devices (SV22 virtual mode only)

Device No.	Signal name	Device No.	Signal name
D800 to D809	Axis 1 monitor device	D960 to D969	Axis 17 monitor device
D810 to D819	Axis 2 monitor device	D970 to D979	Axis 18 monitor device
D820 to D829	Axis 3 monitor device	D980 to D989	Axis 19 monitor device
D830 to D839	Axis 4 monitor device	D990 to D999	Axis 20 monitor device
D840 to D849	Axis 5 monitor device	D1000 to D1009	Axis 21 monitor device
D850 to D859	Axis 6 monitor device	D1010 to D1019	Axis 22 monitor device
D860 to D869	Axis 7 monitor device	D1020 to D1029	Axis 23 monitor device
D870 to D879	Axis 8 monitor device	D1030 to D1039	Axis 24 monitor device
D880 to D889	Axis 9 monitor device	D1040 to D1049	Axis 25 monitor device
D890 to D899	Axis 10 monitor device	D1050 to D1059	Axis 26 monitor device
D900 to D909	Axis 11 monitor device	D1060 to D1069	Axis 27 monitor device
D910 to D919	Axis 12 monitor device	D1070 to D1079	Axis 28 monitor device
D920 to D929	Axis 13 monitor device	D1080 to D1089	Axis 29 monitor device
D930 to D939	Axis 14 monitor device	D1090 to D1099	Axis 30 monitor device
D940 to D949	Axis 15 monitor device	D1100 to D1109	Axis 31 monitor device
D950 to D959	Axis 16 monitor device	D1110 to D1119	Axis 32 monitor device

• Details of each axis

Device No.	Signal name
D800+10n D801+10n	Feed current value
D802+10n	Minor error code
D803+10n	Major error code
D804+10n	Execute program No.
D805+10n	M-code
D806+10n D807+10n	Current value after virtual servomotor axis main shaft's differential gear
D808+10n	Error search output axis No.
D809+10n	Data set pointer for constant-speed control

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The unused axis areas in the mechanical system program can be used as a user device.

4) Table of the synchronous encoder axis monitor devices (SV22 virtual mode only)

Device No.	Signal name
D1120 to D1129	Axis 1 monitor device
D1130 to D1139	Axis 2 monitor device
D1140 to D1149	Axis 3 monitor device
D1150 to D1159	Axis 4 monitor device
D1160 to D1169	Axis 5 monitor device
D1170 to D1179	Axis 6 monitor device
D1180 to D1189	Axis 7 monitor device
D1190 to D1199	Axis 8 monitor device
D1200 to D1209	Axis 9 monitor device
D1210 to D1219	Axis 10 monitor device
D1220 to D1229	Axis 11 monitor device
D1230 to D1239	Axis 12 monitor device

• Details of each axis

Device No.	Signal name
D1120+10n D1121+10n	Current value
D1122+10n	Minor error code
D1123+10n	Major error code
D1124+10n D1125+10n	Unusable
D1126+10n D1127+10n	Current value after synchronous encoder axis main shaft's differential gear
D1128+10n	Error search output axis No.
D1129+10n	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q172DSCPU/Q173DCPU(-S1) : Axis No.1 to 12 (n=0 to 11)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The device area more than 9 axes in the Q172DCPU(-S1) can be used as a user device.
However, when the project of Q172DCPU(-S1) is replaced with Q173DSCPU/Q172DSCPU/Q173DCPU(-S1), this area cannot be used as a user device.

5) Table of the cam axis monitor devices (SV22 virtual mode only)

Device No.	Signal name	Device No.	Signal name
D1240 to D1249	Axis 1 monitor device	D1400 to D1409	Axis 17 monitor device
D1250 to D1259	Axis 2 monitor device	D1410 to D1419	Axis 18 monitor device
D1260 to D1269	Axis 3 monitor device	D1420 to D1429	Axis 19 monitor device
D1270 to D1279	Axis 4 monitor device	D1430 to D1439	Axis 20 monitor device
D1280 to D1289	Axis 5 monitor device	D1440 to D1449	Axis 21 monitor device
D1290 to D1299	Axis 6 monitor device	D1450 to D1459	Axis 22 monitor device
D1300 to D1309	Axis 7 monitor device	D1460 to D1469	Axis 23 monitor device
D1310 to D1319	Axis 8 monitor device	D1470 to D1479	Axis 24 monitor device
D1320 to D1329	Axis 9 monitor device	D1480 to D1489	Axis 25 monitor device
D1330 to D1339	Axis 10 monitor device	D1490 to D1499	Axis 26 monitor device
D1340 to D1349	Axis 11 monitor device	D1500 to D1509	Axis 27 monitor device
D1350 to D1359	Axis 12 monitor device	D1510 to D1519	Axis 28 monitor device
D1360 to D1369	Axis 13 monitor device	D1520 to D1529	Axis 29 monitor device
D1370 to D1379	Axis 14 monitor device	D1530 to D1539	Axis 30 monitor device
D1380 to D1389	Axis 15 monitor device	D1540 to D1549	Axis 31 monitor device
D1390 to D1399	Axis 16 monitor device	D1550 to D1559	Axis 32 monitor device

• Details of each axis

Device No.	Signal name
D1240+10n	Unusable
D1241+10n	Execute cam No.
D1242+10n D1243+10n	Execute stroke amount
D1244+10n D1245+10n	Current value within 1 cam shaft revolution
D1246+10n D1247+10n D1248+10n D1249+10n	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q173DCPU(-S1) : Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU : Axis No.1 to 16 (n=0 to 15)
- Q172DCPU(-S1) : Axis No.1 to 8 (n=0 to 7)

(2) The unused axis areas in the mechanical system program can be used as a user device.

6) Table of the servo input axis monitor devices
(SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
D12280 to D12289	Axis 1 monitor device	D12440 to D12449	Axis 17 monitor device
D12290 to D12299	Axis 2 monitor device	D12450 to D12459	Axis 18 monitor device
D12300 to D12309	Axis 3 monitor device	D12460 to D12469	Axis 19 monitor device
D12310 to D12319	Axis 4 monitor device	D12470 to D12479	Axis 20 monitor device
D12320 to D12329	Axis 5 monitor device	D12480 to D12489	Axis 21 monitor device
D12330 to D12339	Axis 6 monitor device	D12490 to D12499	Axis 22 monitor device
D12340 to D12349	Axis 7 monitor device	D12500 to D12509	Axis 23 monitor device
D12350 to D12359	Axis 8 monitor device	D12510 to D12519	Axis 24 monitor device
D12360 to D12369	Axis 9 monitor device	D12520 to D12529	Axis 25 monitor device
D12370 to D12379	Axis 10 monitor device	D12530 to D12539	Axis 26 monitor device
D12380 to D12389	Axis 11 monitor device	D12540 to D12549	Axis 27 monitor device
D12390 to D12399	Axis 12 monitor device	D12550 to D12559	Axis 28 monitor device
D12400 to D12409	Axis 13 monitor device	D12560 to D12569	Axis 29 monitor device
D12410 to D12419	Axis 14 monitor device	D12570 to D12579	Axis 30 monitor device
D12420 to D12429	Axis 15 monitor device	D12580 to D12589	Axis 31 monitor device
D12430 to D12439	Axis 16 monitor device	D12590 to D12599	Axis 32 monitor device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D12280+10n D12281+10n	Md.300	Servo input axis current value
D12282+10n D12283+10n	Md.301	Servo input axis speed
D12284+10n D12285+10n	Md.302	Servo input axis phase compensation amount
D12286+10n D12287+10n	Md.303	Servo input axis rotation direction restriction amount
D12288+10n D12289+10n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area can be used more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

7) Table of the servo input axis control devices
(SV22 advanced synchronous control only) QDS Ver.!

Device No.	Symbol	Signal name	Device No.	Symbol	Signal name
D14600 D14601	Pr.302	Axis 1 servo input axis phase compensation advance time	D14632 D14633	Pr.302	Axis 17 servo input axis phase compensation advance time
D14602 D14603		Axis 2 servo input axis phase compensation advance time	D14634 D14635		Axis 18 servo input axis phase compensation advance time
D14604 D14605		Axis 3 servo input axis phase compensation advance time	D14636 D14637		Axis 19 servo input axis phase compensation advance time
D14606 D14607		Axis 4 servo input axis phase compensation advance time	D14638 D14639		Axis 20 servo input axis phase compensation advance time
D14608 D14609		Axis 5 servo input axis phase compensation advance time	D14640 D14641		Axis 21 servo input axis phase compensation advance time
D14610 D14611		Axis 6 servo input axis phase compensation advance time	D14642 D14643		Axis 22 servo input axis phase compensation advance time
D14612 D14613		Axis 7 servo input axis phase compensation advance time	D14644 D14645		Axis 23 servo input axis phase compensation advance time
D14614 D14615		Axis 8 servo input axis phase compensation advance time	D14646 D14647		Axis 24 servo input axis phase compensation advance time
D14616 D14617		Axis 9 servo input axis phase compensation advance time	D14648 D14649		Axis 25 servo input axis phase compensation advance time
D14618 D14619		Axis 10 servo input axis phase compensation advance time	D14650 D14651		Axis 26 servo input axis phase compensation advance time
D14620 D14621		Axis 11 servo input axis phase compensation advance time	D14652 D14653		Axis 27 servo input axis phase compensation advance time
D14622 D14623		Axis 12 servo input axis phase compensation advance time	D14654 D14655		Axis 28 servo input axis phase compensation advance time
D14624 D14625		Axis 13 servo input axis phase compensation advance time	D14656 D14657		Axis 29 servo input axis phase compensation advance time
D14626 D14627		Axis 14 servo input axis phase compensation advance time	D14658 D14659		Axis 30 servo input axis phase compensation advance time
D14628 D14629		Axis 15 servo input axis phase compensation advance time	D14660 D14661		Axis 31 servo input axis phase compensation advance time
D14630 D14631		Axis 16 servo input axis phase compensation advance time	D14662 D14663		Axis 32 servo input axis phase compensation advance time

POINT

(1) The range of axis No.1 to 16 is valid in the Q172DSCPU.
(2) The device area more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

MEMO

8) Table of the command generation axis monitor devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
D12600 to D12619	Axis 1 monitor device	D12920 to D12939	Axis 17 monitor device
D12620 to D12639	Axis 2 monitor device	D12940 to D12959	Axis 18 monitor device
D12640 to D12659	Axis 3 monitor device	D12960 to D12979	Axis 19 monitor device
D12660 to D12679	Axis 4 monitor device	D12980 to D12999	Axis 20 monitor device
D12680 to D12699	Axis 5 monitor device	D13000 to D13019	Axis 21 monitor device
D12700 to D12719	Axis 6 monitor device	D13020 to D13039	Axis 22 monitor device
D12720 to D12739	Axis 7 monitor device	D13040 to D13059	Axis 23 monitor device
D12740 to D12759	Axis 8 monitor device	D13060 to D13079	Axis 24 monitor device
D12760 to D12779	Axis 9 monitor device	D13080 to D13099	Axis 25 monitor device
D12780 to D12799	Axis 10 monitor device	D13100 to D13119	Axis 26 monitor device
D12800 to D12819	Axis 11 monitor device	D13120 to D13139	Axis 27 monitor device
D12820 to D12839	Axis 12 monitor device	D13140 to D13159	Axis 28 monitor device
D12840 to D12859	Axis 13 monitor device	D13160 to D13179	Axis 29 monitor device
D12860 to D12879	Axis 14 monitor device	D13180 to D13199	Axis 30 monitor device
D12880 to D12899	Axis 15 monitor device	D13200 to D13219	Axis 31 monitor device
D12900 to D12919	Axis 16 monitor device	D13220 to D13239	Axis 32 monitor device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D12600+20n D12601+20n	Md.340	Command generation axis feed current value
D12602+20n	Md.341	Command generation axis minor error code
D12603+20n	Md.342	Command generation axis major error code
D12604+20n	Md.343	Command generation axis execute program No.
D12605+20n	Md.344	Command generation axis M-code
D12606+20n D12607+20n	Md.345	Command generation axis accumulative current value
D12608+20n	—	Unusable
D12609+20n	Md.346	Command generation axis data set pointer for constant-speed control
D12610+20n D12611+20n	Md.347	Command generation axis current value per cycle
D12612+20n D12613+20n	Md.348	Command generation axis command speed
D12614+20n D12615+20n D12616+20n D12617+20n D12618+20n D12619+20n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area can be used more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

9) Table of the command generation axis control devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
D14680 to D14683	Axis 1 control device	D14744 to D14747	Axis 17 control device
D14684 to D14687	Axis 2 control device	D14748 to D14751	Axis 18 control device
D14688 to D14691	Axis 3 control device	D14752 to D14755	Axis 19 control device
D14692 to D14695	Axis 4 control device	D14756 to D14759	Axis 20 control device
D14696 to D14699	Axis 5 control device	D14760 to D14763	Axis 21 control device
D14700 to D14703	Axis 6 control device	D14764 to D14767	Axis 22 control device
D14704 to D14707	Axis 7 control device	D14768 to D14771	Axis 23 control device
D14708 to D14711	Axis 8 control device	D14772 to D14775	Axis 24 control device
D14712 to D14715	Axis 9 control device	D14776 to D14779	Axis 25 control device
D14716 to D14719	Axis 10 control device	D14780 to D14783	Axis 26 control device
D14720 to D14723	Axis 11 control device	D14784 to D14787	Axis 27 control device
D14724 to D14727	Axis 12 control device	D14788 to D14791	Axis 28 control device
D14728 to D14731	Axis 13 control device	D14792 to D14795	Axis 29 control device
D14732 to D14735	Axis 14 control device	D14796 to D14799	Axis 30 control device
D14736 to D14739	Axis 15 control device	D14800 to D14803	Axis 31 control device
D14740 to D14743	Axis 16 control device	D14804 to D14807	Axis 32 control device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D14680+4n D14681+4n	Cd.340	Command generation axis JOG speed setting
D14682+4n	Pr.348	Command generation axis JOG operation parameter block setting
D14683+4n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
- Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area can be used more than 17 axes in the Q172DSCPU can be used as a user device.
However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

10) Table of the synchronous encoder axis monitor devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name
D13240 to D13259	Axis 1 monitor device
D13260 to D13279	Axis 2 monitor device
D13280 to D13299	Axis 3 monitor device
D13300 to D13319	Axis 4 monitor device
D13320 to D13339	Axis 5 monitor device
D13340 to D13359	Axis 6 monitor device
D13360 to D13379	Axis 7 monitor device
D13380 to D13399	Axis 8 monitor device
D13400 to D13419	Axis 9 monitor device
D13420 to D13439	Axis 10 monitor device
D13440 to D13459	Axis 11 monitor device
D13460 to D13479	Axis 12 monitor device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D13240+20n D13241+20n	Md.320	Synchronous encoder axis current value
D13242+20n D13243+20n	Md.321	Synchronous encoder axis current value per cycle
D13244+20n D13245+20n	Md.322	Synchronous encoder axis speed
D13246+20n D13247+20n	Md.323	Synchronous encoder axis phase compensation amount
D13248+20n D13249+20n	Md.324	Synchronous encoder axis rotation direction restriction amount
D13250+20n	Md.327	Synchronous encoder axis minor error code
D13251+20n	Md.326	Synchronous encoder axis major error code
D13252+20n D13253+20n D13254+20n D13255+20n D13256+20n D13257+20n D13258+20n D13259+20n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
• Q173DSCPU/Q172DSCPU: Axis No.1 to 12 (n=0 to 11)

11) Table of the synchronous encoder axis control devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name
D14820 to D14829	Axis 1 control device
D14830 to D14839	Axis 2 control device
D14840 to D14849	Axis 3 control device
D14850 to D14859	Axis 4 control device
D14860 to D14869	Axis 5 control device
D14870 to D14879	Axis 6 control device
D14880 to D14889	Axis 7 control device
D14890 to D14899	Axis 8 control device
D14900 to D14909	Axis 9 control device
D14910 to D14919	Axis 10 control device
D14920 to D14929	Axis 11 control device
D14930 to D14939	Axis 12 control device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D14820+10n D14821+10n	Pr.326	Synchronous encoder axis phase compensation advance time
D14822+10n	Cd.320	Synchronous encoder axis control start condition
D14823+10n	Cd.321	Synchronous encoder axis control method
D14824+10n D14825+10n	Cd.322	Synchronous encoder axis current value setting address
D14826+10n D14827+10n	Cd.325	Input value for synchronous encoder via device
D14828+10n D14829+10n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
- Q173DSCPU/Q172DSCPU: Axis No.1 to 12 (n=0 to 11)

12) Table of the output axis monitor devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
D13600 to D13629	Axis 1 monitor device	D14080 to D14109	Axis 17 monitor device
D13630 to D13659	Axis 2 monitor device	D14110 to D14139	Axis 18 monitor device
D13660 to D13689	Axis 3 monitor device	D14140 to D14169	Axis 19 monitor device
D13690 to D13719	Axis 4 monitor device	D14170 to D14199	Axis 20 monitor device
D13720 to D13749	Axis 5 monitor device	D14200 to D14229	Axis 21 monitor device
D13750 to D13779	Axis 6 monitor device	D14230 to D14259	Axis 22 monitor device
D13780 to D13809	Axis 7 monitor device	D14260 to D14289	Axis 23 monitor device
D13810 to D13839	Axis 8 monitor device	D14290 to D14319	Axis 24 monitor device
D13840 to D13869	Axis 9 monitor device	D14320 to D14349	Axis 25 monitor device
D13870 to D13899	Axis 10 monitor device	D14350 to D14379	Axis 26 monitor device
D13900 to D13929	Axis 11 monitor device	D14380 to D14409	Axis 27 monitor device
D13930 to D13959	Axis 12 monitor device	D14410 to D14439	Axis 28 monitor device
D13960 to D13989	Axis 13 monitor device	D14440 to D14469	Axis 29 monitor device
D13990 to D14019	Axis 14 monitor device	D14470 to D14499	Axis 30 monitor device
D14020 to D14049	Axis 15 monitor device	D14500 to D14529	Axis 31 monitor device
D14050 to D14079	Axis 16 monitor device	D14530 to D14559	Axis 32 monitor device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D13600+30n D13601+30n	Md.400	Current value after composite main shaft gear
D13602+30n D13603+30n	Md.401	Current value per cycle after main shaft gear
D13604+30n D13605+30n	Md.402	Current value per cycle after auxiliary shaft gear
D13606+30n D13607+30n	Md.422	Main shaft clutch slippage (accumulative)
D13608+30n D13609+30n	Md.425	Auxiliary shaft clutch slippage (accumulative)
D13610+30n D13611+30n	Md.406	Cam axis phase compensation amount
D13612+30n D13613+30n	Md.407	Cam axis current value per cycle
D13614+30n D13615+30n	Md.408	Cam reference position
D13616+30n D13617+30n	Md.409	Cam axis current feed value
D13618+30n	Md.410	Execute cam No.
D13619+30n	—	Unusable
D13620+30n D13621+30n	Md.411	Execute cam stroke amount
D13622+30n D13623+30n	Md.412	Execute cam axis length per cycle
D13624+30n D13625+30n D13626+30n D13627+30n D13628+30n D13629+30n	—	Unusable

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
 • Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
 • Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area can be used more than 17 axes in the Q172DSCPU can be used as a user device.
 However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

13) Table of the command generation axis control devices (SV22 advanced synchronous control only) QDS Ver.!

Device No.	Signal name	Device No.	Signal name
D15000 to D15149	Axis 1 control device	D17400 to D17549	Axis 17 control device
D15150 to D15299	Axis 2 control device	D17550 to D17699	Axis 18 control device
D15300 to D15449	Axis 3 control device	D17700 to D17849	Axis 19 control device
D15450 to D15599	Axis 4 control device	D17850 to D17999	Axis 20 control device
D15600 to D15749	Axis 5 control device	D18000 to D18149	Axis 21 control device
D15750 to D15899	Axis 6 control device	D18150 to D18299	Axis 22 control device
D15900 to D16049	Axis 7 control device	D18300 to D18449	Axis 23 control device
D16050 to D16199	Axis 8 control device	D18450 to D18599	Axis 24 control device
D16200 to D16349	Axis 9 control device	D18600 to D18749	Axis 25 control device
D16350 to D16499	Axis 10 control device	D18750 to D18899	Axis 26 control device
D16500 to D16649	Axis 11 control device	D18900 to D19049	Axis 27 control device
D16650 to D16799	Axis 12 control device	D19050 to D19199	Axis 28 control device
D16800 to D16949	Axis 13 control device	D19200 to D19349	Axis 29 control device
D16950 to D17099	Axis 14 control device	D19350 to D19499	Axis 30 control device
D17100 to D17249	Axis 15 control device	D19500 to D19649	Axis 31 control device
D17250 to D17399	Axis 16 control device	D19650 to D19799	Axis 32 control device

Ver.! : Refer to Section 1.3 for the software version that supports this function.

• Details of each axis

Device No.	Symbol	Signal name
D15000+150n	Pr.400	Main input axis No.
D15001+150n	Pr.401	Sub input axis No.
D15002+150n	Pr.402	Composite main shaft gear
D15003+150n	—	Unusable
D15004+150n D15005+150n	Pr.403	Main shaft gear: Numerator
D15006+150n D15007+150n	Pr.404	Main shaft gear: Denominator
D15008+150n	Pr.405	Main shaft clutch control setting
D15009+150n	Pr.406	Main shaft clutch reference address setting
D15010+150n D15011+150n	Pr.407	Main shaft clutch ON address
D15012+150n D15013+150n	Pr.408	Travel value before main shaft clutch ON
D15014+150n D15015+150n	Pr.409	Main shaft clutch OFF address
D15016+150n D15017+150n	Pr.410	Travel value before main shaft clutch OFF
D15018+150n	Pr.411	Main shaft clutch smoothing method
D15019+150n	Pr.412	Main shaft clutch smoothing time constant
D15020+150n D15021+150n	Pr.413	Slippage amount at main shaft clutch ON
D15022+150n D15023+150n	Pr.414	Slippage amount at main shaft clutch OFF
D15024+150n	Pr.418	Auxiliary shaft axis No.
D15025+150n	Pr.419	Composite auxiliary shaft gear
D15026+150n D15027+150n	Pr.420	Auxiliary shaft gear: Numerator
D15028+150n D15029+150n	Pr.421	Auxiliary shaft gear: Denominator
D15030+150n	Pr.422	Auxiliary shaft clutch control setting
D15031+150n	Pr.423	Auxiliary shaft clutch reference address setting
D15032+150n D15033+150n	Pr.424	Auxiliary shaft clutch ON address
D15034+150n D15035+150n	Pr.425	Travel value before auxiliary shaft clutch ON
D15036+150n D15037+150n	Pr.426	Auxiliary shaft clutch OFF address
D15038+150n D15039+150n	Pr.427	Travel value before auxiliary shaft clutch OFF
D15040+150n	Pr.428	Auxiliary shaft clutch smoothing method
D15041+150n	Pr.429	Auxiliary shaft clutch smoothing time constant
D15042+150n D15043+150n	Pr.430	Slippage amount at auxiliary shaft clutch ON
D15044+150n D15045+150n	Pr.431	Slippage amount at auxiliary shaft clutch OFF
D15046+150n	Pr.434	Speed change gear 1
D15047+150n	Pr.435	Speed change gear 1 smoothing time constant
D15048+150n D15049+150n	Pr.436	Speed change ratio 1: Numerator

• Details of each axis (Continued)

Device No.	Symbol	Signal name
D15050+150n D15051+150n	Pr.437	Speed change ratio 1: Denominator
D15052+150n	Pr.490	Speed change gear 2
D15053+150n	Pr.491	Speed change gear 2 smoothing time constant
D15054+150n D15055+150n	Pr.492	Speed change ratio 2: Numerator
D15056+150n D15057+150n	Pr.493	Speed change ratio 2: Denominator
D15058+150n	Pr.438	Cam axis cycle unit setting
D15059+150n	Pr.442	Cam axis length per cycle change setting
D15060+150n D15061+150n	Pr.439	Cam axis length per cycle
D15062+150n	Pr.440	Cam No.
D15063+150n	—	Unusable
D15064+150n D15065+150n	Pr.441	Cam stroke amount
D15066+150n D15067+150n	Pr.444	Cam axis phase compensation advance time
D15068+150n	Pr.445	Cam axis phase compensation time constant
D15069+150n	Pr.448	Synchronous control parameter block No.
D15070+150n	Pr.447	Output axis smoothing time constant
D15071+150n D15072+150n D15073+150n D15074+150n D15075+150n D15076+150n D15077+150n D15078+150n D15079+150n D15080+150n D15081+150n D15082+150n D15083+150n D15084+150n D15085+150n D15086+150n D15087+150n D15088+150n D15089+150n D15090+150n D15091+150n D15092+150n D15093+150n D15094+150n D15095+150n D15096+150n D15097+150n D15098+150n D15099+150n	—	Unusable

• Details of each axis (Continued)

Device No.	Symbol	Signal name
D15100+150n	Pr.460	Setting method of current value per cycle after main shaft gear
D15101+150n	Pr.461	Setting method of current value per cycle after auxiliary shaft gear
D15102+150n	Pr.462	Cam axis position restoration object
D15103+150n	Pr.463	Setting method of cam reference position
D15104+150n	Pr.464	Setting method of cam axis current value per cycle
D15105+150n	—	Unusable
D15106+150n D15107+150n	Pr.465	Current value per cycle after main shaft gear (Initial setting)
D15108+150n D15109+150n	Pr.466	Current value per cycle after auxiliary shaft gear (Initial setting)
D15110+150n D15111+150n	Pr.467	Cam reference position (Initial setting)
D15112+150n D15113+150n	Pr.468	Cam axis current value per cycle (Initial setting)
D15114+150n D15115+150n D15116+150n D15117+150n D15118+150n D15119+150n D15120+150n D15121+150n D15122+150n D15123+150n D15124+150n D15125+150n D15126+150n D15127+150n D15128+150n D15129+150n	—	Unusable
D15130+150n	Cd.407	Synchronous control change command
D15131+150n	Cd.409	Synchronous control reflection time
D15132+150n D15133+150n	Cd.408	Synchronous control change value
D15134+150n D15135+150n D15136+150n D15137+150n D15138+150n D15139+150n D15140+150n D15141+150n D15142+150n D15143+150n D15144+150n D15145+150n D15146+150n	—	Unusable

• Details of each axis (Continued)

Device No.	Symbol	Signal name
D15147+150n	—	Unusable
D15148+150n		
D15149+150n		

POINT

(1) "n" in the above device No. shows the numerical value which correspond to axis No.
 • Q173DSCPU: Axis No.1 to 32 (n=0 to 31)
 • Q172DSCPU: Axis No.1 to 16 (n=0 to 15)

(2) The device area can be used more than 17 axes in the Q172DSCPU can be used as a user device.
 However, when the project of Q172DSCPU is replaced with Q173DSCPU, this area cannot be used as a user device.

14) Table of the common devices (SV13/SV22)

Device No.	Signal name		Signal direction
D704	PLC ready flag request		Command device
D705	Speed switching point specified flag request		
D706	All axes servo ON command request		
D707	Real mode/virtual mode switching request (SV22) (Note-1)		
D708	JOG operation simultaneous start command request		
D709	Unusable		—
D710 to D713	JOG operation simultaneous start axis setting register		Command device
D714 D715	Manual pulse generator axis 1 No. setting register		
D716 D717	Manual pulse generator axis 2 No. setting register		
D718 D719	Manual pulse generator axis 3 No. setting register		
D720	Axis 1	Manual pulse generators 1 pulse input magnification setting register (Note-2, 3)	
D721	Axis 2		
D722	Axis 3		
D723	Axis 4		
D724	Axis 5		
D725	Axis 6		
D726	Axis 7		
D727	Axis 8		
D728	Axis 9		
D729	Axis 10		
D730	Axis 11		
D731	Axis 12		
D732	Axis 13		
D733	Axis 14		
D734	Axis 15		
D735	Axis 16		
D736	Axis 17		
D737	Axis 18		
D738	Axis 19		
D739	Axis 20		

Device No.	Signal name		Signal direction
D740	Axis 21	Manual pulse generators 1 pulse input magnification setting register (Note-2, 3)	Command device
D741	Axis 22		
D742	Axis 23		
D743	Axis 24		
D744	Axis 25		
D745	Axis 26		
D746	Axis 27		
D747	Axis 28		
D748	Axis 29		
D749	Axis 30		
D750	Axis 31		
D751	Axis 32		
D752	Manual pulse generator 1 smoothing magnification setting register		
D753	Manual pulse generator 2 smoothing magnification setting register		
D754	Manual pulse generator 3 smoothing magnification setting register		
D755	Manual pulse generator 1 enable flag request		
D756	Manual pulse generator 2 enable flag request		
D757	Manual pulse generator 3 enable flag request		
D758 to D799	Unusable (42 points)		—

(Note-1): It is unusable in the SV22 advanced synchoronous control.
(Note-2): The following range is valid.
Q172DSCPU: Axis No.1 to 16, Q172DCPU(-S1): Axis No.1 to 8
(Note-3): The following device area is unusable.
Q172DSCPU: 17 axes or more, Q172DCPU(-S1): 9 axes or more

1.3 Restrictions by the Software's Version

There are restrictions in the function that can be used by the version of the operating system software and programming software.
The combination of each version and a function is shown in Table1.1.

Table 1.1 Restrictions by the Software's Version

Function	Operating system software version (Note-1), (Note-2)		
	Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)	
Checking Motion controller's serial number and operating system software version in GX Developer	—	00D	
Advanced S-curve acceleration/deceleration (Except constant-speed control (CPSTART) of servo program.)	—	00H	
Direct drive servo MR-J3-□B-RJ080W	—	00H	
Servo amplifier display servo error code (#8008+20n)	—	00H	
0.44ms fixed-cycle event task	—	00H	
444μs coasting timer (SD720, SD721)	—	00H	
Synchronous encoder current value monitor in real mode	—	00H	
Display of the past ten times history in current value history monitor	—	00H	
Amplifier-less operation	—	00H	
Servo instruction (Home position return (ZERO), high speed oscillation (OSC)) and manual pulse generator operation in mixed function of virtual mode/real mode	—	00H	
Advanced S-curve acceleration/deceleration in constant-speed control (CPSTART) of servo program.	—	00K	
External input signal (DOG) of servo amplifier in home position return of count method and speed-position switching control	—	00G	
Communication via PERIPHERAL I/F	—	00H	
Motion SFC operation control instruction Type conversion (DFLT, SFLT)	—	00L	
Vision system dedicated function (MVOPEN, MVLOAD, MVTRG, MVPST, MVIN, MVFIN, MVCLOSE, MVCOM)	—	00L	
Home position return of scale home position signal detection method	—	00L	
Real time display function in digital oscilloscope function	—	00N	
Rapid stop deceleration time setting error invalid function	—	00S	

Programming software version				Section of reference
MELSOFT MT Works2 (MT Developer2)		MR Configurator2	MR Configurator	
Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)			
—	—	—	—	(Note-2)
1.39R	1.06G	—	—	(Note-3)
1.39R	1.06G	1.01B	C2	
—	—	—	—	(Note-3), (Note-4)
1.39R	1.06G	—	—	Chapter 9
—	—	—	—	(Note-5)
—	—	—	—	(Note-4)
1.39R	1.06G	—	—	(Note-5)
—	—	—	—	(Note-5)
1.39R	1.09K	—	—	(Note-4)
1.39R	1.09K	—	—	(Note-3)
1.39R	1.15R	—	—	
1.39R	1.15R	—	—	(Note-5)
1.39R	1.15R	—	—	Section 5.7.7 Section 5.7.8
1.39R	1.15R	—	—	Section 5.15 APPENDIX 3
1.39R	1.15R	—	—	(Note-3)
1.39R	1.17T	—	—	
—	—	—	—	(Note-3)

—: There is no restriction by the version.

(Note-1): SV13/SV22 is the completely same version.

(Note-2): The operating system software version can be confirmed in the operating system software (CD-ROM), MT Developer2 or GX Works2/GX Developer. (Refer to "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON) Section 1.3, 1.4".)

(Note-3): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)

(Note-4): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)

(Note-5): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)

(Note-6): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (Safety Observation)

(Note-7): Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)

Table 1.1 Restrictions by the Software's Version (continued)

Function		Operating system software version (Note-1), (Note-2)		
		Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)	
Vision system dedicated function (MVOUT)		—	00S	
Motion SFC operation control instruction Program control (IF - ELSE - IEND, SELECT -CASE - SEND, FOR -NEXT, BREAK)		—	00R	
Display format depending on the error setting data information of motion error history device (#8640 to #8735)		—	00S	
Product information list device (#8736 to #8751)		—	00S	
Safety observation function		—	00S	
Feed current value update command (M3212+20n) valid in speed control (Ⅰ)		00B	Not support	
External forced stop input ON latch (SM506)		00B	00S	
Operation method (SD560)		00B	Not support	
Advanced synchronous control		00B	Not support	
Limit switch output function expansion		00B	Not support	
Driver communication function (SSCNETⅢ)		00C	Not support	
Intelligent function module support		00C	Not support	
SSCNETⅢ/H head module connection		00C	Not support	
Cam auto-generation (CAMMK) easy stroke ratio cam		00C	Not support	
Acceleration/deceleration time change function		00C	Not support	
Home position return of dogless home position signal reference method		00C	Not support	
Setting range expansion of backlash compensation amount		00C	Not support	
Multiple CPU synchronous control		00C	Not support	
Cam axis length per cycle change during synchronous control		00C	Not support	
Servo driver VCⅡ series manufactured by Nikki Denso Co., Ltd.	SSCNETⅢ	—	00L	
	SSCNETⅢ/H	00D	Not support	
Inverter FR-A700 series		—	—	
Synchronous encoder via servo amplifier		00D	Not support	
Driver communication function (SSCNETⅢ/H)		00D	Not support	
Optical hub unit connection		00F	Not support	
Home position return of driver home position return method		00H	Not support	
Stepping motor module AlphaStep/5-phase manufactured by ORIENTAL MOTOR Co., Ltd.		00H	Not support	
Servo driver VPH series manufactured by Nikki Denso Co., Ltd.		00H	Not support	
IAI electric actuator controller manufactured by IAI Corporation		00H	Not support	
Inverter FR-A800 series		00J	Not support	

Programming software version				Section of reference
MELSOFT MT Works2 (MT Developer2)		MR Configurator2	MR Configurator	
Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)			
1.39R	1.39R	—	—	Section 5.15.6 APPENDIX 3
1.39R	1.39R	—	—	Section 5.17
—	—	—	—	Section 12.2
—	—	—	—	(Note-3), (Note-4)
1.39R	1.39R	—	—	(Note-6)
—	Not support	—	—	(Note-3)
—	—	—	—	(Note-5)
—	Not support	—	—	(Note-5)
1.47Z	Not support	—	—	(Note-7)
1.47Z	Not support	—	—	(Note-5)
—	Not support	—	—	(Note-5)
1.56J	Not support	—	—	(Note-5)
1.56J	Not support	—	—	(Note-5)
1.56J	Not support	—	—	Section 5.18.4
1.56J	Not support	—	—	(Note-3)
1.56J	Not support	—	—	(Note-3)
1.56J	Not support	—	—	(Note-3)
1.56J	Not support	—	—	(Note-7)
1.56J	Not support	—	—	(Note-7)
1.34L	1.15R	—	—	(Note-3)
1.56J	Not support	—	—	(Note-3)
1.34L	1.15R	—	—	(Note-3)
1.68W	Not support	1.23Z	Not support	(Note-7)
1.68W	Not support	1.23Z	Not support	(Note-5)
—	Not support	—	—	(Note-3)
1.118Y	Not support	—	—	(Note-3)
1.118Y	Not support	—	—	(Note-3)
1.118Y	Not support	—	—	(Note-3)
1.118Y	Not support	—	—	(Note-3)
1.120A	Not support	—	—	(Note-3)

—: There is no restriction by the version.

(Note-1): SV13/SV22 is the completely same version.

(Note-2): The operating system software version can be confirmed in the operating system software (CD-ROM), MT Developer2 or GX Works2/GX Developer. (Refer to "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON) Section 1.3, 1.4".)

(Note-3): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)

(Note-4): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)

(Note-5): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)

(Note-6): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (Safety Observation)

(Note-7): Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)

Table 1.1 Restrictions by the Software's Version (continued)

Function	Operating system software version [Note-1, Note-2]		
	Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)	
Improvement of absolute positioning operation for servo driver VCⅡ/VPH series manufactured by Nikki Denso Co., Ltd., and stepping motor module AlphaStep/5-phase manufactured by ORIENTAL MOTOR Co., Ltd.	00L	Not support	

Programming software version					Section of reference
MELSOFT MT Works2 (MT Developer2)		MR Configurator2	MR Configurator		
Q173DSCPU/Q172DSCPU	Q173DCPU(-S1)/Q172DCPU(-S1)				
—	Not support	—	—		(Note-3)

—: There is no restriction by the version.

(Note-1): SV13/SV22 is the completely same version.

(Note-2): The operating system software version can be confirmed in the operating system software (CD-ROM), MT Developer2 or GX Works2/GX Developer. (Refer to "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON) Section 1.3, 1.4".)

(Note-3): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)

(Note-4): Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)

(Note-5): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)

(Note-6): Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (Safety Observation)

(Note-7): Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)

1.4 Programming Software Version

The programming software versions that support Motion CPU are shown below.

Motion CPU	MELSOFT MT Works2 (MT Developer2)		MR Configurator2	MR Configurator
	SV13/SV22	SV43		
Q173DSCPU	1.39R (Note-1)		1.10L	Not support
Q172DSCPU	1.39R (Note-1)		1.10L	Not support
Q173DCPU-S1	1.00A (Note-2)	1.03D (Note-3)	1.00A	C0 (Note-4)
Q172DCPU-S1	1.00A (Note-2)	1.03D (Note-3)	1.00A	C0 (Note-4)
Q173DCPU	1.00A	1.03D	1.00A	C0 (Note-4)
Q172DCPU	1.00A	1.03D	1.00A	C0 (Note-4)

(Note-1): Use version 1.47Z or later to use advanced synchronous control method.
(Note-2): Use version 1.12N or later to communicate via PERIPHERAL I/F.
(Note-3): Use version 1.23Z or later to communicate via PERIPHERAL I/F.
(Note-4): Use version C1 or later to use MR Configurator combination with MT Developer2.

2. STRUCTURE OF THE MOTION CPU PROGRAM

(1) Motion CPU programs are created in the Motion SFC of flowchart format. The motion control of servomotors is performed using the real-mode servo programs specified by motion-control steps in a Motion SFC program in SV13/SV22 real mode.

(2) Virtual servomotors in a mechanical system program are controlled using the virtual mode servo programs specified by motion-control steps so as to enable synchronous control in SV22 virtual mode.

(3) By setting the synchronous control parameter and starting the synchronous control for each output axis, the SV22 advanced synchronous control is performed in synchronization with the input axis (servo input axis, command generation axis, synchronous encoder axis).

(4) By using the sequence program in the PLC CPU, Motion dedicated PLC instructions in the Motion CPU perform the following controls.
• Start of Motion SFC program
• Start of servo program
• Change of current value/speed/torque limit value
• Start of event task

(5) Refer to the following for the details of Motion SFC programs, motion control in real mode, motion control in virtual mode, and motion control in advanced synchronous control, and Motion dedicated PLC instructions in the PLC CPU.

Item	Reference
Motion dedicated PLC instructions in the PLC CPU	Chapter 3
Motion SFC program	Chapter 4
Motion control in SV13/SV22 real mode (Servo program)	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)
Motion control in SV22 virtual mode (Mechanical system program)	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)
Motion control in SV22 advanced synchronous control (Synchronous control parameter)	Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)

2.1 Motion Control in SV13/SV22 Real Mode

(1) System with servomotor is controlled directly using the servo program in (SV13/SV22) real mode.

(2) Setting of the positioning parameter and creation of the servo program/ Motion SFC program are required.

(3) The procedure of positioning control is shown below:

1) Motion SFC program is requested to start using the D(P).SFCS instruction of the sequence program.
(Motion SFC program can also be started automatically by parameter setting.)

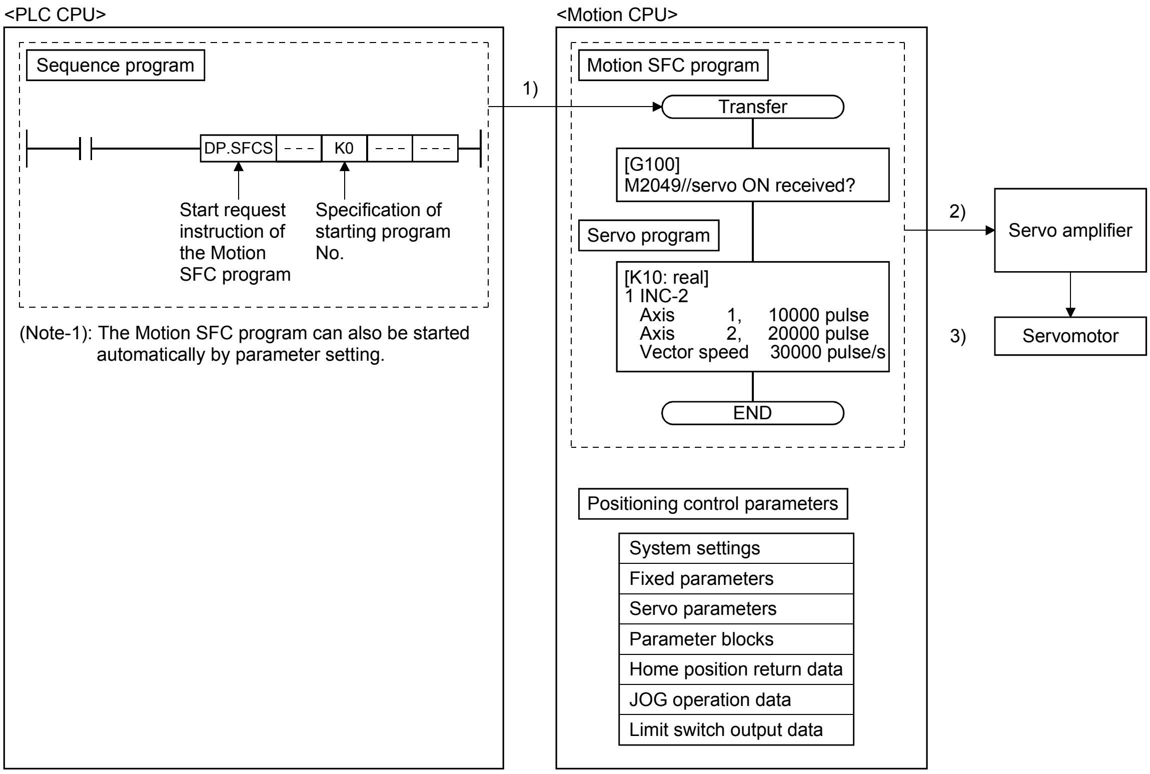
↓

2) Execute the positioning control using the specified the Motion SFC program.
(Output to the servo amplifier)

↓

3) The servo motor is controlled.

Program structure in SV13/SV22 real mode



2.2 Motion Control in SV22 Virtual Mode

(1) Software-based synchronous control is performed using the mechanical system program constructed by virtual main shaft and mechanical module in (SV22) virtual mode.

(2) Mechanical system programs is required in addition to the positioning parameter, servo program/Motion SFC program used in real mode.

(3) The procedure of positioning control in virtual mode is shown below:

1) Motion SFC program for virtual mode is requested to start using the D(P).SFCS instruction of the sequence program.
(Motion SFC program can also be started automatically by parameter setting.)

↓

2) The virtual servomotor in the mechanical system program is started.

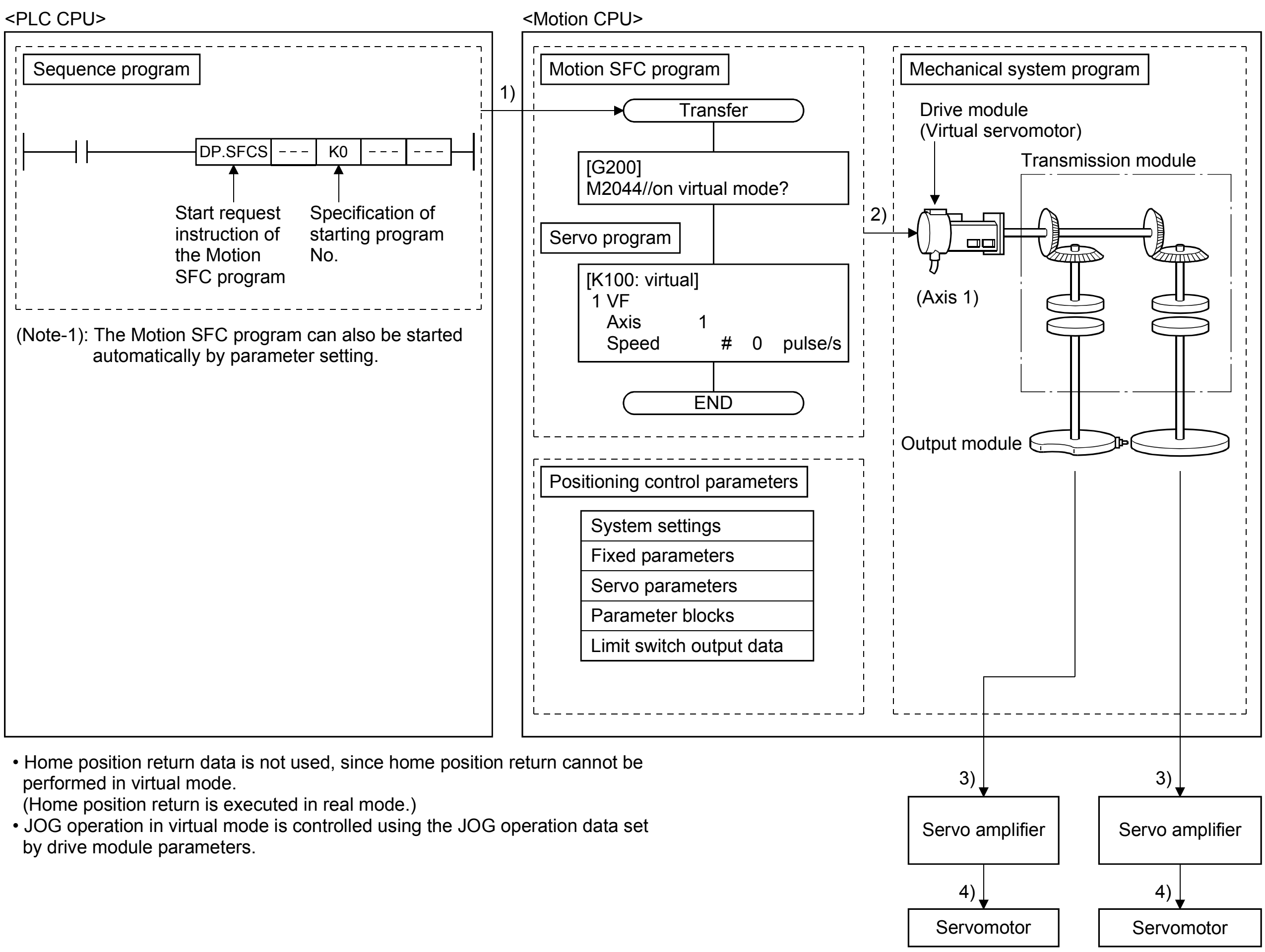
↓

3) Output the operation result obtained through the transmission module to the servo amplifier set as the output module.

↓

4) The servo motor is controlled.

Program structure in SV22 virtual mode



• Home position return data is not used, since home position return cannot be performed in virtual mode.
(Home position return is executed in real mode.)
• JOG operation in virtual mode is controlled using the JOG operation data set by drive module parameters.

2.3 Motion Control in SV22 Advanced Synchronous Control

(1) Advanced synchronous control can be achieved using software instead of controlling mechanically with gear, shaft, speed change gear or cam etc.

(2) The synchronous control parameter is required in addition to the positioning parameter, servo program/Motion SFC program used in real mode.

(3) The procedure of positioning control in advanced synchronous control is shown below:

1) Motion SFC program for advanced synchronous control is requested to start using the D(P).SFCS instruction of the sequence program.
(Motion SFC program can also be started automatically by parameter setting.)

↓

2) The command generation axis in the advanced synchronous control is started.

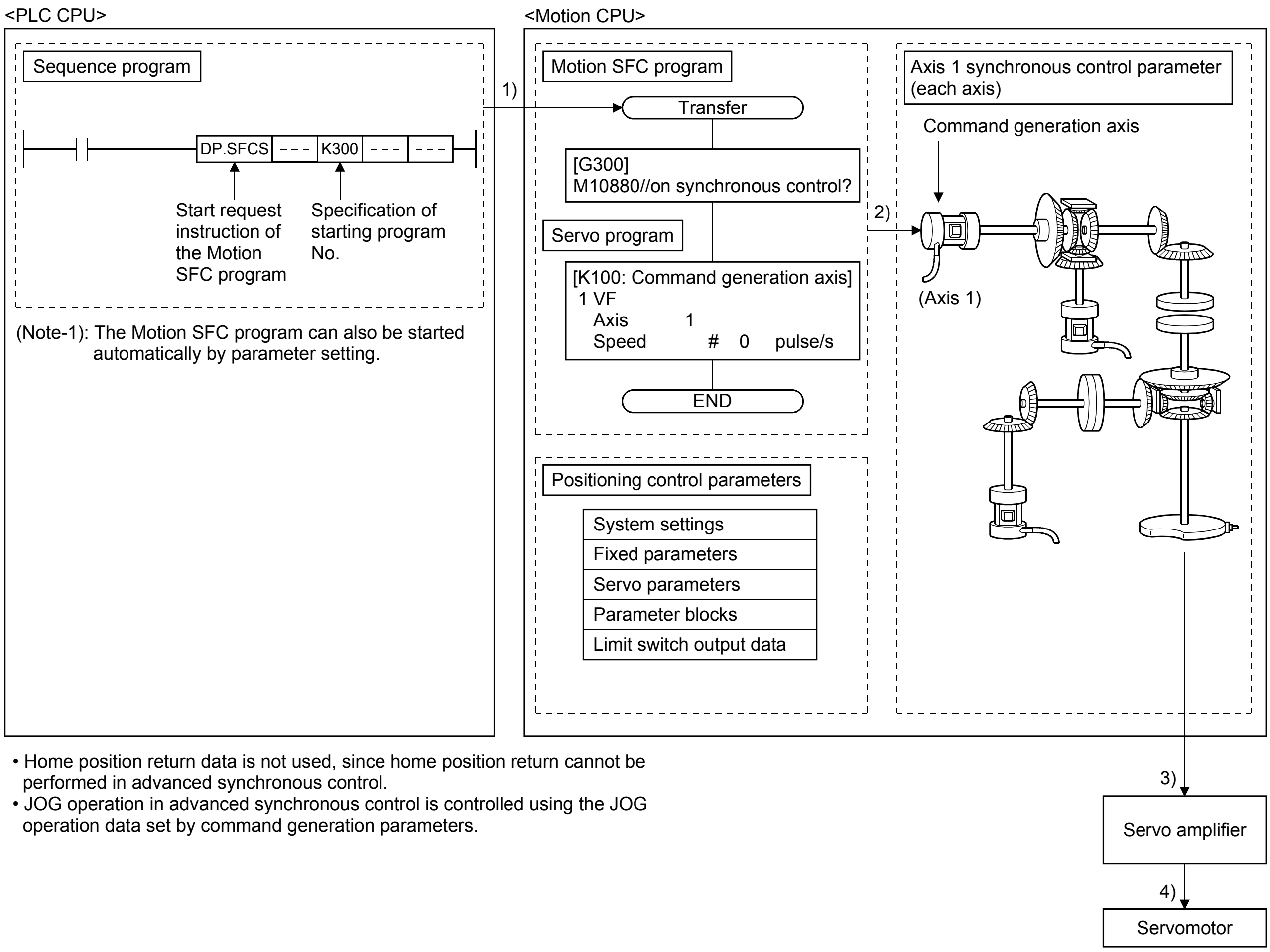
↓

3) Output the synchronous control parameter to the servo amplifier of each axis.

↓

4) The servo motor is controlled.

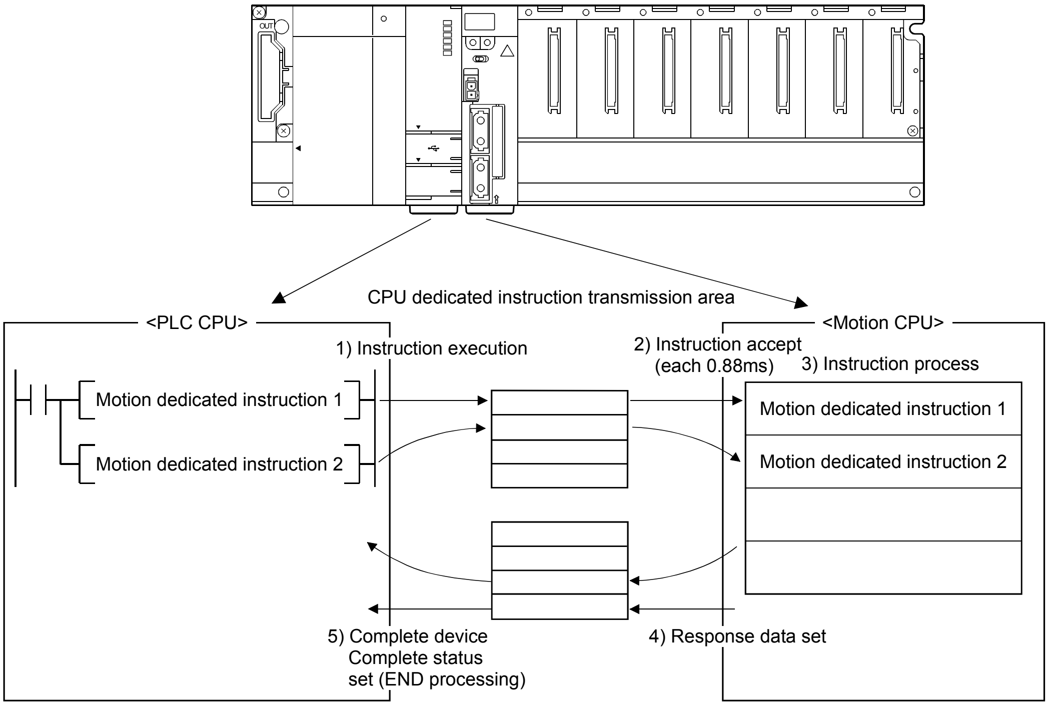
Program structure in SV22 advanced synchronous control



3. MOTION DEDICATED PLC INSTRUCTION

3.1 Outline of Motion Dedicated PLC Instruction

Motion dedicated PLC instruction is used to access the device data and start-up program of Motion CPU from PLC CPU.
Motion dedicated PLC instruction is transmitted through the CPU dedicated instruction transmission area set up in system area on the shared memory at the Multiple CPU high speed transmission.
Outline operation for Motion dedicated PLC instruction is shown below.



3.2 Motion Dedicated PLC Instruction

The Motion dedicated PLC instruction that can be executed toward the Motion CPU which installed the operating system software (SV13/SV22) for Q173D(S)CPU/ Q172D(S)CPU is shown below.

Instruction		Description	SV13	SV22		
				Real mode		Virtual mode
					Advanced synchronous control QDS Ver.!	
D(P).SFCS		Start request of the specified Motion SFC program	○	○	○	○
D(P).SVST		Start request of the specified servo program	○	○	○	○
D(P).CHGA	J	Current value change request of the specified axis	○	○	○ (Note-1)	○
	E		×	×	× (Note-2)	○
	C		×	×	× (Note-2)	○
D(P).CHGAS		Current value change request of the specified command generation axis QDS Ver.!	×	×	○	× (Note-4)
D(P).CHGV		Speed change request of the specified axis	○	○	○ (Note-3)	○
D(P).CHGVS		Speed change request of the specified command generation axis QDS Ver.!	×	×	○	× (Note-4)
D(P).CHGT		Torque control value change request of the specified axis	○	○	○	○
D(P).CHGT2		Torque control value individual change request of the specified axis QDS	○	○	○	○
D(P).DDWR		Write device data of the self CPU to the device of another Motion CPU	○	○	○	○
D(P).DDRD		Read device data of another Motion CPU to the device of self CPU	○	○	○	○
D(P).GINT		Execute request of an event task of Motion SFC program	○	○	○	○

○ : Possible, × : Not possible

(Note-1): If the instruction is executed for the axis during the synchronous control, a minor error (error code: 300) occurs and the instruction is not executed.

(Note-2): The error code (2203(H)) is stored in the complete status storage device specified in the D(P).CHGA instruction, and the instruction is not executed.
Execute the current value change of the synchronous encoder by using [Rq.320] Synchronous encoder axis control request (M11601+4n). Execute the current value change of the cam axis by using the synchronous control change function.
(Refer to the "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for details.)

(Note-3): If the instruction is executed toward the axis during the synchronous control, the instruction is ignored.

(Note-4): The error code (2002(H)) is stored in the complete status storage device specified in the D(P).CHGAS instruction or the D(P).CHGVS instruction, and the instruction is not executed.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

3.2.1 Motion SFC start request from the PLC CPU to the Motion CPU: D(P).SFCS (PLC instruction: D(P).SFCS)

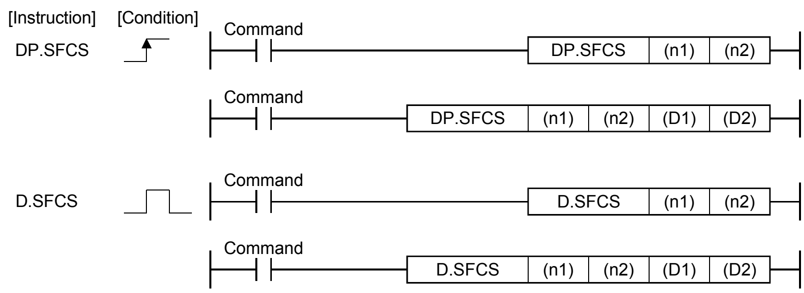
Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): Local devices cannot be used.

(Note-3): Setting data (n1) to (D2): Index qualification possible (except constant)



[Setting data]

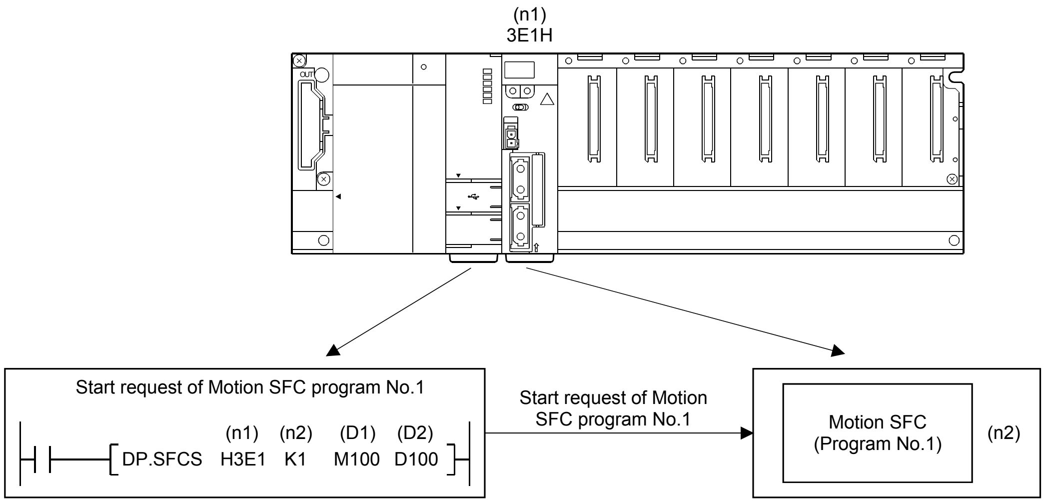
Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(n2)	Motion SFC program No. to start.	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

(Note-1): Omission possible with both of (D1) and (D2) omission.

[Controls]

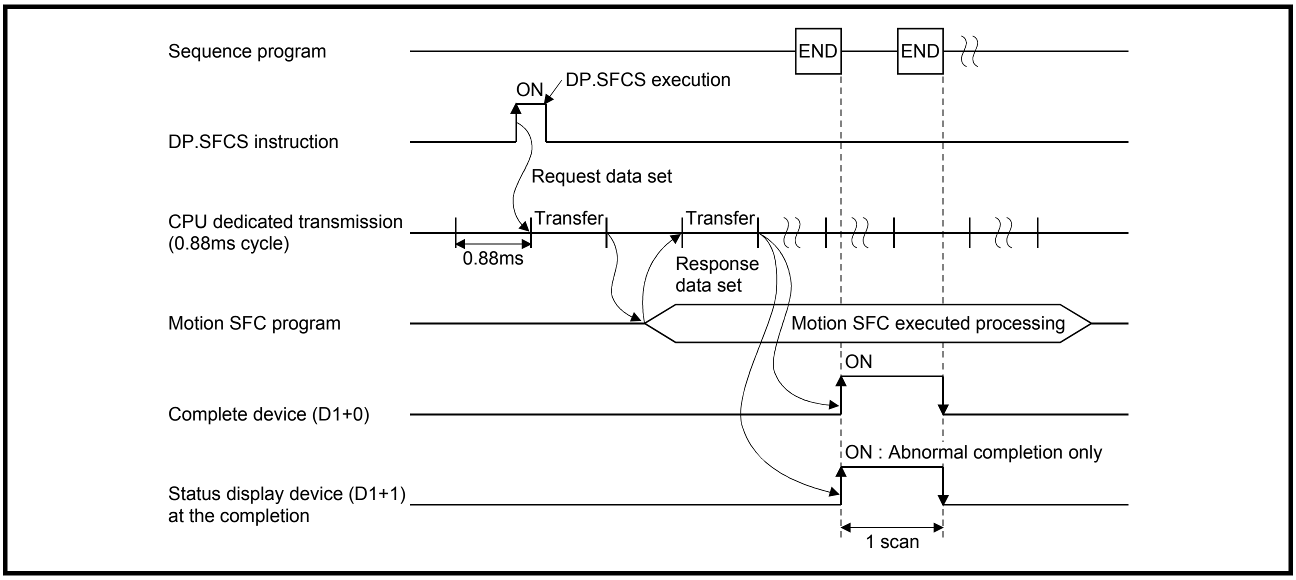
(1) Request to start the Motion SFC program of program No. specified with (n2). The Motion SFC program can start any task setting of the normal task, event task and NMI task.

(2) This instruction is always valid regardless of the state of real mode/virtual mode/ mode switching when the operating system software of Motion CPU is SV22.



[Operation]

Outline operation between CPUs at the DP.SFCS instruction execution is shown below.



[Setting range]

(1) Setting of Motion SFC program

(n2) usable range
0 to 255

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code) (H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct a sequence program.
2100	There are 65 or more simultaneous D(P).SFCS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	
2200	The Motion SFC program No. to start is outside the range of 0 to 255.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

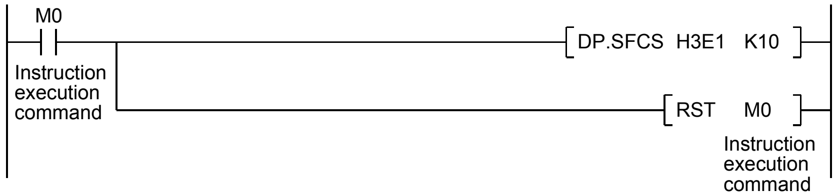
Error code(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct a sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device which cannot be used for the instruction specified is specified.	

(Note): 0 (Normal)

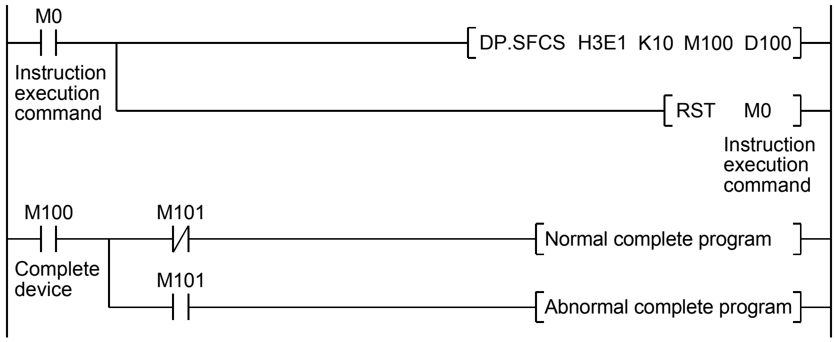
[Program example]

(1) Program which starts the Motion SFC program No.10 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

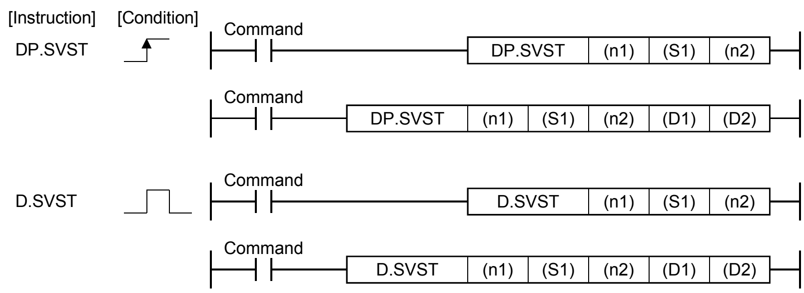


3.2.2 Servo program start request from the PLC CPU to the Motion CPU: D(P).SVST (PLC instruction: D(P).SVST)

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2): Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No.("Jn") (Note-2) to start. Q173DSCPU/Q173DCPU(-S1) : J1 to J32 Q172DSCPU : J1 to J16 Q172DCPU(-S1) : J1 to J8	User	Character sequence
(n2)	Servo program No. to execute	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

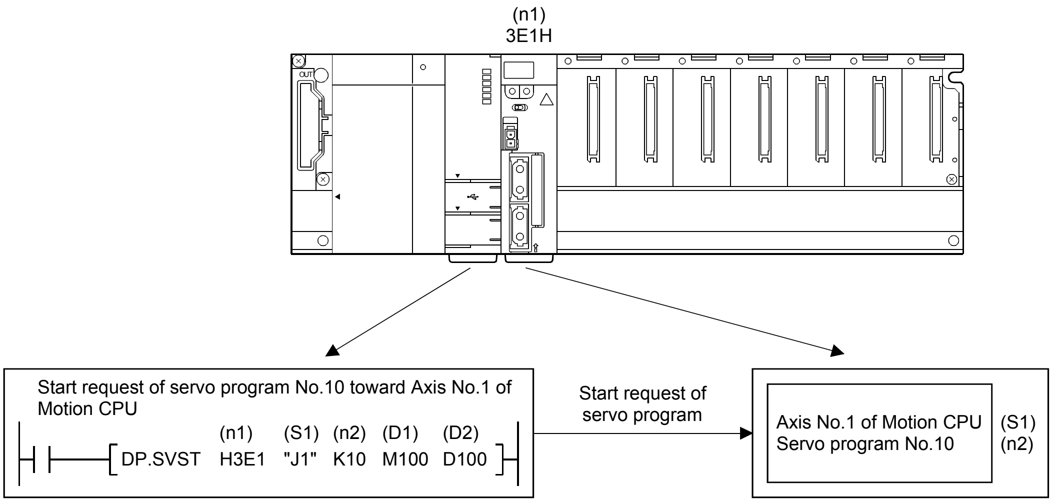
(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) Request to start the servo program specified with (n2).

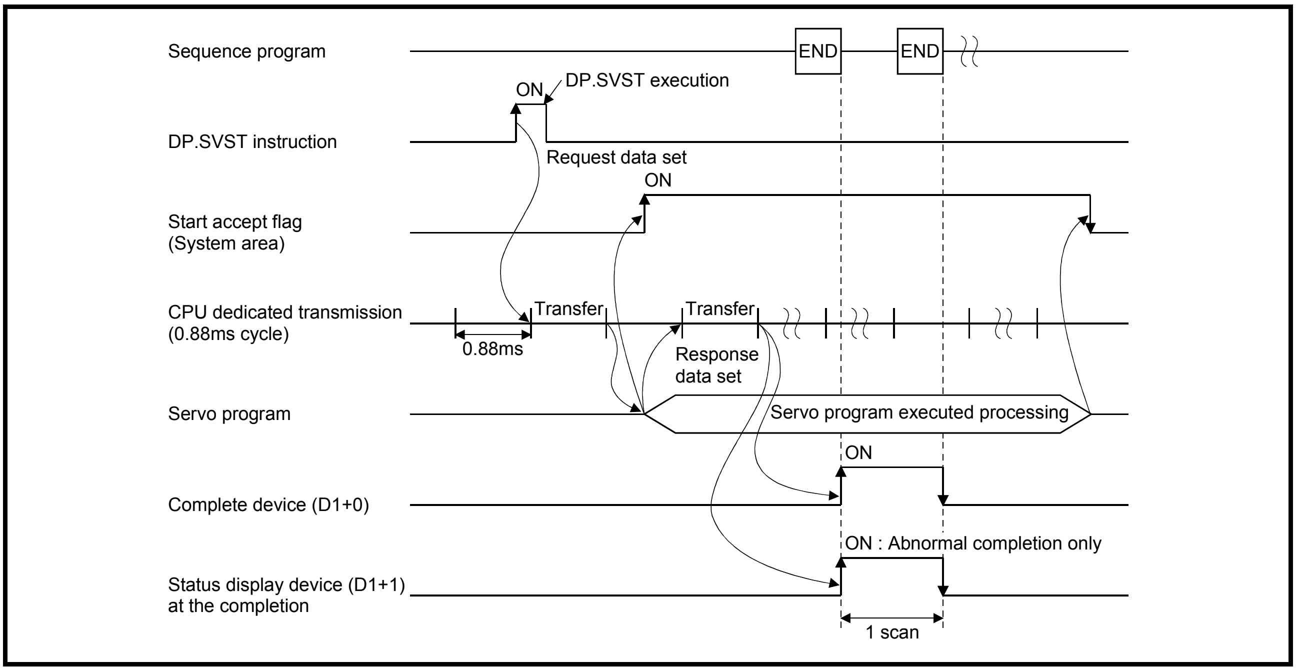
(2) It is necessary to take an inter-lock by the start accept flag of CPU shared memory and user device so that multiple instructions may not be executed toward the same axis of the same Motion CPU No.



POINT
Refer to Section "3.3 Precautions" for details of the start accept flag.

[Operation]

Outline operation between CPUs at the DP.SVST instruction execution is shown below.



[Setting range]

(1) Setting of the starting axis

The starting axis set as (S1) is set J + Axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q173DCPU(-S1)	
Q172DSCPU	1 to 16
Q172DCPU(-S1)	1 to 8

Up to 8 axes can be set. Set them without dividing in a space etc. for multiple axes setting.
Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.
And, the axis No. to start does not need to be a order.

Example) When multiple axes (Axis1, Axis2, Axis10, Axis11) are set.
"J1J2J10J11"

(2) Setting of the servo program No.

(n2) usable range
0 to 4095

[Start accept flag (System area)]

The complete status of start accept flag is stored in the address of start accept flag in the CPU shared memory for target CPU.

<table>
<tr><th>CPU shared memory address
( ) is decimal address</th><th>Description</th></tr>
<tr><td>204H(516)
205H(517)</td><td>The start accept flag for 32 axes are stored corresponding to each bit.
Bits are actually set as the following:
• Q173DSCPU/Q173DCPU(-S1) : J1 to J32
• Q172DSCPU : J1 to J16
• Q172DCPU(-S1) : J1 to J8
OFF : Start accept enable
ON : Start accept disable
<table>
<tr><th></th><th>b15</th><th>b14 … b2</th><th>b1</th><th>b0</th></tr>
<tr><td>204H(516) address</td><td>J16</td><td>• • • • • • • • •</td><td>J2</td><td>J1</td></tr>
<tr><td>205H(517) address</td><td>J32</td><td>• • • • • • • • •</td><td>J18</td><td>J17</td></tr>
</table></td></tr>
<tr><td>20EH(526)
20FH(527)
QDS</td><td>The command generation axis start accept flag for 32 axes are stored corresponding to each bit.
Bits are actually set as the following:
• Q173DSCPU: J1 to J32
• Q172DSCPU: J1 to J16
OFF : Start accept enable
ON : Start accept disable
<table>
<tr><th></th><th>b15</th><th>b14 … b2</th><th>b1</th><th>b0</th></tr>
<tr><td>20EH(526) address</td><td>J16</td><td>• • • • • • • • •</td><td>J2</td><td>J1</td></tr>
<tr><td>20FH(527) address</td><td>J32</td><td>• • • • • • • • •</td><td>J18</td><td>J17</td></tr>
</table></td></tr>
</table>

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note-1) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2100	There are the following number or more simultaneous D(P).SVST/ D(P).CHGA/D(P).CHGAS QDS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • Q173DSCPU/Q172DSCPU : 129 or more (Note-2) • Q173DCPU(-S1)/Q172DCPU(-S1) : 65 or more	
2201	The servo program No. to execute is outside the range of 0 to 4095.	
2202	Axis No. set by D(P).SVST instruction is wrong.	

(Note-1): 0000H (Normal)
(Note-2): 65 or more for operating system software version "00A".

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

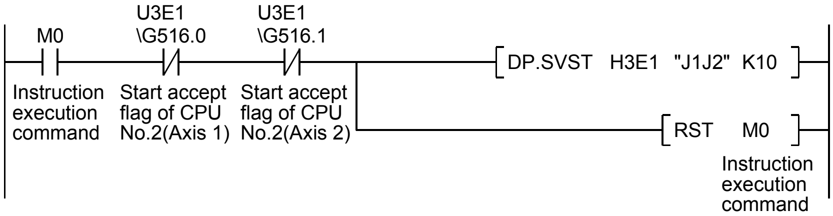
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

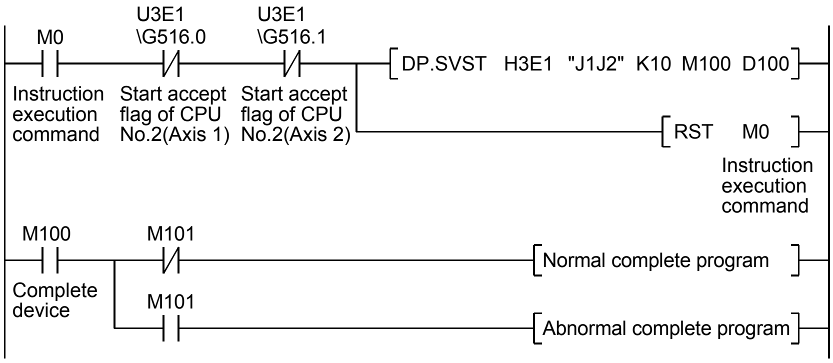
[Program example]

(1) Program which requests to start of the servo program No.10 toward Axis 1, Axis 2 of the Motion CPU (CPU No.2), when M0 turned ON.

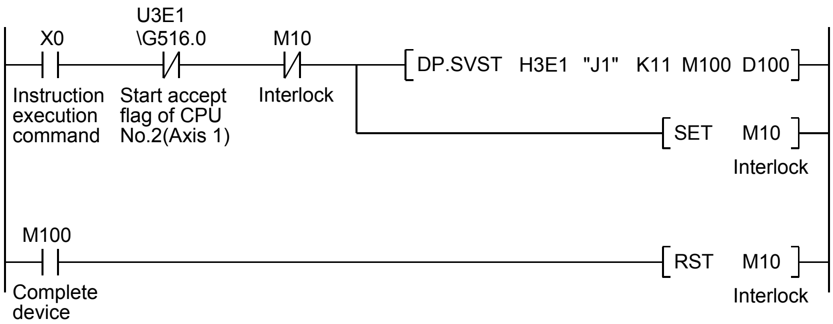
<Example 1> Program which omits the complete device and complete status.



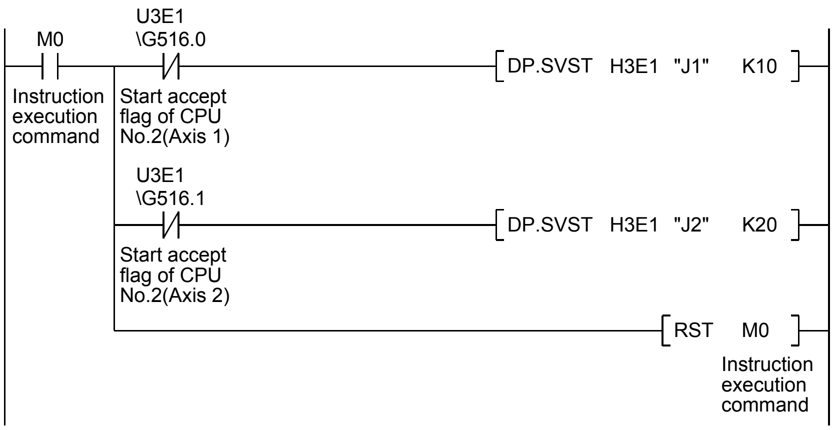
<Example 2> Program which uses the complete device and complete status.



(2) Program which executes continuous start of the servo program No.11 toward Axis 1 of the Motion CPU (CPU No.2), while X0 is ON.



(3) Program which continuously executes the servo program No.10 toward Axis 1 of the Motion CPU (CPU No.2) and the servo program No.20 toward Axis 2, when M0 turned ON.



3.2.3 Current value change instruction from the PLC CPU to the Motion CPU: D(P).CHGA (PLC instruction: D(P).CHGA)

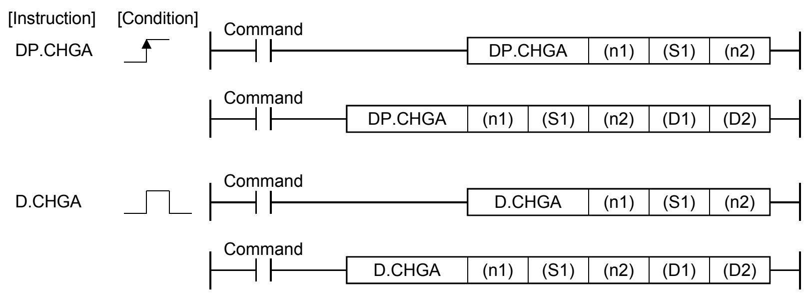
Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): Local devices cannot be used.

(Note-3): Setting data (n1) to (D2): Index qualification possible (except constant)



[Setting data]

<table>
<tr><th>Setting data</th><th>Description</th><th>Set by</th><th>Data type</th></tr>
<tr><td>(n1)</td><td>(First I/O No. of the target CPU)/16
Value to specify actually is the following.
CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H
(Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.</td><td>User</td><td>16-bit binary</td></tr>
<tr><td rowspan="3">(S1)</td><td>Axis No. ("Jn") (Note-2) to execute the current value change.
Q173DSCPU/Q173DCPU(-S1) : J1 to J32
Q172DSCPU : J1 to J16
Q172DCPU(-S1) : J1 to J8</td><td rowspan="3">User</td><td rowspan="3">Character sequence</td></tr>
<tr><td>Synchronous encoder axis No. ("En") (Note-3) to execute the current value change.
Q173DSCPU/Q173DCPU(-S1)/Q172DSCPU : E1 to J12
Q172DCPU(-S1) : E1 to E8</td></tr>
<tr><td>Cam axis No. ("Cn") (Note-2) to execute the current value change within 1 revolution.
Q173DSCPU/Q173DCPU(-S1) : C1 to C32
Q172DSCPU : C1 to C16
Q172DCPU(-S1) : C1 to C8</td></tr>
<tr><td>(n2)</td><td>Current value to change</td><td>User</td><td>32-bit binary</td></tr>
<tr><td>(D1) (Note-1)</td><td>Complete devices
(D1+0): Device which make turn on for one scan at accept completion of instruction.
(D1+1): Device which make turn on for one scan at accept abnormal completion of instruction.
("D1+0" also turns on at the abnormal completion.)</td><td>System</td><td>Bit</td></tr>
<tr><td>(D2) (Note-1)</td><td>Complete status storage device</td><td>System</td><td>Word</td></tr>
</table>

(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

(Note-3): "n" shows the numerical value correspond to synchronous encoder axis No. (n=1 to 12)

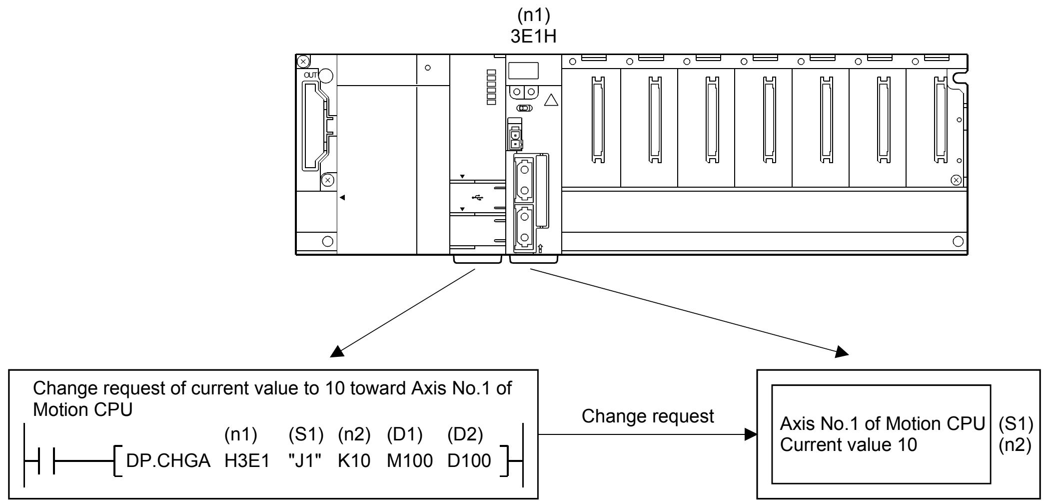
● When axis No. "Jn" is specified with (S1)

[Controls]

(1) The current value change of axis (stopped axis) specified with (S1) is changed to the current value specified with (n2).

(2) It is necessary to take an inter-lock by the start accept flag and user device of CPU shared memory so that multiple instructions may not be executed toward the same axis of same Motion CPU.

(3) The current change value is also possible when the servo program which makes the CHGA instruction toward an axis is executed in the D(P).SVST instruction.

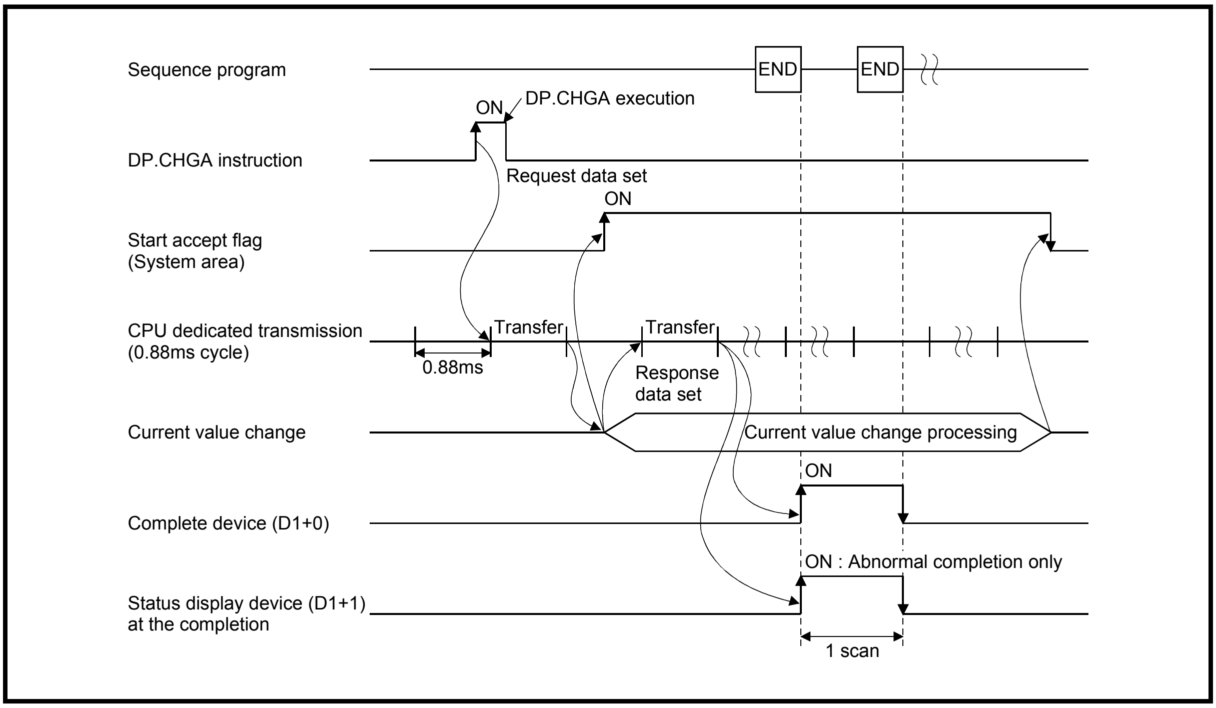


POINT

Refer to Section "3.3 Precautions" for details of the start accept flag.

[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "Jn" as Axis No. is shown below.



[Setting range]

(1) Setting of axis to execute the current value change
The axis to execute the current value change set as (S1) sets J + Axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q173DCPU(-S1)	
Q172DSCPU	1 to 16
Q172DCPU(-S1)	1 to 8

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

[Start accept flag (System area)]

When the instruction is executed by specifying "Jn" as Axis No., the complete status of start accept flag is stored in the address of the start accept flag in the CPU shared memory for target CPU.

<table>
<tr><th>CPU shared memory address
( ) is decimal address</th><th>Description</th></tr>
<tr><td>204H(516)
205H(517)</td><td>The start accept flag for 32 axes are stored corresponding to each bit.
Bits are actually set as the following:
• Q173DSCPU/Q173DCPU(-S1) : J1 to J32
• Q172DSCPU : J1 to J16
• Q172DCPU(-S1) : J1 to J8
OFF: Start accept enable
ON : Start accept disable
<table>
<tr><th></th><th>b15</th><th>b14</th><th></th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>204H(516) address</td><td>J16</td><td colspan="3">• • • • • • • •</td><td>J2</td><td>J1</td></tr>
<tr><td>205H(517) address</td><td>J32</td><td colspan="3">• • • • • • • •</td><td>J18</td><td>J17</td></tr>
</table>
</td></tr>
</table>

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

<table>
<tr><th>Complete status (Note-1)
(Error code)(H)</th><th>Error factor</th><th>Corrective action</th></tr>
<tr><td>0010</td><td>Instruction request to Motion CPU from PLC CPU exceeds the permissible value.</td><td rowspan="3">Confirm a program, and correct it to a correct sequence program.</td></tr>
<tr><td>2100</td><td>There are the following number or more simultaneous D(P).SVST/ D(P).CHGA/D(P).CHGAS QDS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.
• Q173DSCPU/Q172DSCPU : 129 or more (Note-2)
• Q173DCPU(-S1)/Q172DCPU(-S1) : 65 or more</td></tr>
<tr><td>2203</td><td>Axis No. set by D(P).CHGA instruction is wrong.</td></tr>
</table>

(Note-1): 0000H (Normal)
(Note-2): 65 or more for operating system software version "00A".

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

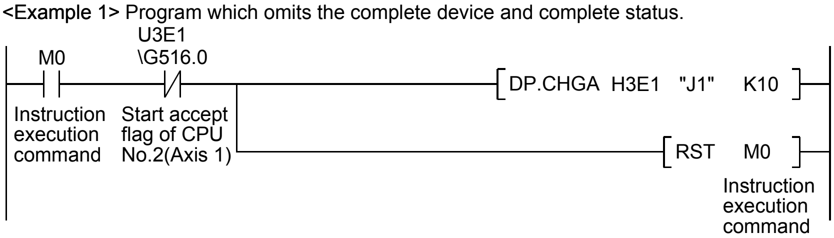
Error code(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

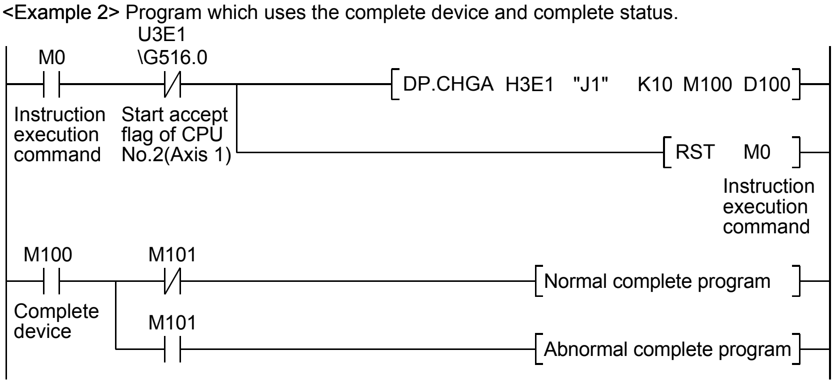
[Program example]

(1) Program which changes the current value to 10 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

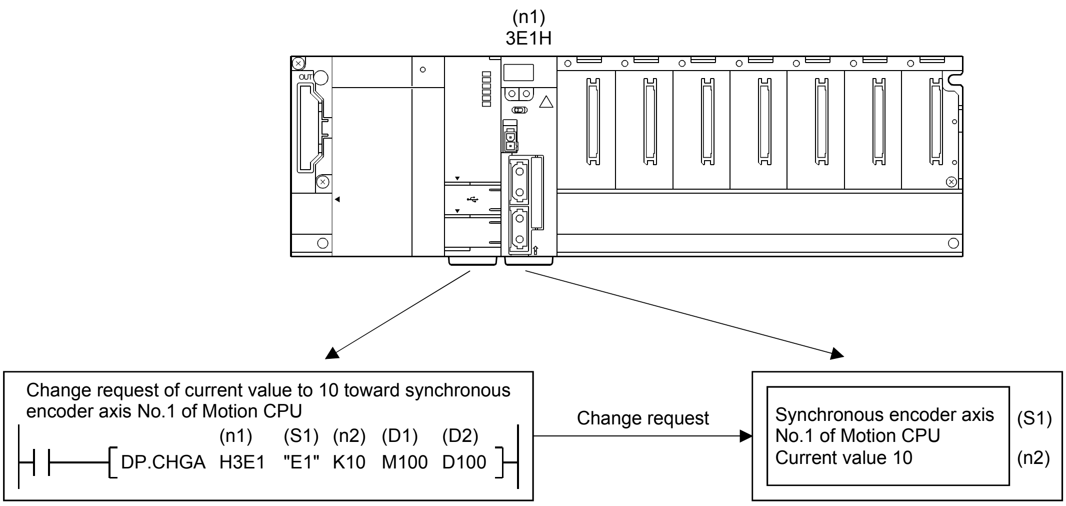


● When axis No. "En" is specified with (S1)

[Controls]

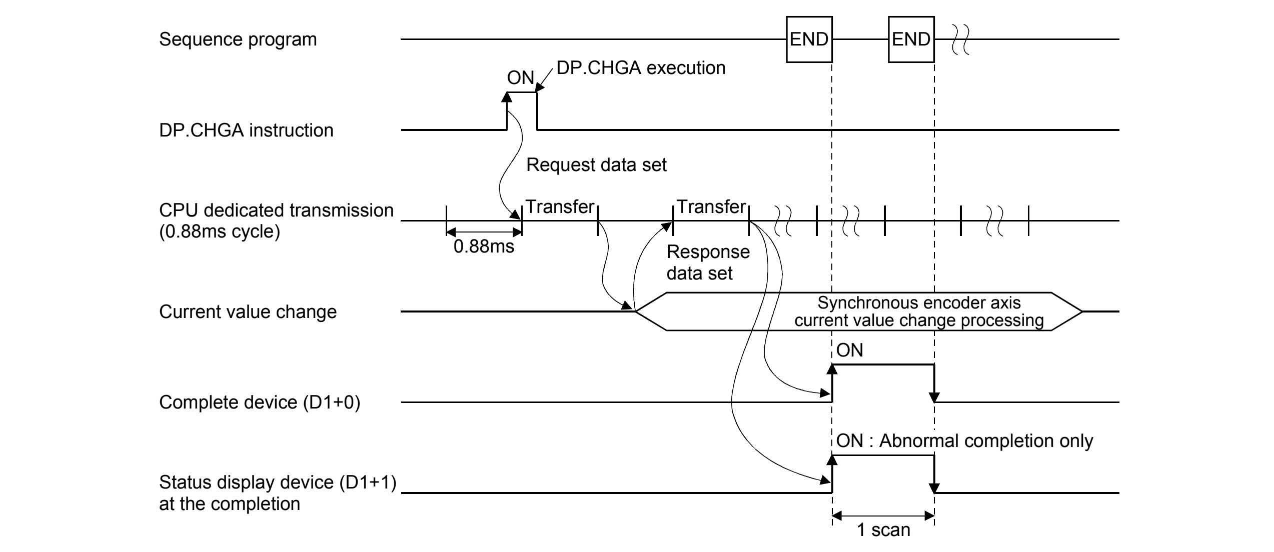
(1) The synchronous encoder axis current value specified with (S1) is changed to the current value specified with (n2) in the virtual mode.
(The current value change can be executed in real mode for the version (Refer to Section 1.3) that supports "incremental synchronous encoder current value in real mode".)

(2) There is not an interlock signal for status of synchronous encoder current value change.
When the multiple instructions are executed toward the same synchronous encoder axis of same Motion CPU, the current value is changed to specified value by last instruction.



[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "En" as Axis No. is shown below.



[Setting range]

(1) Setting of synchronous encoder axis to execute the current value change
The synchronous encoder axis to execute the current value change set as (S1) sets E + synchronous encoder axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 12
Q173DCPU(-S1)	
Q172DSCPU	
Q172DCPU(-S1)	1 to 8

The number of axes which can set are only 1 axis.
Set "E" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note-1) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2002	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed. (Axis No. "En" was specified by operating system software except SV22.)	
2100	There are the following number or more simultaneous D(P).SVST/ D(P).CHGA/D(P).CHGAS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • Q173DSCPU/Q172DSCPU : 129 or more (Note-2) • Q173DCPU(-S1)/Q172DCPU(-S1) : 65 or more	
2203	Axis No. set by D(P).CHGA instruction is wrong.	

(Note-1): 0000H (Normal)
(Note-2): 65 or more for operating system software version "00A".

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

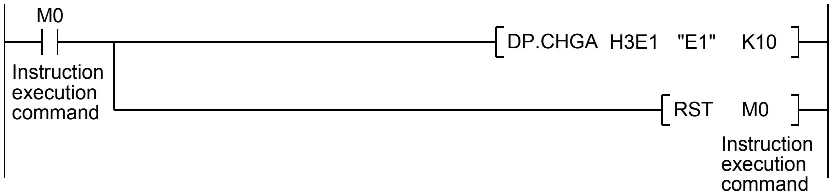
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

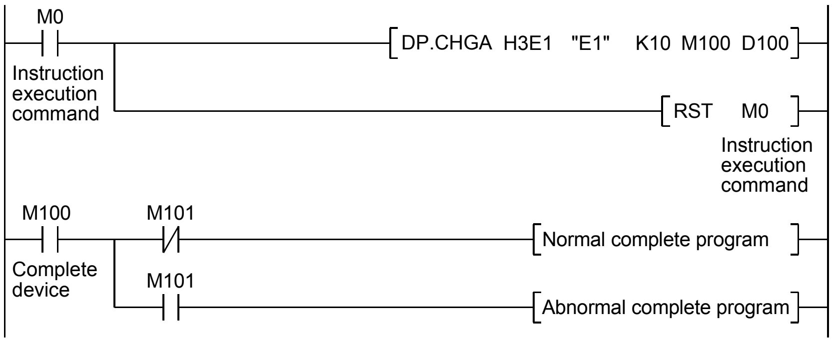
[Program example]

(1) Program which changes the current value to 10 for synchronous encoder axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

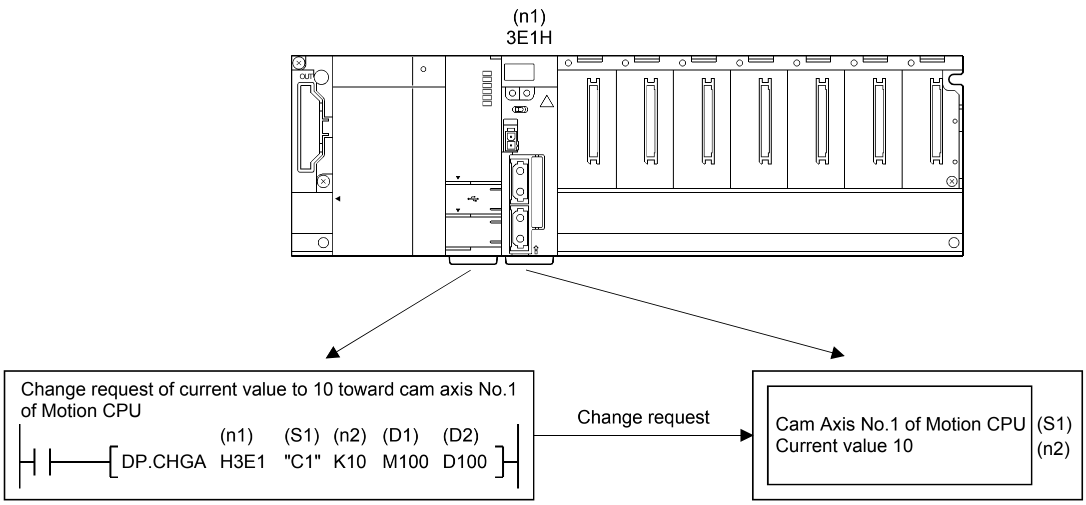


● When axis No. "Cn" is specified with (S1)

[Controls]

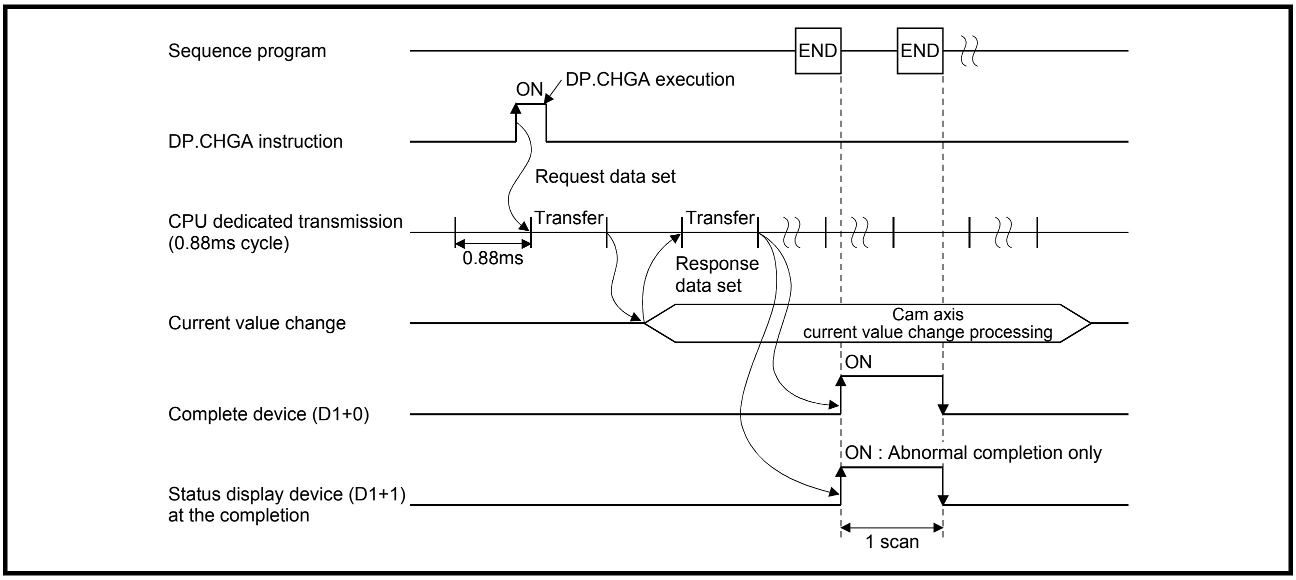
(1) The current value within 1 cam shaft revolution specified with (S1) is changed to the current value specified with (n2) in the virtual mode.

(2) There is not an interlock signal for status of current value within 1 cam shaft revolution change.
When the multiple instructions are executed toward the same cam axis of same Motion CPU, the current value is changed to specified value by last instruction.



[Operation]

Outline operation between CPUs at the DP.CHGA instruction execution by specifying "Cn" as Axis No. is shown below.



[Setting range]

(1) Setting of cam axis to execute the current value change within 1 cam shaft revolution

The cam axis to execute the current value change within 1 cam shaft revolution set as (S1) sets C + cam axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q173DCPU(-S1)	
Q172DSCPU	1 to 16
Q172DCPU(-S1)	1 to 8

The number of axes which can set are only 1 axis.
Set "C" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note-1) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2002	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed. (Axis No. "En" was specified by operating system software except SV22.)	
2100	There are the following number or more simultaneous D(P).SVST/ D(P).CHGA/D(P).CHGAS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • Q173DSCPU/Q172DSCPU : 129 or more (Note-2) • Q173DCPU(-S1)/Q172DCPU(-S1) : 65 or more	
2203	Axis No. set by D(P).CHGA instruction is wrong.	

(Note-1): 0000H (Normal)
(Note-2): 65 or more for operating system software version "00A".

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

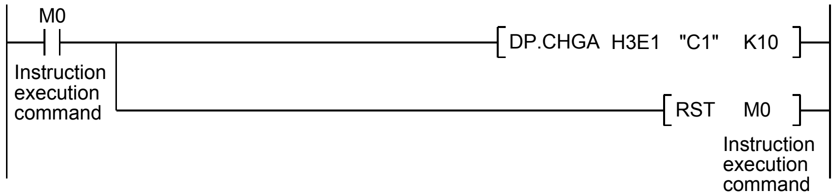
Error code(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

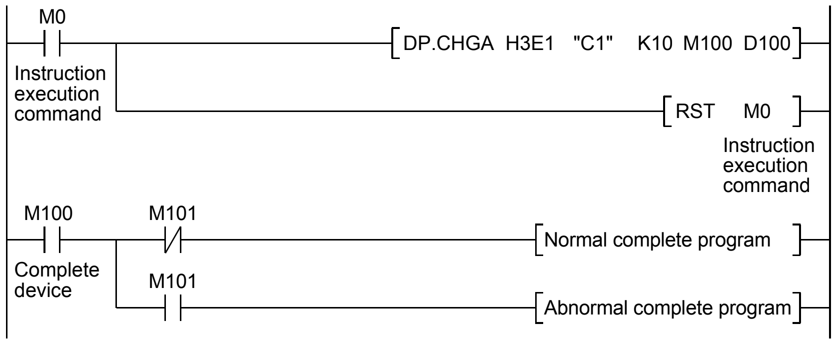
[Program example]

(1) Program which changes the current value to 10 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

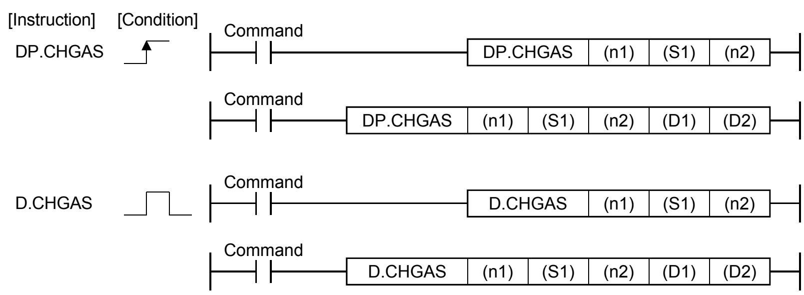


3.2.4 Current value change instruction of command generation axis from the PLC CPU to the Motion CPU: D(P).CHGAS (PLC instruction: D(P).CHGAS) (SV22 advanced synchronous control only)

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") (Note-2) to execute the current value change of command generation axis. Q173DSCPU: J1 to J32 Q172DSCPU: J1 to J16	User	Character sequence
(n2)	Current value to change	User	32-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

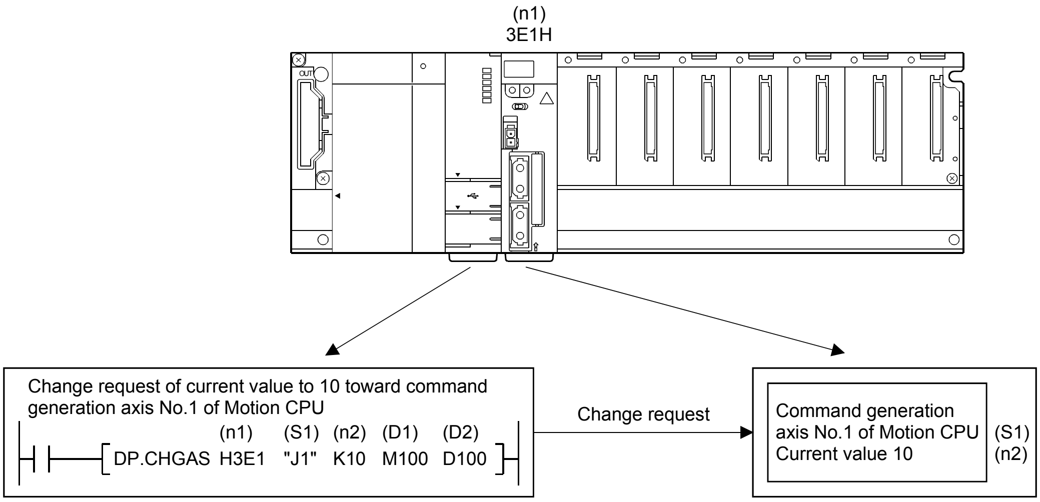
(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) The current value change of command generation axis (stopped axis) specified with (S1) is changed to the current value specified with (n2).

(2) It is necessary to take an inter-lock by the start accept flag and user device of CPU shared memory so that multiple instructions may not be executed toward the same axis of same Motion CPU.

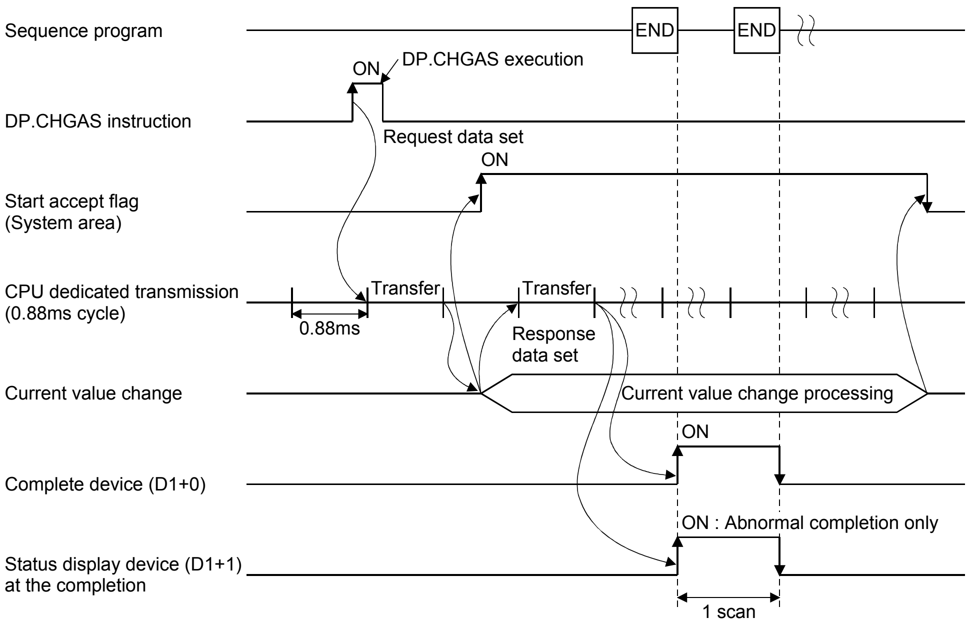
(3) The current change value is also possible when the servo program which makes the CHGA instruction toward an command generation axis is executed in the D(P).SVST instruction.



POINT
Refer to Section "3.3 Precautions" for details of the start accept flag.

[Operation]

Outline operation between CPUs at the DP.CHGAS instruction execution by specifying "Jn" as Axis No. is shown below.



[Setting range]

(1) Setting of axis to execute the current value change

The axis to execute the current value change set as (S1) sets J + Axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q172DSCPU	1 to 16

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the command generation axis parameter as the axis No. to start.
Refer to the "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for command generation axis parameter.

(2) Setting of the current value to change

(n2) usable range
-2147483648 to 2147483647

[Start accept flag (System area)]

When the instruction is executed by specifying "Jn" as Axis No., the complete status of start accept flag is stored in the address of the start accept flag in the CPU shared memory for target CPU.

CPU shared memory address ( ) is decimal address	Description
20EH(526) 20FH(527) QDS	The command generation axis start accept flag for 32 axes are stored corresponding to each bit. Bits are actually set as the following: • Q173DSCPU: J1 to J32 • Q172DSCPU: J1 to J16 OFF : Start accept enable ON : Start accept disable

	b15	b14 … b2	b1	b0
20EH(526) address	J16	• • • • • • • •	J2	J1
20FH(527) address	J32	• • • • • • • •	J18	J17

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note-1) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2100	There are the following number or more simultaneous D(P).SVST/ D(P).CHGA/D(P).CHGAS QDS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • Q173DSCPU/Q172DSCPU: 129 or more	
2207	Axis No. set by D(P).CHGAS instruction is wrong.	

(Note-1): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

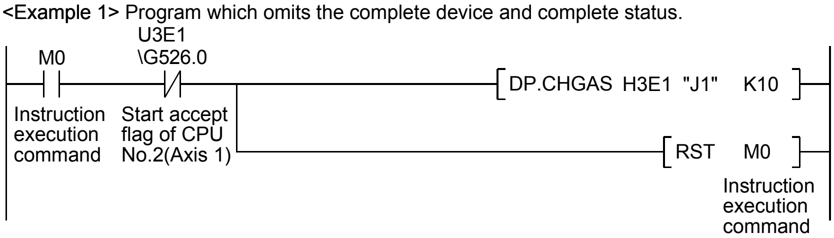
Error code(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

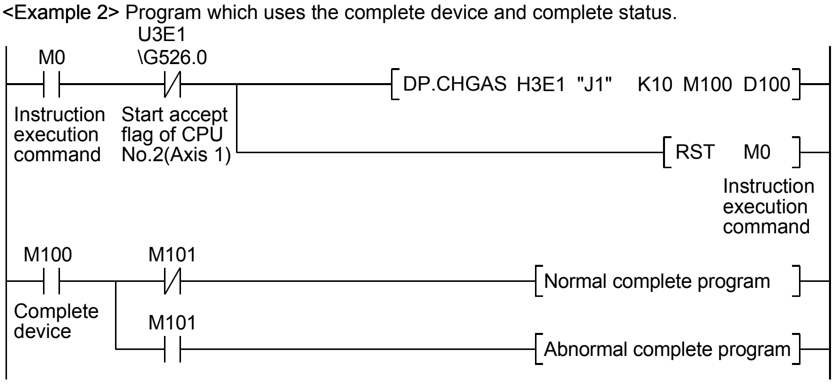
[Program example]

(1) Program which changes the current value to 10 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

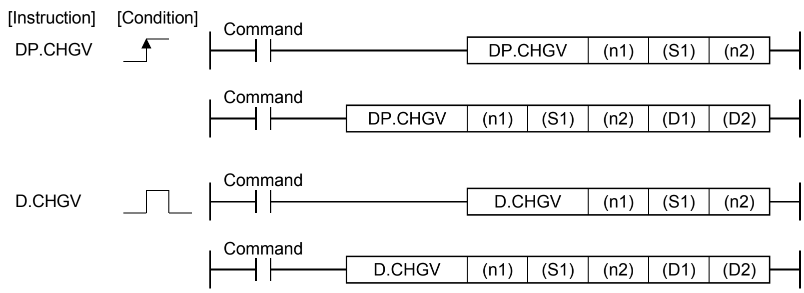


3.2.5 Speed change instruction from the PLC CPU to the Motion CPU: D(P).CHGV (PLC instruction: D(P).CHGV)

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") (Note-2) to execute the speed change. Q173DSCPU/Q173DCPU(-S1) : J1 to J32 Q172DSCPU : J1 to J16 Q172DCPU(-S1) : J1 to J8	User	Character sequence
(n2)	Speed to change	User	32-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

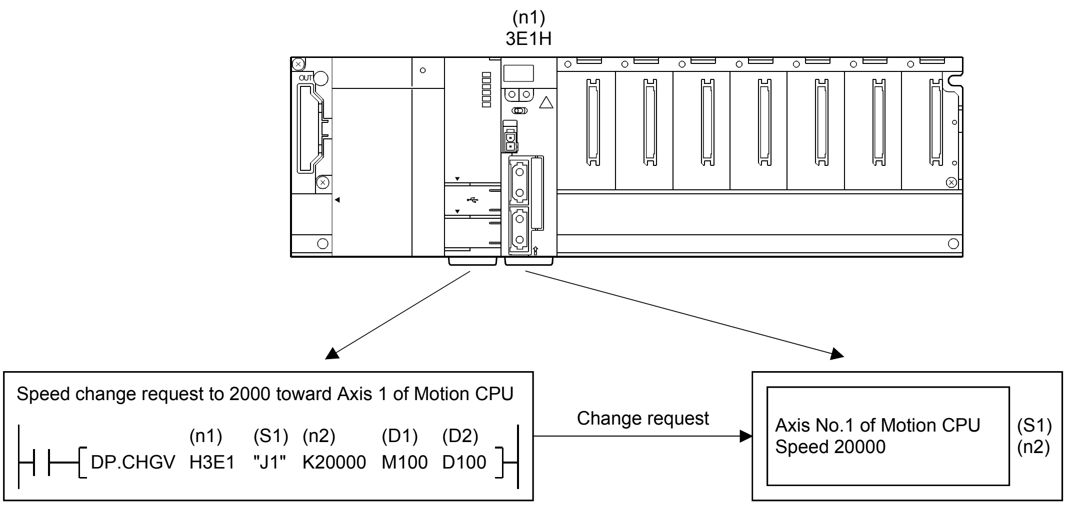
(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) The speed of axis specified with (S1) is changed to the speed specified with (n2) during positioning or JOG operating.

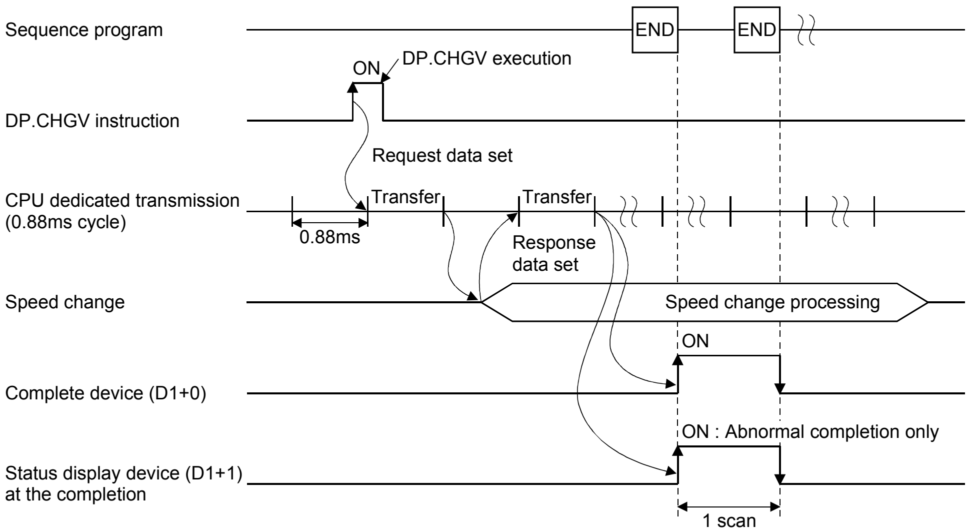
(2) There is not an interlock signal on the shared memory during speed change. When the multiple instructions are executed toward the same axis of same Motion CPU, the speed is changed to specified value by last instruction.



(3) Acceleration/deceleration time at speed change can be changed by setting the acceleration/deceleration time change parameter of the axis specified with (S1). QDS Ver.!
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for acceleration/deceleration time change parameter and acceleration/deceleration time change function.

[Operation]

Outline operation between CPUs at the DP.CHGV instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the speed change
The axis to execute the speed change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q173DCPU(-S1)	
Q172DSCPU	1 to 16
Q172DCPU(-S1)	1 to 8

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(2) Setting of the speed to change

	(n2) usable range
mm	-600000000 to 600000000$\times 10^{-2}$[mm/min]
inch	-600000000 to 600000000$\times 10^{-3}$[inch/min]
degree	-2147483647 to 2147483647$\times 10^{-3}$[degree/min]
pulse	-2147483647 to 2147483647[pulse/s]

(Note): When the "speed control 10×multiplier speed setting for degree axis" is set to "valid", the setting range is -2147483647 to 2147483647$\times 10^{-2}$ [degree/min].

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2204	Axis No. set by D(P).CHGV instruction is wrong.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

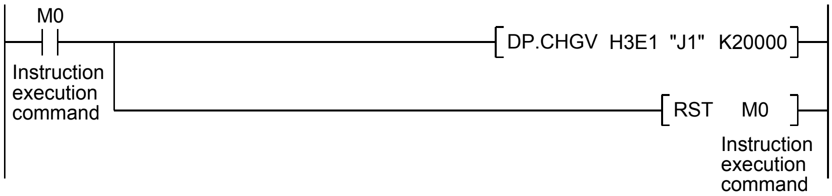
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

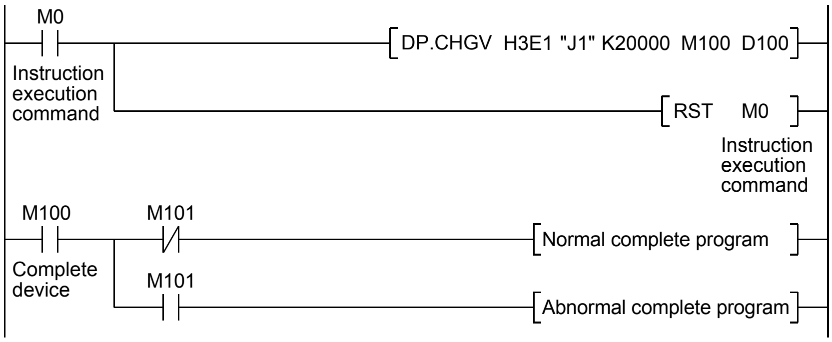
[Program example]

(1) Program which changes the positioning speed to 20000 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

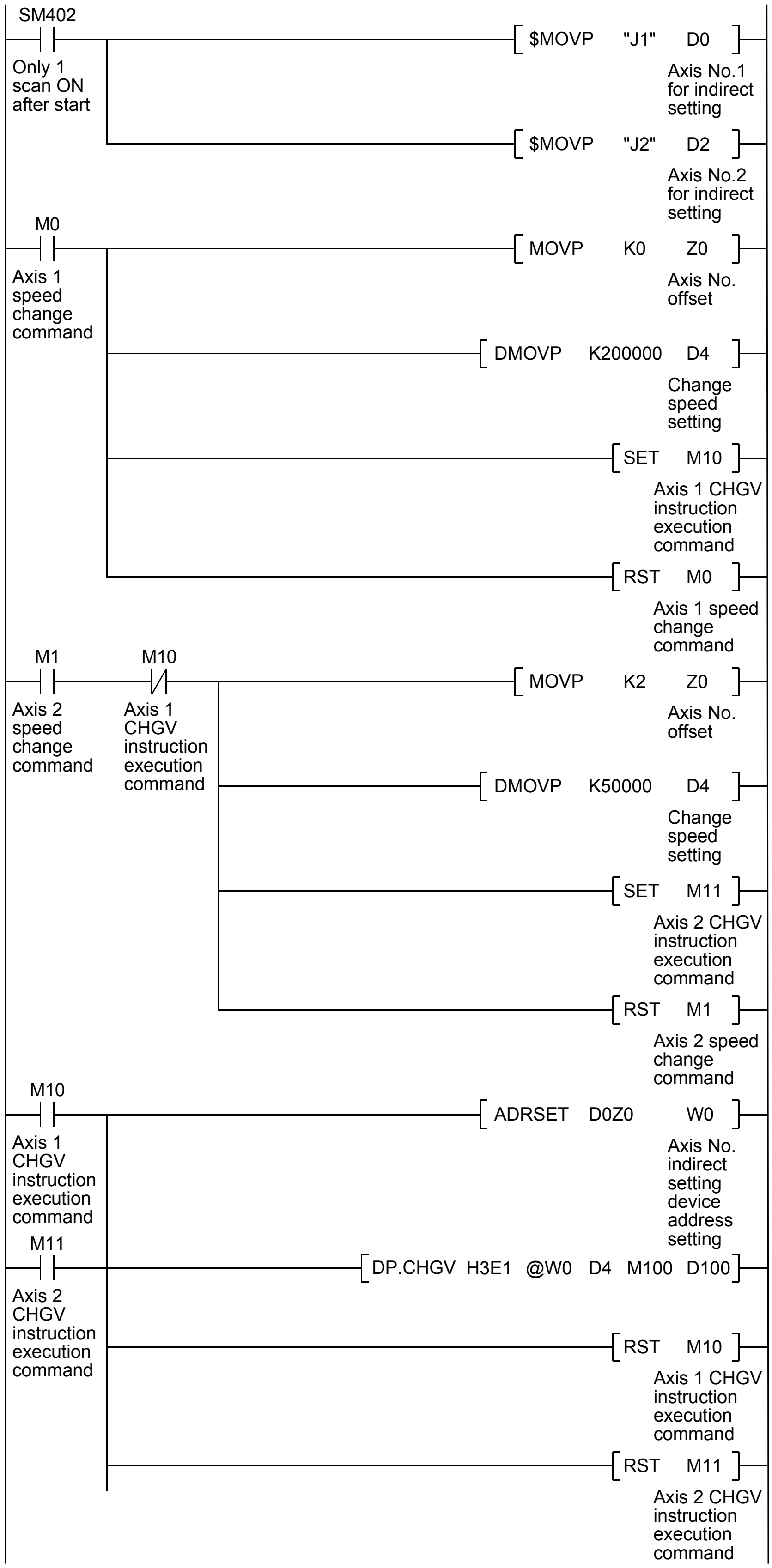
<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



(2) Program which changes the positioning speed to 200000 for Axis 1 of the Motion CPU (CPU No.2), when M0 that sets Axis No. as indirect setting method turned ON, and then changes the positioning speed to 50000 for Axis 2, when M1 turned ON.

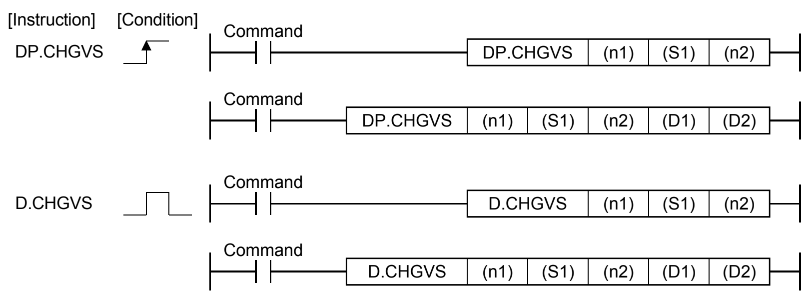


3.2.6 Speed change instruction of command generation axis from the PLC CPU to the Motion CPU: D(P).CHGVS (PLC instruction: D(P).CHGVS) (SV22 advanced synchronous control only) QDS Ver.!

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") (Note-2) to execute the speed change. Q173DSCPU: J1 to J32 Q172DSCPU: J1 to J16	User	Character sequence
(n2)	Speed to change	User	32-bit binary
(D1) (Note-1)	Complete devices (D1+0) : Device which make turn on for one scan at accept completion of instruction. (D1+1) : Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

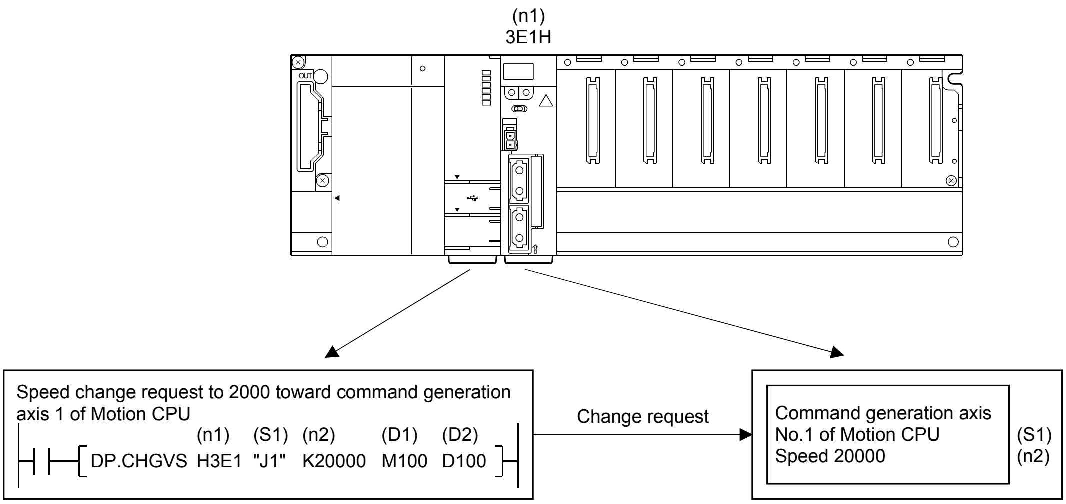
(Note-1): Omission possible with both of (D1) and (D2) omission.

(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) The speed of command generation axis specified with (S1) is changed to the speed specified with (n2) during positioning or JOG operating.

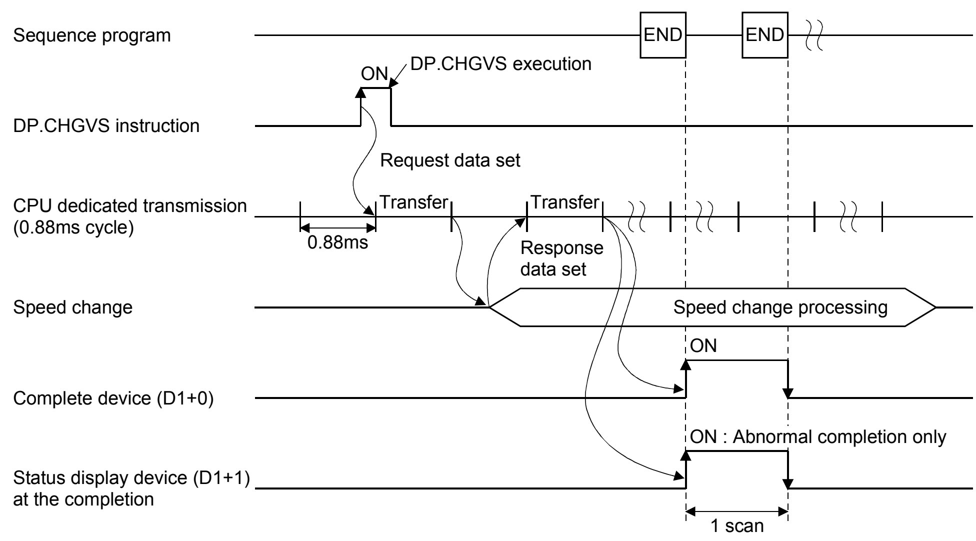
(2) There is not an interlock signal on the shared memory during speed change. When the multiple instructions are executed toward the same axis of same Motion CPU, the speed is changed to specified value by last instruction.



(3) Acceleration/deceleration time at speed change can be changed by setting the acceleration/deceleration time change parameter of the axis specified with (S1).
Refer to the "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for acceleration/deceleration time change parameter.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for acceleration/deceleration time change function.

[Operation]

Outline operation between CPUs at the DP.CHGVS instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the speed change
The axis to execute the speed change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q172DSCPU	1 to 16

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the command generation axis parameter as the axis No. to start.
Refer to the "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for command generation axis parameter.

(2) Setting of the speed to change

	(n2) usable range
mm	-600000000 to 600000000$\times 10^{-2}$[mm/min]
inch	-600000000 to 600000000$\times 10^{-3}$[inch/min]
degree	-2147483647 to 2147483647$\times 10^{-3}$[degree/min]
pulse	-2147483647 to 2147483647[pulse/s]

(Note): When the "speed control 10×multiplier speed setting for degree axis" is set to "valid", the setting range is -2147483647 to 2147483647$\times 10^{-2}$ [degree/min].

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2208	Axis No. set by D(P).CHGVS instruction is wrong.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

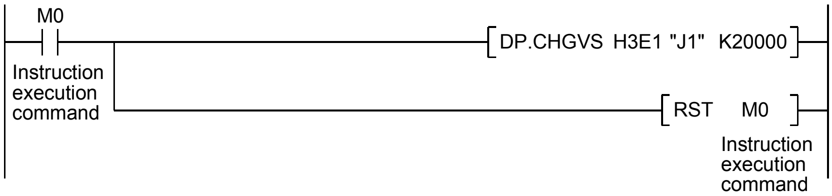
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

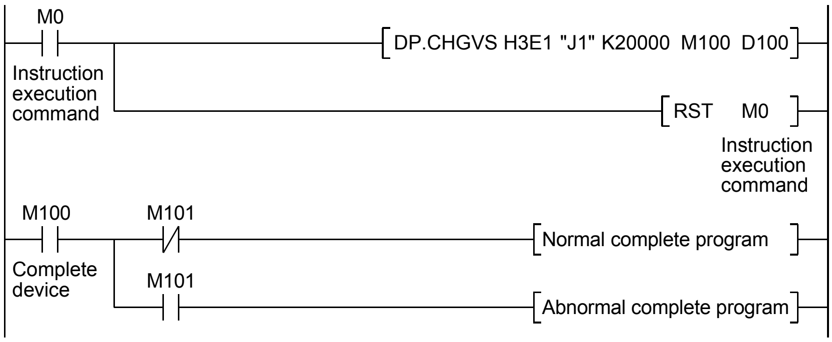
[Program example]

(1) Program which changes the positioning speed to 20000 for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

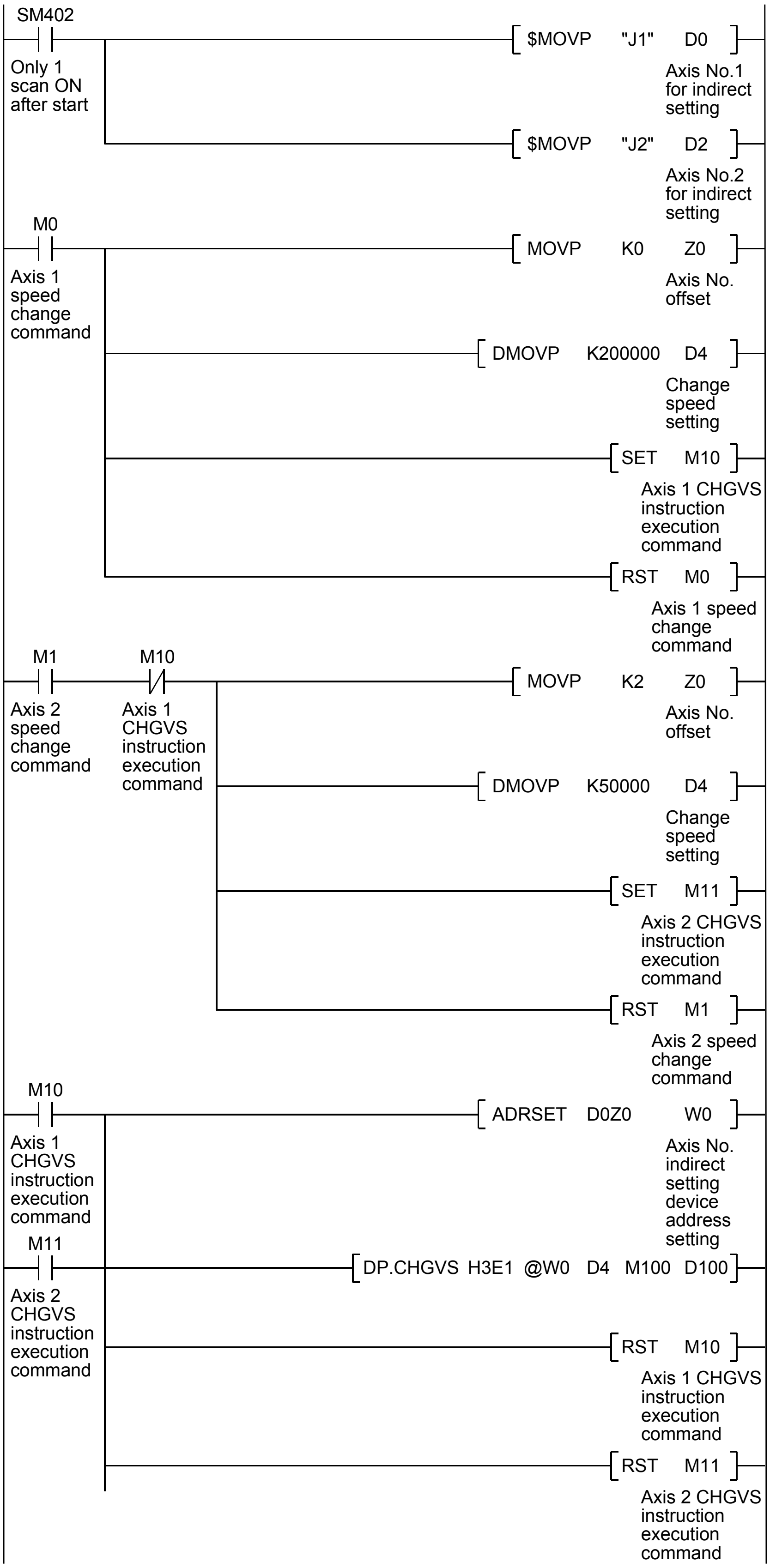
<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



(2) Program which changes the positioning speed to 200000 for Axis 1 of the Motion CPU (CPU No.2), when M0 that sets Axis No. as indirect setting method turned ON, and then changes the positioning speed to 50000 for Axis 2, when M1 turned ON.

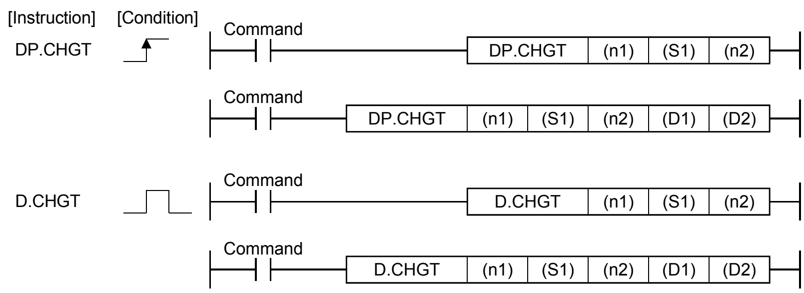


3.2.7 Torque limit value change request instruction from the PLC CPU to the Motion CPU: D(P).CHGT (PLC instruction: $\boxed{\text{D(P).CHGT}}$)

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") (Note-2) to execute the torque limit value change. Q173DSCPU/Q173DCPU(-S1) : J1 to J32 Q172DSCPU : J1 to J16 Q172DCPU(-S1) : J1 to J8	User	Character sequence
(n2)	Torque limit value to change [%]	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

(Note-1): Omission possible with both of (D1) and (D2) omission.

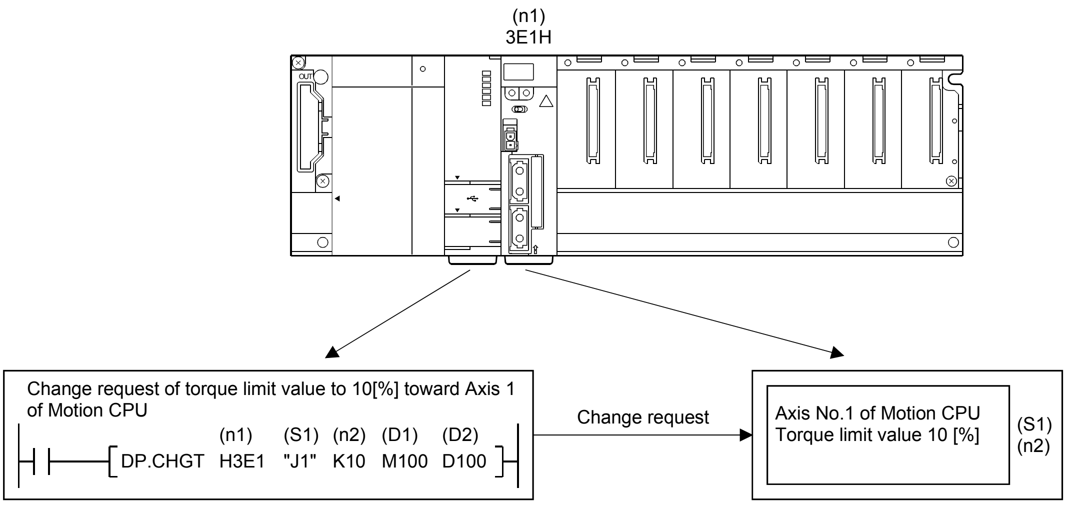
(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) The torque limit value of axis specified with (S1) is changed to the value specified with (n2) for the positive direction and negative direction regardless of while being operating or stopping in the real mode.

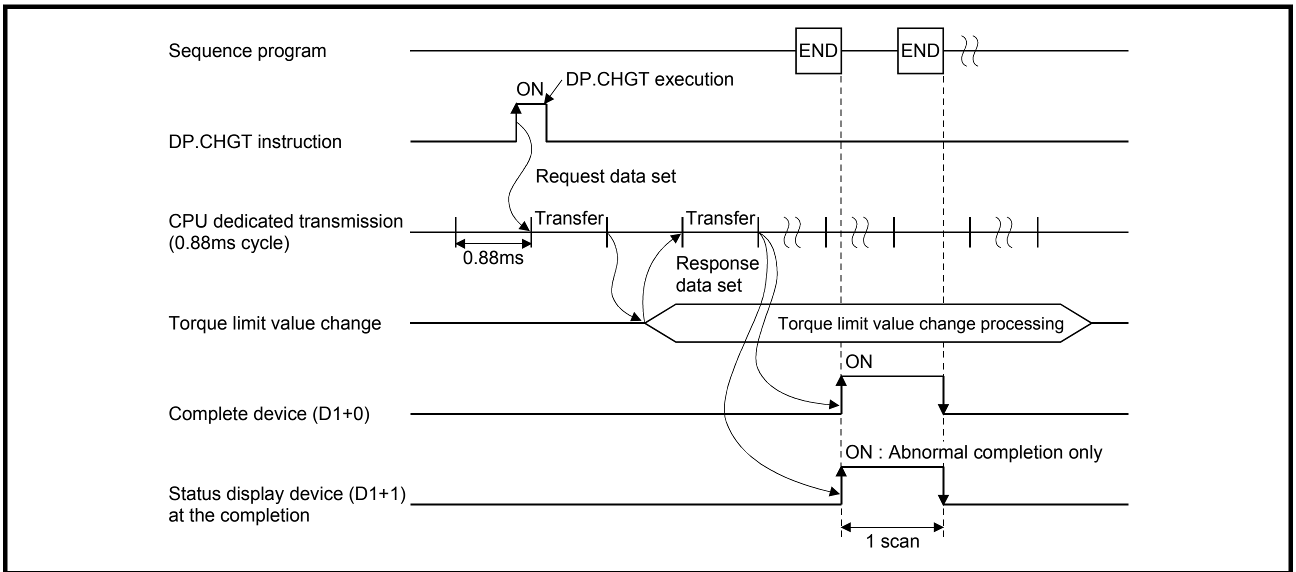
(2) There is not an interlock signal for status of axis torque change.
When the multiple instructions are executed toward the same axis of same Motion CPU, the torque is changed to specified value by last instruction.

(3) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for relation between torque limit value specified with servo program and torque limit value change instruction.



[Operation]

Outline operation between CPUs at the DP.CHGT instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the torque limit value change
The axis to execute the torque limit change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q173DCPU(-S1)	
Q172DSCPU	1 to 16
Q172DCPU(-S1)	1 to 8

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the torque limit value to change

(n2) usable range	Unit
1 to 1000	[%]

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2205	Axis No. set by D(P).CHGT instruction is wrong.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

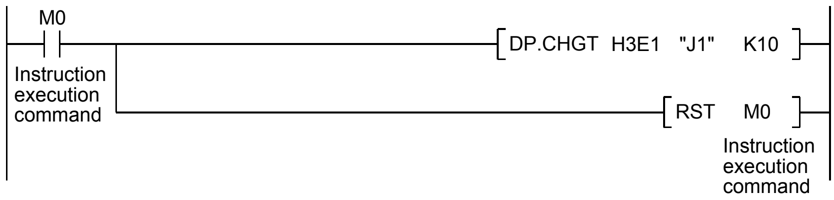
Error code(Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

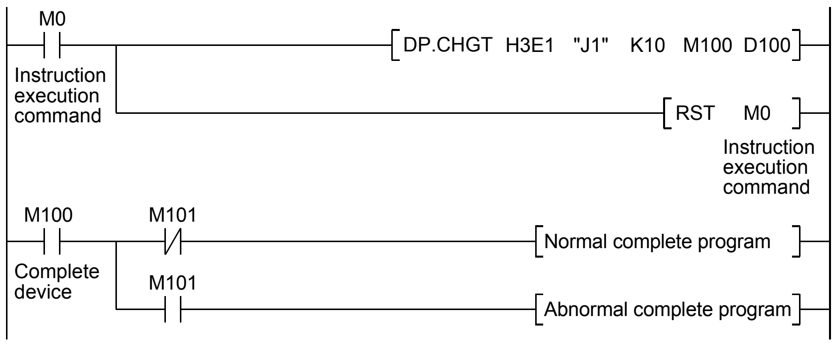
[Program example]

(1) Program which changes the torque limit value to 10[%] for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.

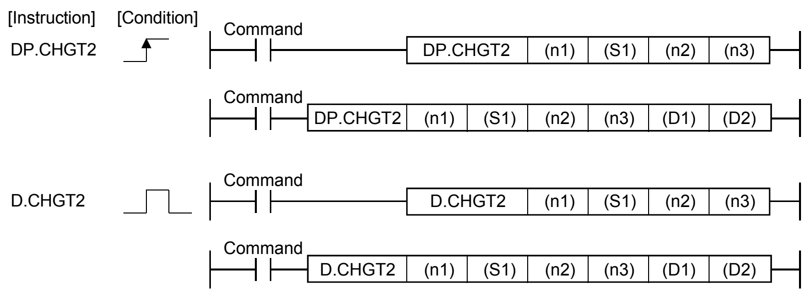


3.2.8 Torque limit value individual change request instruction from the PLC CPU to the Motion CPU: D(P).CHGT2 (PLC instruction: D(P).CHGT2) QDS

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		○		○							○	
(n2)		○		○						○		
(n3)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Axis No. ("Jn") (Note-2) to execute the torque limit value change. Q173DSCPU: J1 to J32 Q172DSCPU: J1 to J16	User	Character sequence
(n2)	Positive direction torque limit value to change individually (×0.1[%])	User	16-bit binary
(n3)	Negative direction torque limit value to change individually (×0.1[%])	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

(Note-1): Omission possible with both of (D1) and (D2) omission.

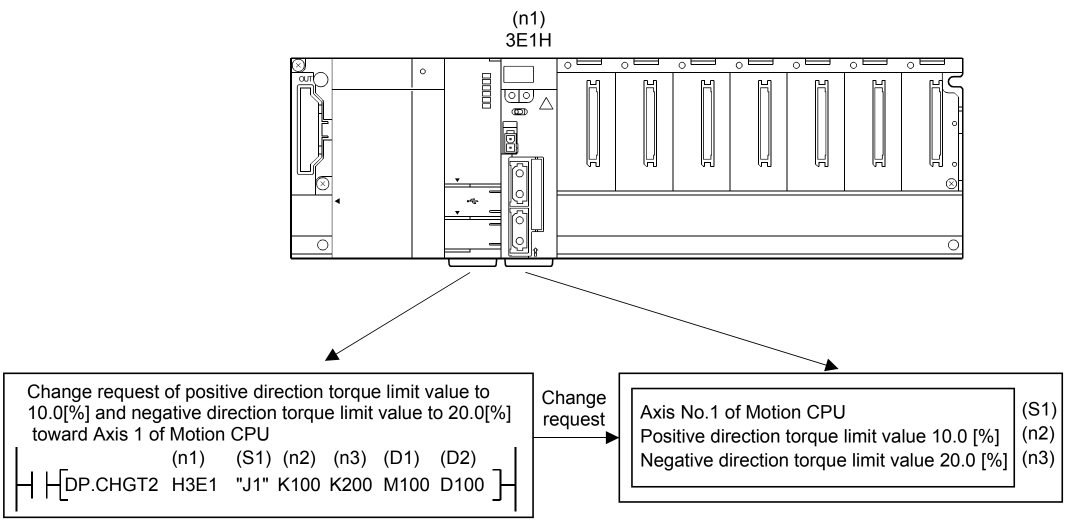
(Note-2): "n" shows the numerical value correspond to axis No. (n=1 to 32)

[Controls]

(1) The torque limit value of axis specified with (S1) is changed to the positive direction torque limit value specified with (n2) and negative direction torque limit value specified with (n3) regardless of while being operating or stopping.

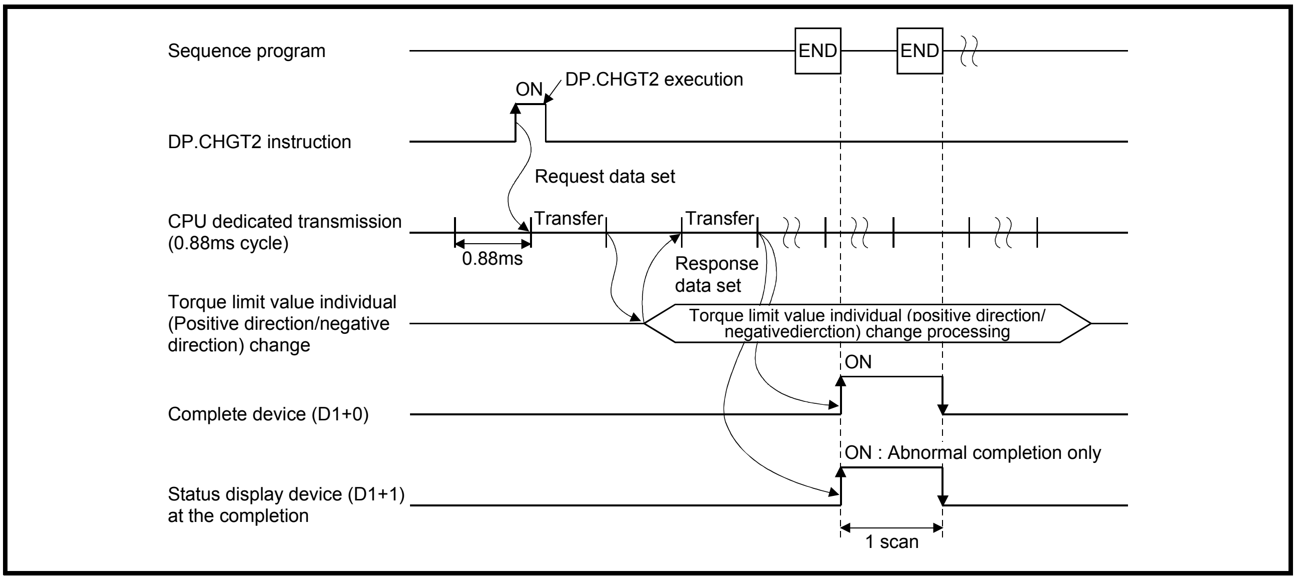
(2) There is not an interlock signal for status of axis torque change.
When the multiple instructions are executed toward the same axis of same Motion CPU, the torque is changed to specified value by last instruction.

(3) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for relation between torque limit value specified with servo program and torque limit value change instruction.



[Operation]

Outline operation between CPUs at the DP.CHGT2 instruction execution is shown below.



[Setting range]

(1) Setting of axis to execute the torque limit value change
The axis to execute the torque limit change set as (S1) sets J + axis No. in a character sequence " ".

	(S1) usable range
Q173DSCPU	1 to 32
Q172DSCPU	1 to 16

The number of axes which can set are only 1 axis.
Set "J" in a capital letter and use the axis No. set in the system setting as the axis No. to start.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for system settings.

(2) Setting of the torque limit value to change

(n2), (n3) usable range	Unit
1 to 10000	0.1[%]

When the positive or negative direction torque limit is not changed, the torque limit value before change is continued for the set direction by setting "-1" in (n2) or (n3).

[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2206	Axis No. set by D(P).CHGT2 instruction is wrong.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

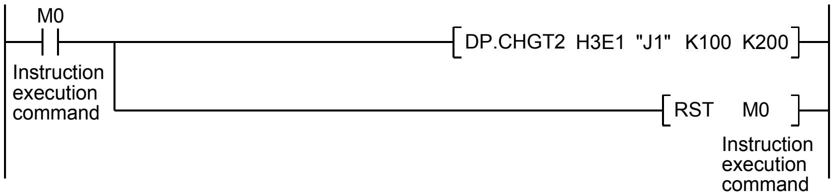
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	

(Note): 0 (Normal)

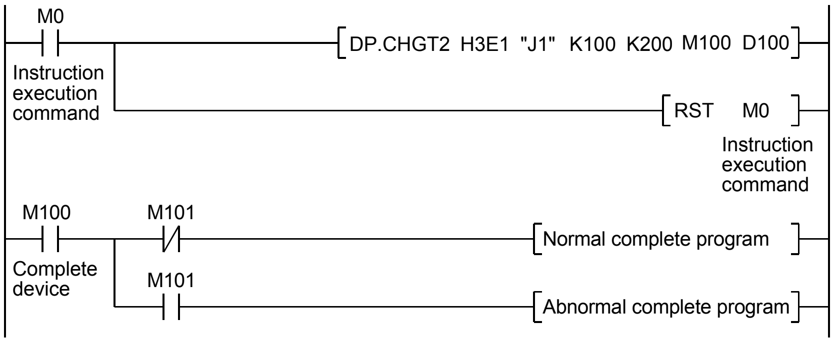
[Program example]

(1) Program which changes the positive torque limit value to 10.0[%] and negative torque limit value to 20.0[%] for Axis 1 of the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



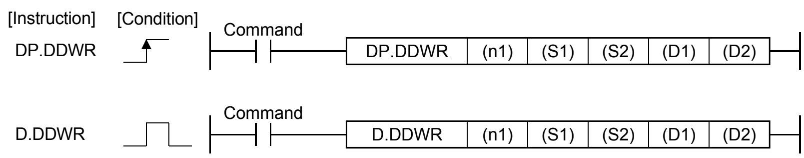
3.2.9 Write device data of the self CPU to the device of other CPU: D(P).DDWR (PLC instruction: D(P).DDWR)

Setting data (Note-2)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		△ (Note-1)		△ (Note-1)								
(S2)		○		○								
(D1)		○									○	
(D2)	△ (Note-1)		△ (Note-1)									

○: Usable △: Usable partly

(Note-1): Local devices cannot be used.

(Note-2): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Start device of the self CPU where control data are stored.	User	Word
(S2)	Start device of the self CPU where writing data are stored.	User	Word
(D1)	Start device of the target Motion CPU that stores writing data. POINT Data can be written in device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting it by a character sequence " ".	User	Word/ Character sequence
(D2)	Complete devices (D2+0): Device which make turn on for one scan at accept completion of instruction. (D2+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D2+0" also turns on at the abnormal completion.)	System	Bit

[Control data]

Device	Item	Setting data	Setting range	Set by
S1+0	Complete status	The status at the instruction completion is stored. 0 : No error (Normal completion) Except 0: Error code	—	System
S1+1	Number of writing data	Set the number of writing data with each word	1 to 20	User

[Controls]

(1) A part for the number of writing data of the control data specified with (S1+1) of data since the device specified with (S2) of the self CPU are stored to since the word device specified with (D1) of the target CPU (n1) in the Multiple CPU system.

(2) Figure specification of the bit device is possible for (S2) and (D1). However, figure specification is 4 figures and a start bit device number is only the multiple of 16. It becomes MULTI-COM.ERROR (Error code: 4353) when other values are specified.

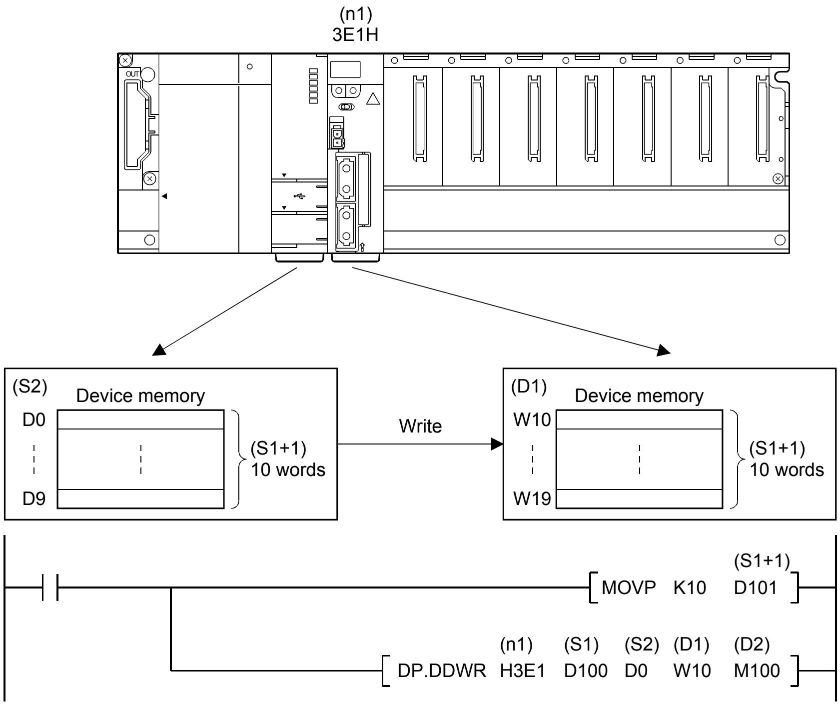
(3) Data can be written in device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting (D1) by a character sequence " ".

(4) D(P).DDWR instruction accepting and normal/abnormal completion can be confirmed with the complete device (D2) or status display device (D2+1) at the completion.

(a) Complete device
It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.

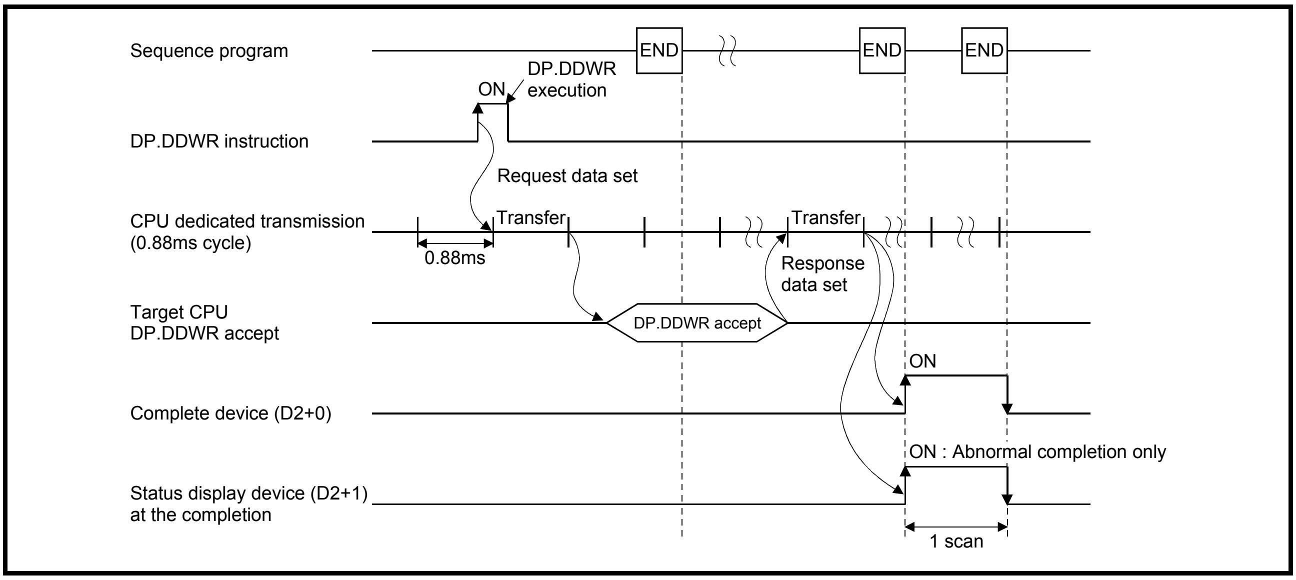
(b) Status display device at the completion
It is turned on/off according to the status of the instruction completion.
- Normal completion : OFF
- Abnormal completion : It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
(Error code is stored in control data (S1+0: Complete status).)

(5) There is a limitation for number of simultaneous instruction execution/ simultaneous acceptance in the Motion dedicated PLC instruction. (Refer to Section 3.3 (2).)
Exchange a large amount of data through the CPU shared memory.



[Operation]

Outline operation between CPUs at the DP.DDWR instruction execution is shown below.



[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (S0+0).

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2001	The specified device cannot be used in the Motion CPU, or it is outside the device range.	
2080	Number of writing data points set by D(P).DDWR instruction is wrong.	
2100	There are 65 or more simultaneous D(P).DDRD/D(P).DDWR instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note): 0000H (Normal)

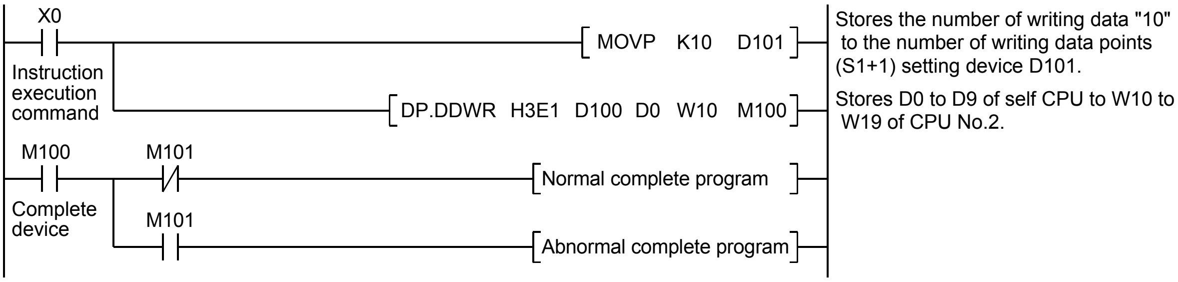
The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code(Note)	Error factor	Corrective action
4101	Number of writing data exceeded the range of storage device.	Confirm a program, and correct it to a correct sequence program.
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	
4355	Number of writing data is outside the range of 1 to 20.	

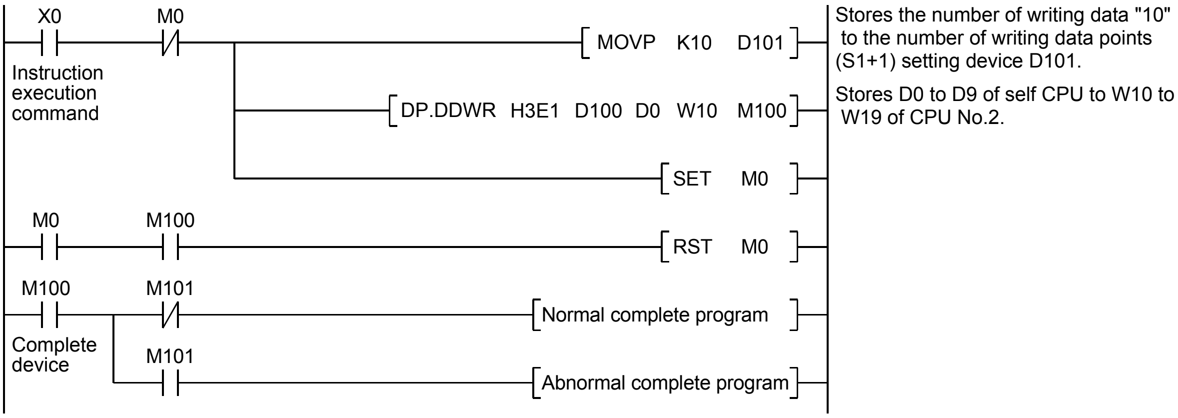
(Note): 0 (Normal)

[Program example]

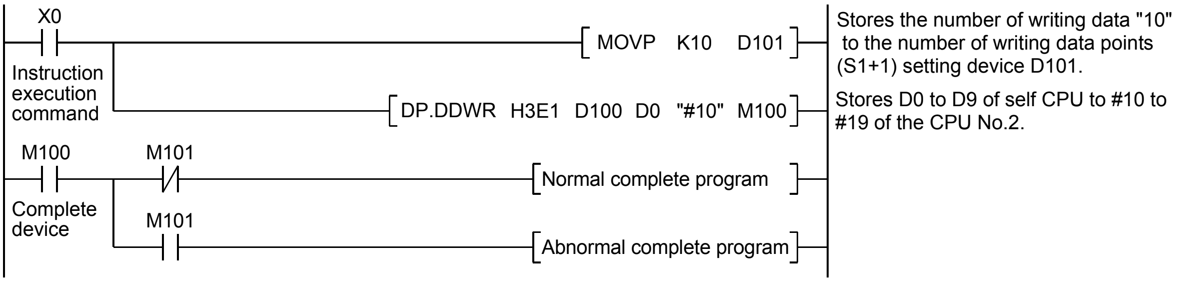
(1) Program which stores data for 10 words from D0 of the self CPU to W10 or later of the CPU No.2, when X0 turned ON.



(2) Program which stores simultaneously data for 10 words from D0 of the self CPU to W10 or later of the CPU No.2, while X0 is ON.



(3) Program which stores data for 10 words from D0 of the self CPU to #10 or later of the CPU No.2, when X0 turned ON.



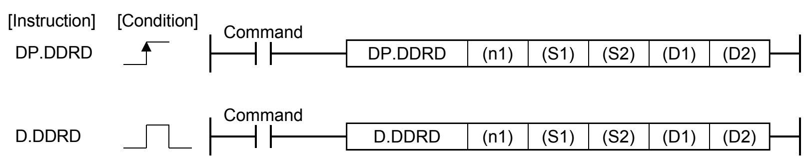
3.2.10 Read device data of other CPU to the device of self CPU: D(P).DDRD (PLC instruction: D(P).DDRD)

Setting data (Note-2)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(S1)		△ (Note-1)		△ (Note-1)								
(S2)		○									○	
(D1)		△ (Note-1)		△ (Note-1)								
(D2)	△ (Note-1)		△ (Note-1)									

○: Usable △: Usable partly

(Note-1): Local devices cannot be used.

(Note-2): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H (Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(S1)	Start device of the self CPU where control data is stored.	User	Word
(S2)	Start device of the target CPU where reading data is stored. POINT Data can be read from device like a motion register (#) etc. of Motion CPU outside the range in the PLC CPU that executes this instruction, by setting it by a character sequence " ".	User	Word/ Character sequence
(D1)	Start device of the self CPU which stores the reading data.	User	Word
(D2)	Complete devices (D2+0): Device which make turn on for one scan at accept completion of instruction. (D2+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit

[Control data]

Device	Description	Setting data	Setting range	Set by
S1+0	Complete status	The status at the instruction completion is stored. 0 : No error (Normal completion) Except 0: Error code	—	System
S1+1	Number of reading data	Set the number of reading data.	1 to 20	User

[Controls]

(1) A part for the number of reading data of the control data specified with (S1+1) of data since the device specified with (S2) in the target CPU (n1) is stored to since the word device specified with (D1) of the self CPU in the Multiple CPU system.

(2) Figure specification of the bit device is possible for (S2) and (D1). However, figure specification is 4 figures and a start bit device number is only the multiple of 16. It becomes MULTI-COM.ERROR (Error code: 4353) when other values are specified.

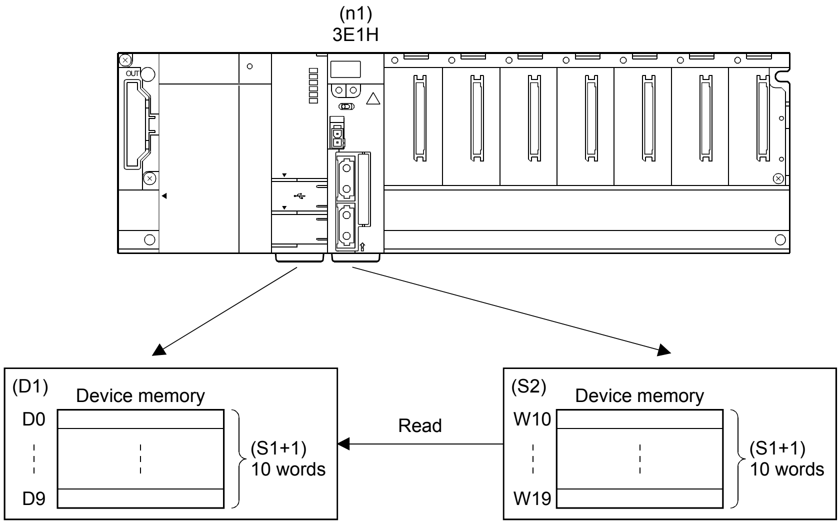
(3) Data can be read from device of the Motion CPU outside the range in the PLC CPU that executes this instruction like a motion register (#) etc., by setting (S2) by a character sequence " ".

(4) D(P).DDRD instruction accepting and normal/abnormal completion can be confirmed with the complete device (D2) or status display device (D2+1) at the completion.

(a) Complete device
It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.

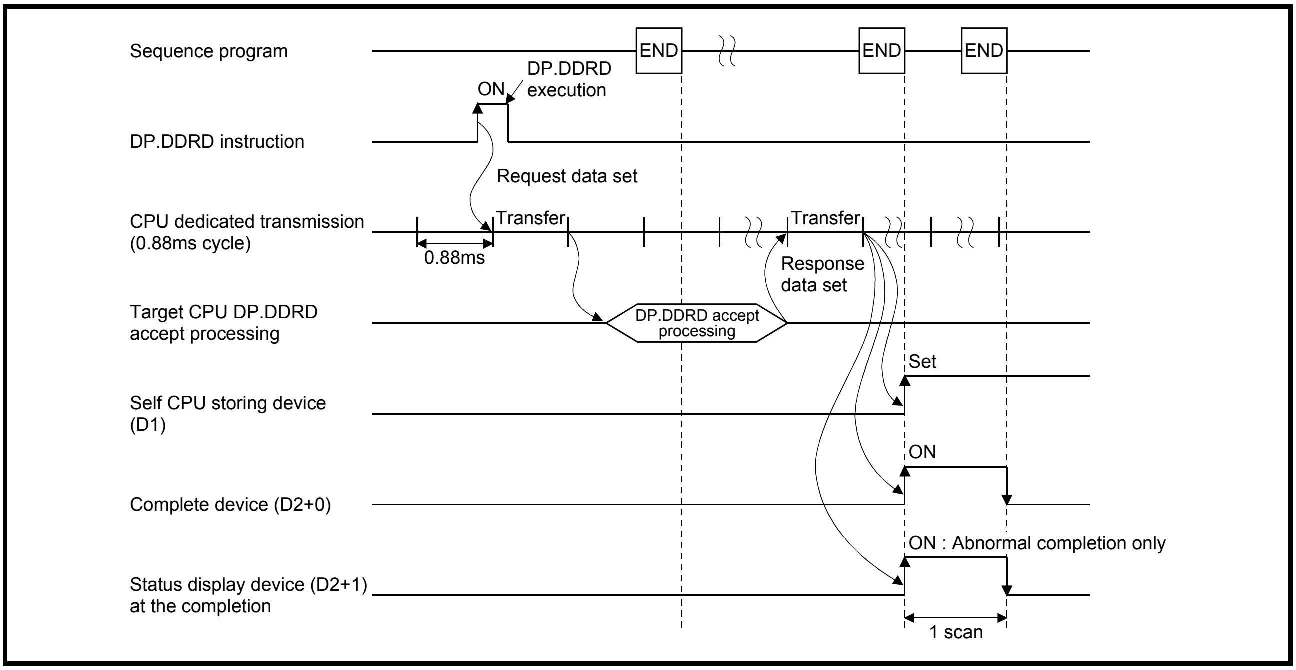
(b) Status display device at the completion
It is turned on/off according to the status of the instruction completion.
- Normal completion : OFF
- Abnormal completion : It is turned on by the END processing of scan which the instruction completed, and turned off by the next END processing.
(Error code is stored in control data (S1+0: Complete status).)

(5) There is a limitation for number of simultaneous instruction execution/ simultaneous acceptance in the Motion dedicated PLC instruction. (Refer to Section 3.3 (2).)
Exchange a large amount of data through the CPU shared memory.



[Operation]

Outline operation between CPUs at the DP.DDRD instruction execution is shown below.



[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (S0+0).

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2001	The specified device cannot be used in the Motion CPU, or it is outside the device range.	
2081	Number of reading data points set by D(P).DDRD instruction is wrong.	
2100	There are 65 or more simultaneous D(P).DDRD/D(P).DDWR instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note): 0000H (Normal)

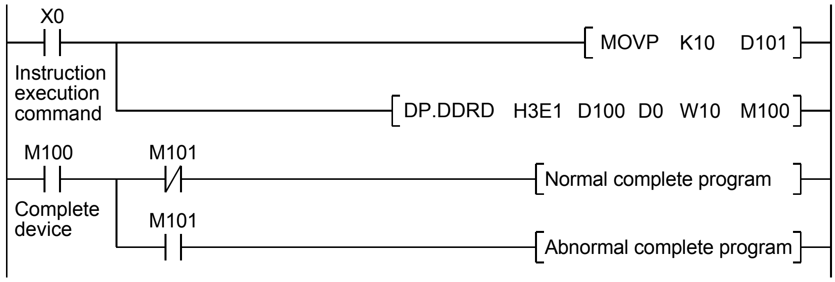
The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

Error code(Note)	Error factor	Corrective action
4101	Number of writing data exceeded the range of storage device.	
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	
4354	The character string that cannot be handled by the instruction specified is specified.	
4355	Number of writing data is outside the range of 1 to20.	

(Note): 0 (Normal)

[Program example]

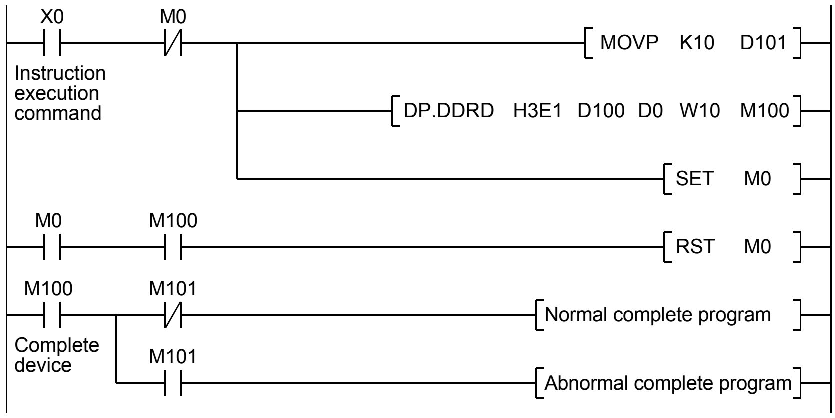
(1) Program which stores data for 10 words from D0 of the CPU No.2 to W10 or later of the self CPU, when X0 turned ON.



Stores the number of reading data "10" to the number of reading data points (S1+1) setting device D101.

Stores D0 to D9 of CPU No.2 to W10 to W19 of self CPU.

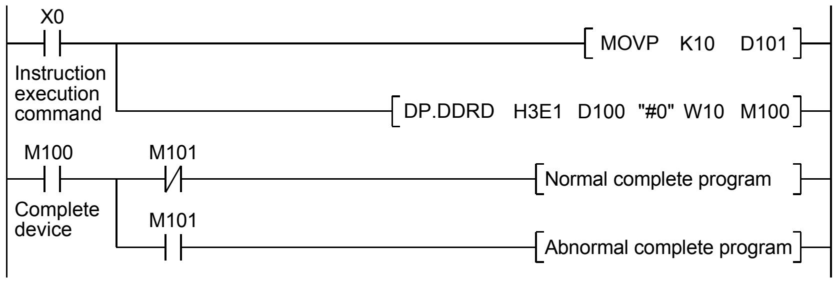
(2) Program which stores simultaneously data for 10 words from D0 of the CPU No.2 to W10 or later of the self CPU, while X0 turned ON.



Stores the number of reading data "10" to the number of reading data points (S1+1) setting device D101.

Stores D0 to D9 of CPU No.2 to W10 to W19 of self CPU.

(3) Program which stores data for 10 words from D0 of the CPU No.2 to #10 or later of the self CPU, when X0 turned ON.



Stores the number of reading data "10" to the number of reading data points (S1+1) setting device D101.

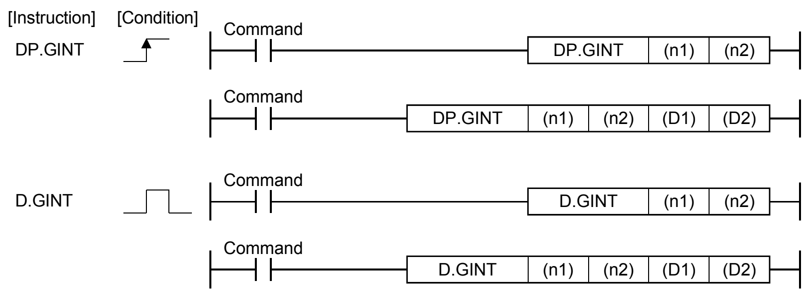
Stores D0 to D9 of CPU No. to #10 to #19 of self CPU.

3.2.11 Interrupt instruction to the other CPU: D(P).GINT (PLC instruction: D(P).GINT)

Setting data (Note-3)	Usable devices											
	Internal devices (System, User)		File register		Link direct device J□\G		Unit access device U□\G□		Index register Z□	Constant		Others
	Bit	Word	Bit	Word	Bit	Word	Bit	Word		Decimal K, Hexadecimal H	Real character string	
(n1)		○		○						○		
(n2)		○		○						○		
(D1) (Note-1)	△ (Note-2)		△ (Note-2)									
(D2) (Note-1)		△ (Note-2)		△ (Note-2)								

○: Usable △: Usable partly

(Note-1): Omission possible with both of (D1) and (D2) omission.
(Note-2): Local devices cannot be used.
(Note-3): Setting data (n1) to (D2) : Index qualification possible (except constant)



[Setting data]

Setting data	Description	Set by	Data type
(n1)	(First I/O No. of the target CPU)/16 Value to specify actually is the following. (Note-1) CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H Note): Motion CPU cannot be set as CPU No.1 in the Multiple CPU configuration.	User	16-bit binary
(n2)	Interrupt instruction No.	User	16-bit binary
(D1) (Note-1)	Complete devices (D1+0): Device which make turn on for one scan at accept completion of instruction. (D1+1): Device which make turn on for one scan at accept abnormal completion of instruction. ("D1+0" also turns on at the abnormal completion.)	System	Bit
(D2) (Note-1)	Complete status storage device	System	Word

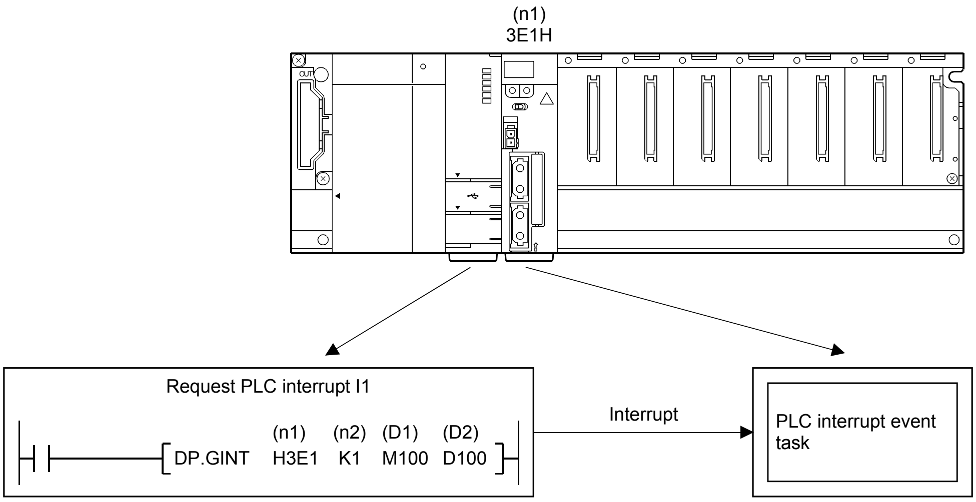
(Note-1): Omission possible with both of (D1) and (D2) omission.

[Controls]

(1) Processing for the active program (operation program status) of Motion SFC program set in the "PLC interruption of event task" is executed by the execution instruction of D(P).GINT instruction.

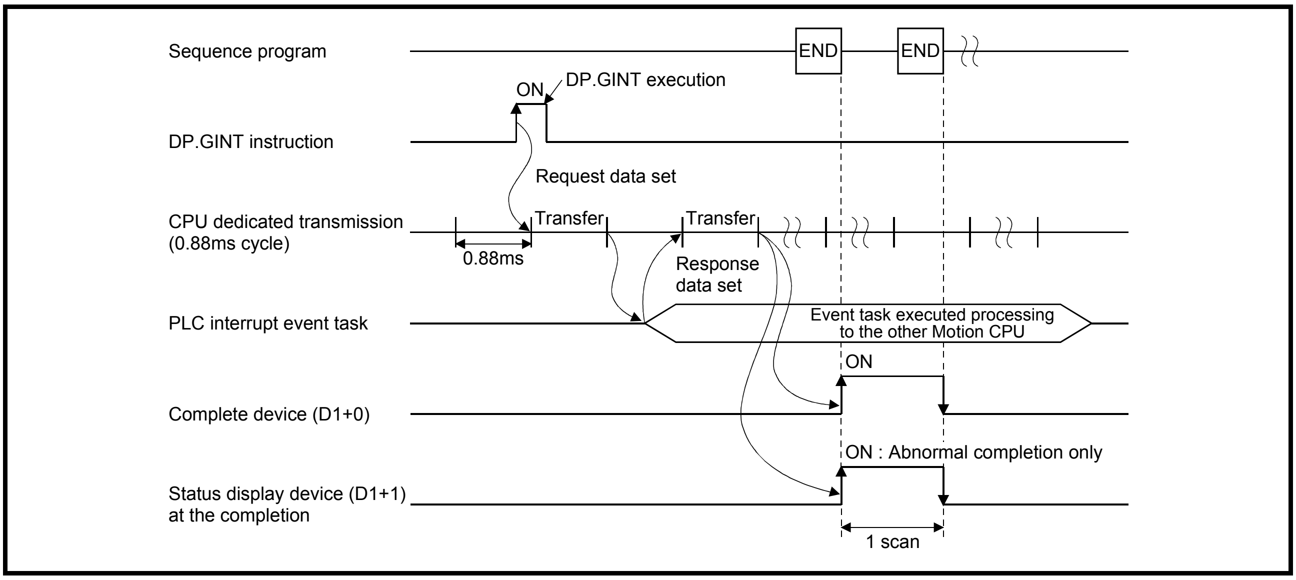
(2) This instruction is always valid regardless of the state of real mode/virtual mode/ mode switching when the operating system software of Motion CPU is SV22.

(3) Event processing is not executed when the Motion CPU side is DI (interrupt disable). Execute the EI (interrupt enable) instruction before event processing.



[Operation]

Outline operation between CPUs at the DP.GINT instruction execution is shown below.



[Errors]

The abnormal completion in the case shown below, and the error code is stored in the device specified with the complete status storage device (D2).
If the complete status storage device (D2) is omitted, an error is not detected and operation becomes "No operation".

Complete status (Note) (Error code)(H)	Error factor	Corrective action
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value.	Confirm a program, and correct it to a correct sequence program.
2082	The interrupt pointer No. set in the D(P).GINT instruction is outside the range of 0 to 15.	
2100	There are 33 or more simultaneous D(P).GINT instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.	

(Note): 0000H (Normal)

The diagnostic error flag (SM0) is turned on an operation error in the case shown below, and an error code is stored in the diagnostic error register (SD0).

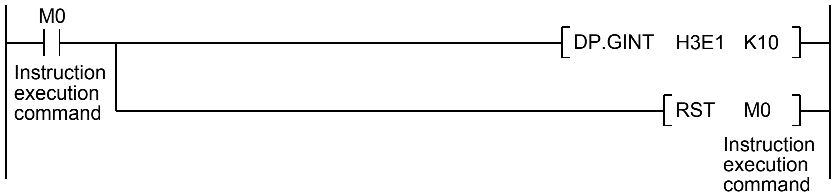
Error code (Note)	Error factor	Corrective action
4350	The target CPU module specified is wrong. (1) The reserved CPU is specified. (2) The uninstalled CPU is specified. (3) The first I/O number of the target CPU/16 (n1) is outside the range of 3E0H to 3E3H.	Confirm a program, and correct it to a correct sequence program.
4351	It cannot be executed to the specified target CPU module. (1) The instruction name is wrong. (2) The instruction unsupported by the target CPU module is specified.	
4352	The number of devices for instruction specified is wrong.	
4353	The device that cannot be used by the instruction specified is specified.	

(Note): 0 (Normal)

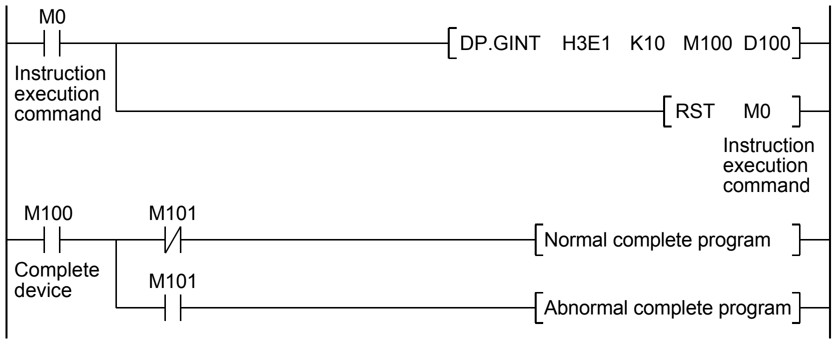
[Program example]

(1) Program which generates interrupt of interrupt pointer number 10 toward the Motion CPU (CPU No.2), when M0 turned ON.

<Example 1> Program which omits the complete device and complete status.



<Example 2> Program which uses the complete device and complete status.



3.3 Precautions

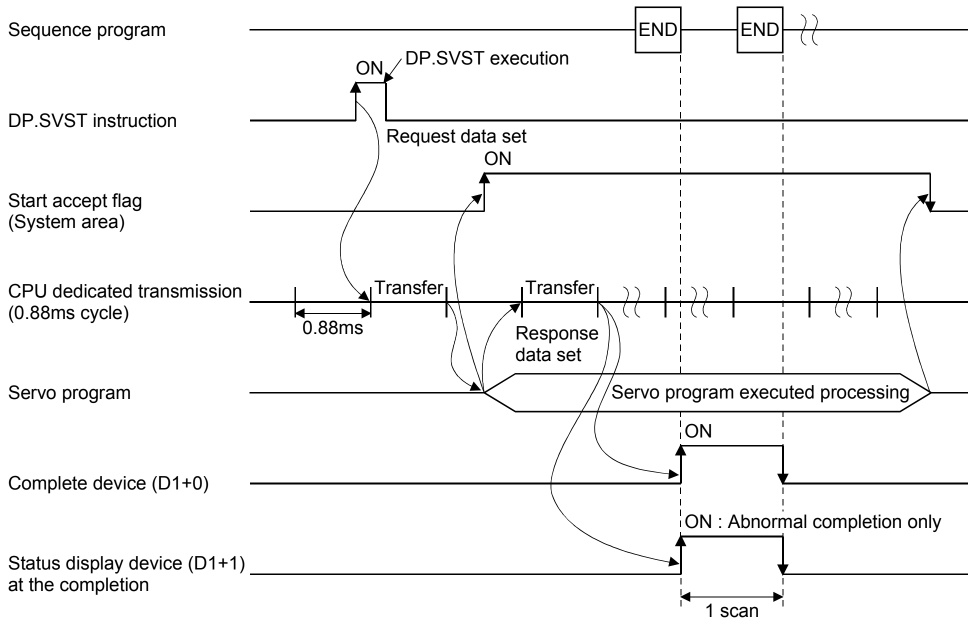
(1) CPU shared memory address used in Motion dedicated instruction

(a) Start accept flag (System area)

The status of each flag is stored in the following address.

<table>
<tr><th>CPU shared memory address
( ) is decimal address</th><th>Description</th></tr>
<tr><td>204H(516)
205H(517)</td><td>The start accept flag for 32 axes are stored corresponding to each bit.
Bits are actually set as the following:
• Q173DSCPU/Q173DCPU(-S1) : J1 to J32
• Q172DSCPU : J1 to J16
• Q172DCPU(-S1) : J1 to J8
OFF: Start accept enable
ON : Start accept disable
<table><tr><td></td><td>b15</td><td>b14 … b2</td><td>b1</td><td>b0</td></tr><tr><td>204H(516) address</td><td>J16</td><td>• • • • • • • • •</td><td>J2</td><td>J1</td></tr><tr><td>205H(517) address</td><td>J32</td><td>• • • • • • • • •</td><td>J18</td><td>J17</td></tr></table></td></tr>
<tr><td>20EH(526)
20FH(527)
QDS</td><td>The command generation axis start accept flag for 32 axes are stored corresponding to each bit.
Bits are actually set as the following:
• Q173DSCPU: J1 to J32
• Q172DSCPU: J1 to J16
OFF: Start accept enable
ON : Start accept disable
<table><tr><td></td><td>b15</td><td>b14 … b2</td><td>b1</td><td>b0</td></tr><tr><td>20EH(526) address</td><td>J16</td><td>• • • • • • • • •</td><td>J2</td><td>J1</td></tr><tr><td>20FH(527) address</td><td>J32</td><td>• • • • • • • • •</td><td>J18</td><td>J17</td></tr></table></td></tr>
</table>

The start accept flag is set after instruction acceptance of by the Motion CPU as follows.

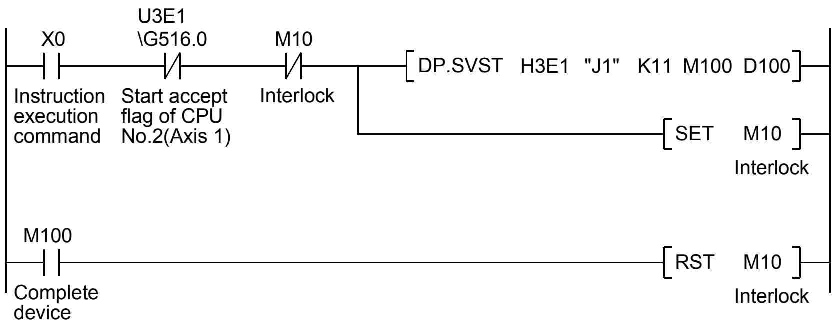


The start accept flag does not turn ON until the instruction accepting of instruction is completed by the Motion CPU after instruction execution by the PLC CPU.

Therefore, use a user device created interlock as required to prevent the execution of the next Motion dedicated PLC instruction and avoid a same axis double start error.

[Program example]

Program which executes continuous start of servo program No.11 for Axis 1 of the Motion CPU (CPU No.2), while X0 is ON.



(b) "Fixed at 0" area

The following area, which is used in Q173HCPU/Q172HCPU/ Q173CPU(N)/Q172CPU(N) is not used in Q173D(S)CPU/Q172D(S)CPU and is therefore "Fixed at 0" for these processor.

The following interlocks are not used in new Q173D(S)CPU/Q172D(S)CPU sequence program.

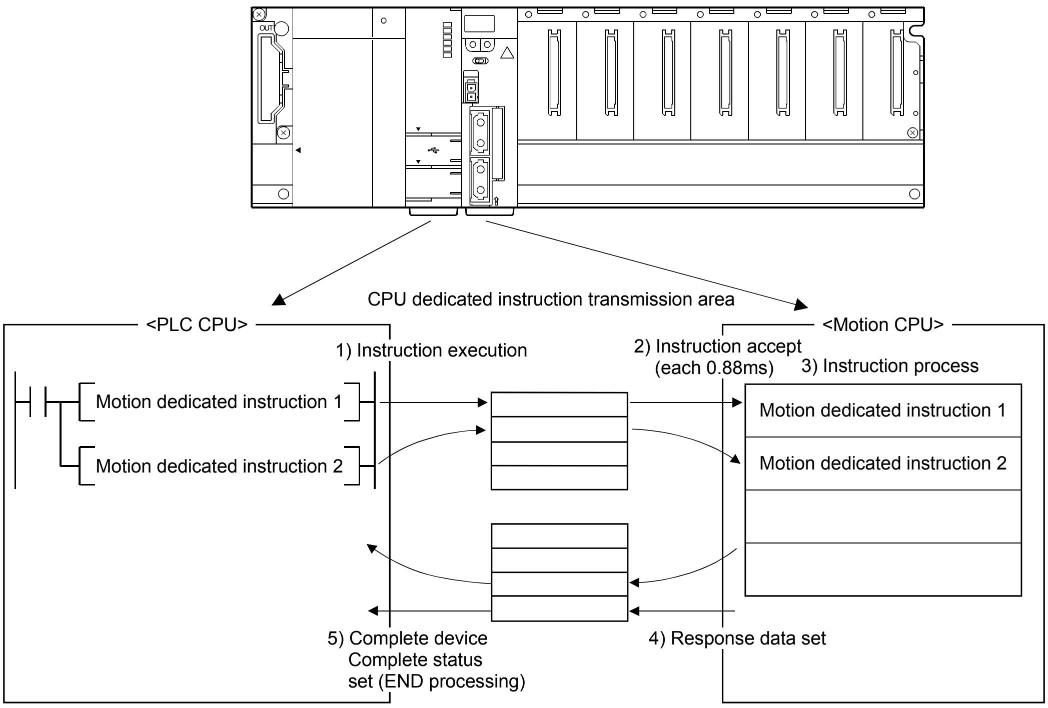
CPU shared memory address (Decimal address)	Description (Q173HCPU/Q172HCPU/Q173CPU(N)/Q172CPU(N))
30H(48)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU1)
31H(49)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU2)
32H(50)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU3)
33H(51)	Fixed at 0 (To self CPU high speed interrupt accept flag from CPU4)
206H(518)	Fixed at 0 (Speed changing flag (Axis1 to 16))
207H(519)	Fixed at 0 (Speed changing flag (Axis17 to 32))
208H(520)	Fixed at 0 (Synchronous encoder current value changing flag (Axis1 to 12))
20CH(524)	Fixed at 0 (Current value within 1cam shaft revolution changing flag (Axis1 to 16))
20DH(525)	Fixed at 0 (Current value within 1cam shaft revolution changing flag (Axis17 to 32))

(2) CPU dedicated instruction transmission

(a) Outline operation of Motion Dedicated PLC Instruction

Motion dedicated PLC instruction is transmitted through the CPU dedicated instruction transmission area set up in the system area on shared memory at the Multiple CPU high speed transmission.

Outline operation for Motion dedicated PLC instruction is shown below.



CPU dedicated instruction transmission area shown in table below is allocated as initial setting.

Table 3.1 Number of CPU dedicated instruction transmission area

Number of Multiple CPU modules	Number of CPU dedicated instruction transmission area for each target CPU
2	47 blocks
3	23 blocks
4	15 blocks

As shown in Table 3.2, each Motion dedicated PLC instruction uses a certain number of blocks in the CPU dedicated instruction transmission area until the "complete device" turns on by the PLC CPU after instruction execution.

Table 3.2 Number of blocks used for Motion dedicated PLC instruction

Instructions	Number of blocks used
D(P).SFCS	1
D(P).SVST	1
D(P).CHGA	1
D(P).CHGAS QDS Ver.!	1
D(P).CHGV	1
D(P).CHGVS QDS Ver.!	1
D(P).CHGT	1
D(P).CHGT2 QDS	1
D(P).DDWR	2 (Note)
D(P).DDRD	2 (Note)
D(P).GINT	1

(Note): When the number of transmitted data is 4 words or less, number of blocks used is 1.

[Operation example]
Below is an example when 12 D(P).SVST instructions and 12 D(P).DDWR instructions (5 word or more each) are executed simultaneously.

The number of blocks used is as follows;
12 D(P).SVST instructions × 1 block each +
12 D(P).DDWR instructions × 2 blocks each
= 36 (Total blocks used)

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(b) Permissible number of executions for dedicated instructions on Multiple CPU high-speed transmission

When the number of blocks being used to communicate with each CPU in the Multiple CPU dedicated instruction transmission area exceeds the set value for maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (special registers SD796 to SD799 of PLC CPU), the system enters a state where the Motion dedicated PLC instruction is not accepted (permissible number of executions exceeded state). At the time of Motion dedicated instruction execution towards the target CPU, an abnormal complete status "0010H" is set in the complete status device. If the complete device is omitted, no operation occurs at all.

An interlock can be created using block information using Multiple CPU high-speed transmission dedicated instruction (SM796 to SM799 of the PLC CPU) so that the permissible number of executions is not exceeded.

• Special relay of PLC CPU

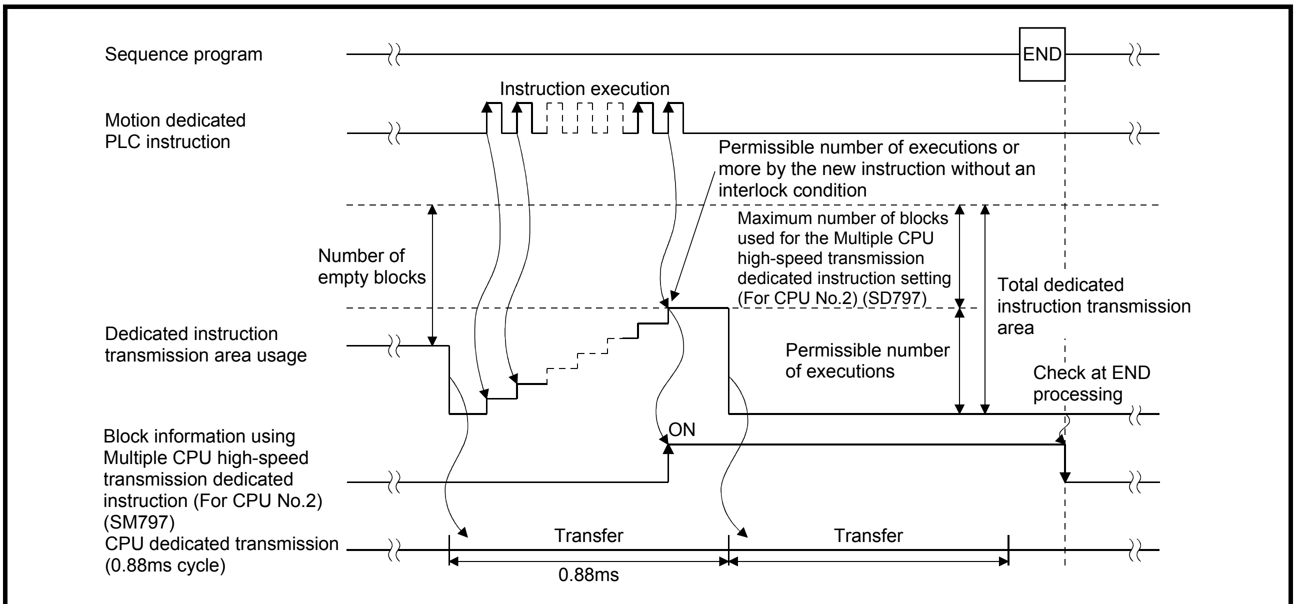
<table>
<tr><th>Device No.</th><th>Name</th><th>Meaning</th><th>Explanation</th><th>Set by</th></tr>
<tr><td>SM796</td><td>Block information using Multiple CPU high-speed transmission dedicated instruction
(For CPU No.1)</td><td>OFF: Block is secured
ON : Block set by SD796 cannot be secured</td><td rowspan="4">Turns ON when the number of the remaining blocks of the dedicated instruction transmission area used for the Multiple CPU high-speed transmission dedicated instruction is less than the number of blocks specified by "SD796 to SD799".
Turns ON at instruction execution. Turns OFF when empty area exists at END processing.</td><td rowspan="4">System
(When instruction/ END processing executed)</td></tr>
<tr><td>SM797</td><td>Block information using Multiple CPU high-speed transmission dedicated instruction
(For CPU No.2)</td><td>OFF: Block is secured
ON : Block set by SD797 cannot be secured</td></tr>
<tr><td>SM798</td><td>Block information using Multiple CPU high-speed transmission dedicated instruction
(For CPU No.3)</td><td>OFF: Block is secured
ON : Block set by SD798 cannot be secured</td></tr>
<tr><td>SM799</td><td>Block information using Multiple CPU high-speed transmission dedicated instruction
(For CPU No.4)</td><td>OFF: Block is secured
ON : Block set by SD799 cannot be secured</td></tr>
</table>

• Special register of PLC CPU

Device No.	Name	Meaning	Explanation	Set by
SD796	Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.1)	Maximum number of blocks range for dedicated instructions Range: 1 to 7 (Default: 2) (Note): When setting other than 1 to 7, the register operates as 7.	Specifies the maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction. When the dedicated instruction of Multiple CPU transmission is executed to the target CPU, and the number of empty blocks of the dedicated instruction transmission area is less than the setting value of this register, "SM796 to SM799" is turned ON, which is used as the interlock signal for consecutive execution of the dedicated instruction of Multiple CPU transmission.	User (At 1 scan after RUN)
SD797	Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.2)			
SD798	Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.3)			
SD799	Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.4)			

[Operation timing]

Operation which executes each Motion dedicated instruction and turns on the Multiple CPU high-speed transmission block information.



[Operation example]
When multiple D(P).DDWR instructions (5 word or more each) are executed simultaneously before turning on each complete device in the 2 Multiple CPUs.
If the number of blocks used for each item is set as follows,

- Number of CPU dedicated instruction transmission area: 47 blocks (Initial value)
- Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.2) (SD797): 2 (Initial value)
- D(P).DDWR number of blocks used: 2

And, when 23 D(P).DDWR instructions are issued within the Multiple CPU high speed transmission cycle (0.88 ms), the number of blocks used is as follows.

- 2 (D(P).DDWR number of blocks) × 23 (D(P).DDWR instructions) = 46 (Total blocks used)

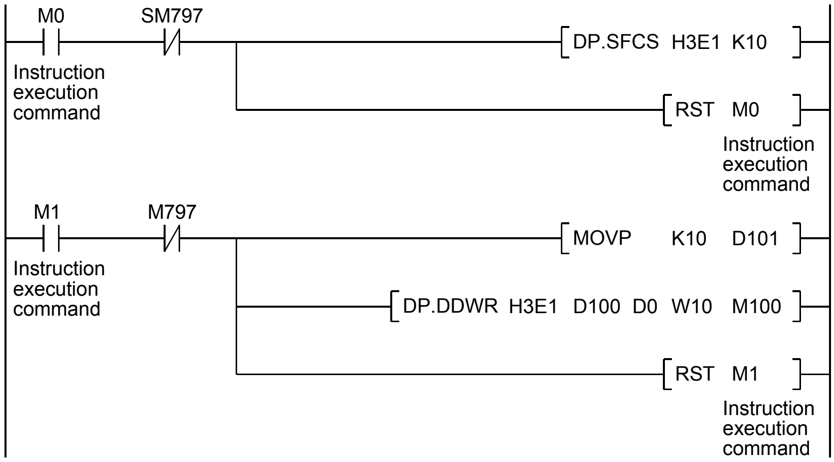
Therefore, the number of empty blocks is as follows;

- 47 (Number of CPU dedicated instruction transmission area) - 46 (Total blocks used) = 1 (Number of empty blocks)
- 1 (Number of empty blocks) < 2 (Maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.2) (SD797))

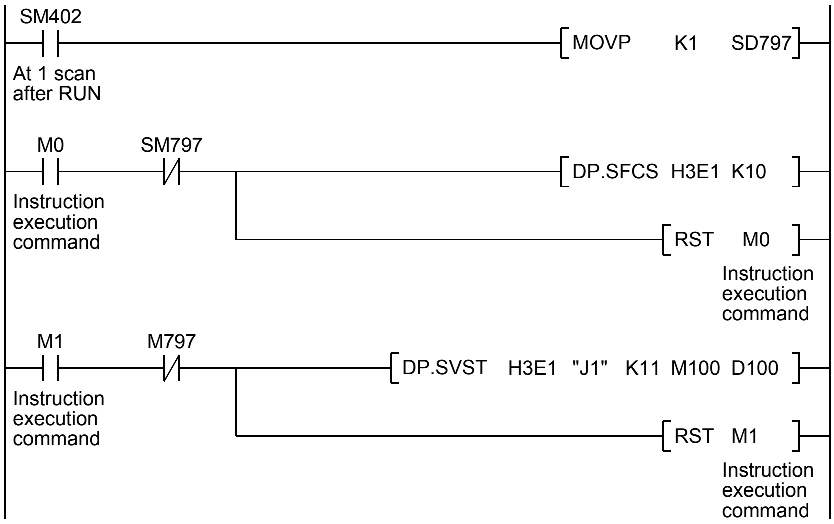
In the above case, the number of empty blocks is less than the maximum number of blocks used for the Multiple CPU high-speed transmission dedicated instruction setting (For CPU No.2) (SD797), therefore block information using Multiple CPU high-speed transmission dedicated instruction (For CPU No.2) (SM797) turns on.
If a new instruction is executed while in this status, it will be more than the permissible number of executions. However, this can be avoided by using SM797 as an interlock.

[Program example]

(1) Program which sets 2 (Initial value) to SD797 and uses SM797 as an interlock when DP.DDWR (Number of blocks used : 2) is executed.



(2) Program which sets 1 to SD797 and uses SM797 as an interlock condition when D(P).DDWR/D(P).DDRD is not executed.



(c) CPU dedicated instruction transmission area

If the size of the CPU dedicated instruction transmission area is insufficient, it can be increased changing the system area size. The size of the CPU dedicated instruction transmission area is decided depending on the number of CPU modules used and selected system area size as follows.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for details of the system area size change.

• Number of Multiple CPU modules: 2

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1k word	47 blocks
2k word	111 blocks

• Number of Multiple CPU modules: 3

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1k word	23 blocks
2k word	55 blocks

• Number of Multiple CPU modules: 4

Selected system area size	Number of CPU dedicated instruction transmission area for each target CPU
1k word	15 blocks
2k word	36 blocks

(d) Number of simultaneous instruction acceptance for Motion CPU

The following number of instructions can be accepted simultaneously in the Motion CPU.

- D(P).SFCS : 64
- Total of D(P).SVST, D(P).CHGA and D(P).CHGAS:
    - Q173DSCPU/Q173DSCPU : 128 (Note-1)
    - Q173DCPU(-S1)/Q173DCPU(-S1) : 64

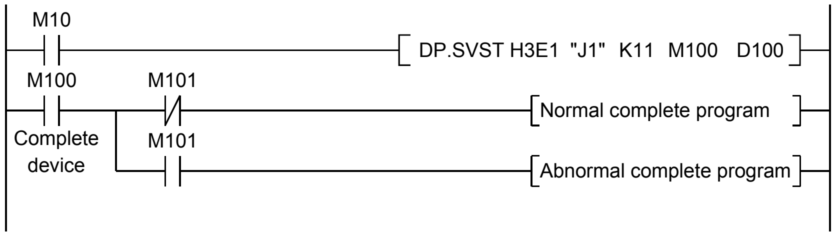
    (Note-1): 64 for operating system software version "00A".
- D(P).GINT : 32
- Total of D(P).DDRD and D(P).DDWR : 64
- D(P).CHGV/D(P).CHGVS /D(P).CHGT/D(P).CHGT2

    : Last instruction for each axis executed is valid. There is not a limitation for number of simultaneous instruction acceptance.

When more than the above number of instructions are executed by the PLC CPU, even if there is enough area in the CPU dedicated instruction transmission area, the Motion CPU cannot accept it.
In this case, 2100 is set to the complete status information and it abnormal completion occurs.

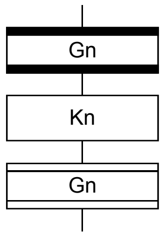
(3) Execution of Motion dedicated PLC instruction

(a) Motion dedicated PLC instruction can be executed with a fixed cycle execute type program and interrupt program. However, the complete device is a pulse-type. If the complete device (M100 in below example) is set, it may not be recognized during the PLC scan. Therefore, the sequence program should scan for completion of the device and use a set bit to execute the Motion instruction string.



(b) The below devices cannot be used as program file registers or local devices.
- Each instruction’s complete device and complete status
- D1 of D(P).DDRD instruction (First device of the self CPU where the reading data is stored.)

(c) When using the Motion dedicated function of the operation control step (Fn/FSn) and servo program (Kn) in Motion CPU, it is necessary to create a user-defined interlock using WAIT transition (Gn) as shown below.



(4) Complete status information

The codes stored in complete status at the completion of Motion dedicated PLC instruction are shown below.

If the complete status storage device is omitted, an error is not detected and operation becomes "No operation".

Complete status (Error code) (H)	Error factor
0	Normal completion
0010	Instruction request to Motion CPU from PLC CPU exceeds the permissible value. (Permissible value is different depending on the number of CPU modules.).
2000 (Note-1)	Command that cannot be decoded in the Motion CPU was specified.
2001 (Note-1)	The specified device cannot be used in the Motion CPU, or it is outside the device range.
2002 (Note-1)	A Motion dedicated PLC instruction that does not correspond with the operating system of the Motion CPU was executed.
2080 (Note-1)	Number of writing data points set by D(P).DDWR instruction is wrong.
2081 (Note-1)	Number of reading data points set by D(P).DDRD instruction is wrong.
2082 (Note-1)	The interrupt pointer No. set in the D(P).GINT instruction is outside the range of 0 to 15.
2100 (Note-1)	• D(P).SFCS instruction use There are 65 or more simultaneous D(P).SFCS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • D(P).SVST/D(P).CHGA/D(P).CHGAS instruction use There are the following number or more simultaneous D(P).SVST/D(P).CHGA/ D(P).CHGAS instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • Q173DSCPU/Q172DSCPU :129 or more (Note-2) • Q173DCPU(-S1)/Q173DCPU(-S1) : 65 or more • D(P).GINT instruction use There are 33 or more simultaneous D(P).GINT instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them. • D(P).DDRD/D(P).DDWR instruction use There are 65 or more simultaneous D(P).DDRD/D(P).DDWR instruction requests to the Motion CPU from the PLC CPU, therefore the Motion CPU cannot process them.
2200 (Note-1)	The starting Motion SFC program No. is outside the range of 0 to 255.
2201 (Note-1)	The servo program No. to execute is outside the range of 0 to 4095.
2202 (Note-1)	Axis No. set by D(P).SVST instruction is wrong.
2203 (Note-1)	Axis No. set by D(P).CHGA instruction is wrong.
2204 (Note-1)	Axis No. set by D(P).CHGV instruction is wrong.
2205 (Note-1)	Axis No. set by D(P).CHGT instruction is wrong.
2206 (Note-1)	Axis No. set by D(P).CHGT2 instruction is wrong.
2207 (Note-1)	Axis No. set by D(P).CHGAS instruction is wrong.
2208 (Note-1)	Axis No. set by D(P).CHGVS instruction is wrong.

(Note-1): The error code is dedicated with the Motion CPU.

(Note-2): 65 or more for operating system software version "00A".

(5) Order of instruction execution

Methods to control using execution data after it is transmitted from the PLC CPU to the Motion CPU are shown below.

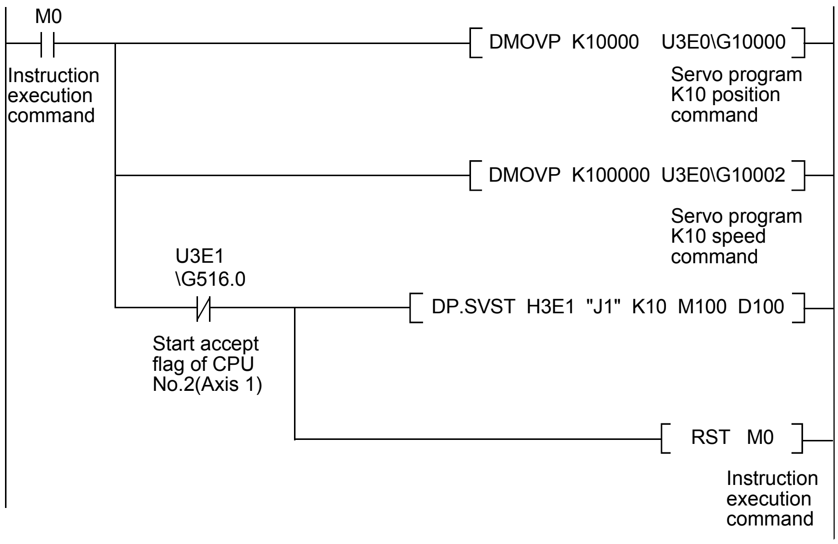
(a) Method to execute after data is written to the shared memory area (Multiple CPU high speed transmission area).

Write the data from PLC CPU to the shared memory area (Multiple CPU high speed transmission area) of the self CPU, and then it can be utilized for Motion dedicated PLC instruction execution.

[Program example]

Program which starts the servo program (positioning) by DP.SVST instruction after the data has been writing to shared memory area (Multiple CPU high speed transmission area (U3E0\G10000 to U3E0\G10003) from PLC CPU (CPU No.1).

Sequence program (PLC CPU side)



Servo program (Motion CPU side)

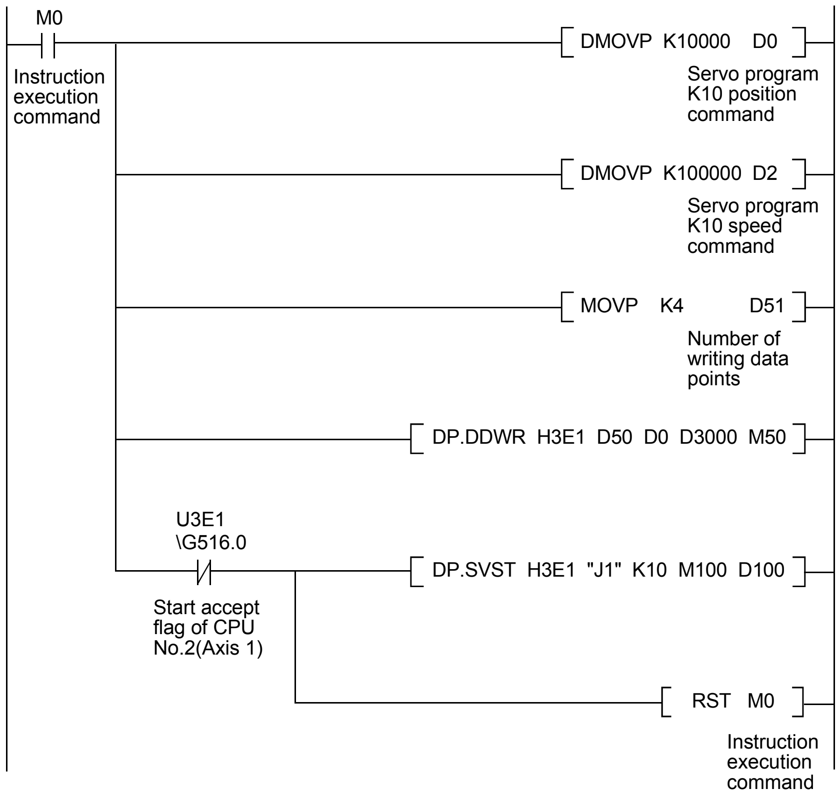
```
[ K 10: Real ]
  1  INC-1
     Axis  1,  U3E0\G10000  μm
     Speed    U3E0\G10002  mm/min
```

(b) Method to execute after data is written by D(P).DDWR instruction
Write the data from the PLC CPU to the Motion CPU by D(P).DDWR instruction, and then it can be utilized for Motion dedicated PLC instruction execution.

[Program example]

Program which starts the servo program (positioning) by DP.SVST instruction after data is written to D3000 to D3002 of the Motion CPU (CPU No.2) from the PLC CPU (CPU No.1) by DP.DDWR.

Sequence program (PLC CPU side)



Servo program (Motion CPU side)

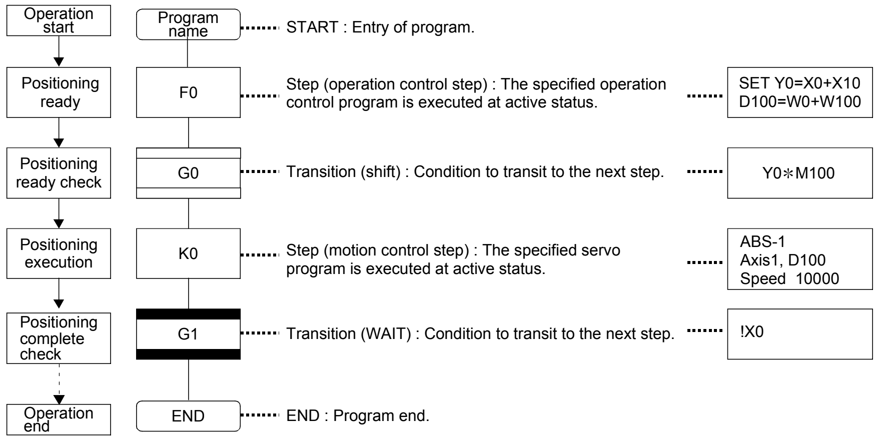
```
[ K 10: Real ]
1  INC-1
   Axis      1,   D3000 μm
   Speed          D3002 mm/min
```

MEMO

4. MOTION SFC PROGRAMS

4.1 Motion SFC Program Configuration

The Motion SFC Program is constituted by the combination of start, steps, transitions, end and others are shows below.



The above Motion SFC program to be started performs the following operations.

(1) The step (F0) is activated and the operation specified with the step (F0) is executed (positioning ready). A step in such an active state is called an active step.

(2) Whether the condition specified with the transition (G0) has enabled or not (whether the positioning program can be started or not) is checked. The active step (F0) is deactivated at the completion of condition and the next step (K0) is activated (servo program (K0) is started).

(3) The operating completion of the step (K0) (positioning completion of the servo program K0) is checked, and control transits to the next step at operating completion (completion of condition).

(4) With the transition of the active step as described in above (1) to (3), control is executed and ends at END.

Refer to Section "9.2.2 Task operation" for details of the execution timing of the Motion SFC program such as above.

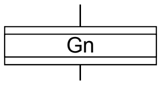
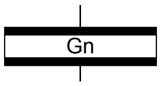
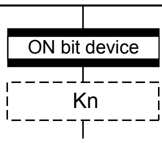
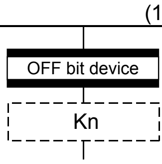
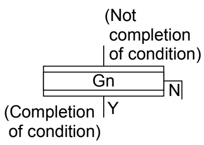
POINT

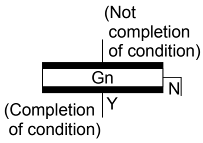
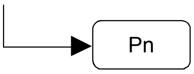
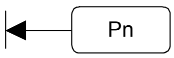
The number of steps which can be active steps simultaneously is up to 256, with those of all Motion SFC programs combined. Excess of 256 will result in the Motion SFC error (error code: 16120).
Each symbol of the Motion SFC program is as follows.
F/FS : Operation control, K : Positioning control, G : Judgment

4.2 Motion SFC Chart Symbol List

Parts as Motion SFC program components are shown below. The operation sequence or transition control is expressed with connecting these parts by directed lines in the Motion SFC program.

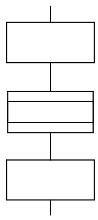
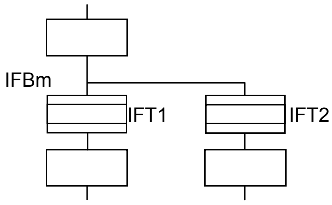
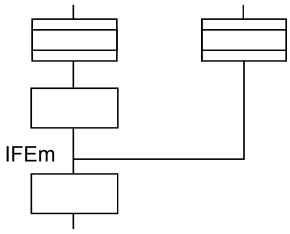
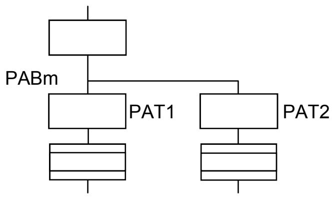
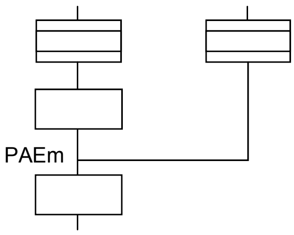
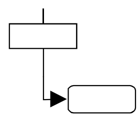
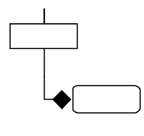
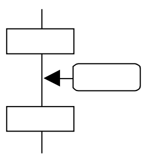
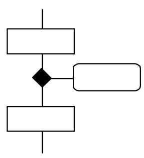
Classification	Name	Symbol (Code size (byte))	List Representation	Function
Program start/end	START	Program name (0)	Program name	• Indicates an entry of program as a program name. • Specify this program name at a subroutine call. • Only one program name for one program.
	END	END (8)	END	• Indicates an end (exit) of program. • When a subroutine call was carried out, returns to the call source program. • Multiple program names or no symbols for one program.
Step	Motion control step	Kn (8)	CALL Kn	• Starts a servo program Kn (K0 to K4095).
	Once execution type operation control step	Fn (8)	CALL Fn	• Execute once the operation control program Fn (F0 to F4095).
	Scan execution type operation control step	FSn (8)	CALL FSn	• Repeats an operation control program FSn (FS0 to FS4095) until the next transition condition enables.
	Subroutine call/start step	Program name (8)	GSUB program name	• When the next of GSUB is WAIT, performs "subroutine call" and transits control to the specified program. Control returns to the call source program at END execution. • When the next of GSUB is except WAIT, performs "subroutine start", and starts the specified program and transits to the next (lower part). The start source and destination programs are executed simultaneously, and the call destination program ends at END execution.
	Clear step	CLR Program name (8)	CLR program name	• Stops and ends the specified program running. After an end, it is started from the initial (start step) by restarting the program. • When the specified program is during "subroutine call", the subroutine program is also stopped to execute. • When the specified program is after "subroutine start", the subroutine program is not stopped to execute. • When clearing to the subroutine by which the "subroutine call" was executed, the specified subroutine is stopped to execute, returns to the call source program, and transits to the next.

Classification	Name	Symbol (Code size (byte))	List representation	Function
Transition	Shift (Pre-read transition)	 (8)	SFT Gn	• When just before is the motion control step, transits to the next step by formation of transition condition Gn (G0 to G4095) without waiting for the motion operating completion. • When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. • When just before is subroutine call or starting step, transits to the next step by formation of transition condition without waiting for the operating completion of subroutine.
	WAIT	 (8)	WAIT Gn	• When just before is the motion control step, waits for the motion operating completion and then transits to the next step by the completion of transition condition Gn (G0 to G4095). • When just before is the operation control step, transits to the next step by formation of transition condition after operating execution. (Same operation as Shift.) • When just before is subroutine call or starting step, waits for the operating completion of subroutine and then transits to the next step by the completion of transition condition.
	WAITON	 (14)	WAITON bit device	• Prepares for starting of the next motion control step, and issues an instruction immediately when the specified bit device turns ON. • Always pair this transition with the motion control step one-for-one.
	WAITOFF	 (14)	WAITOFF bit device	• Prepares for starting of the next motion control step, and issues an instruction immediately when the specified bit device turns OFF. • Always pair this transition with the motion control step one-for-one.
	Shift Y/N		IFBm IFT1 SFT Gn : JMP IFEm IFT2 SFT Gn+? : JMP IFEm IFEm	• When just before is the motion control step, transits to the next step by formation of transition condition Gn (G0 to G4095) without waiting for the motion operating completion. If not formation of transition condition, transits to the right-connected step. • When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. If not the completion of transition condition, transits to the right-connected step. • When just before is "subroutine call" or "starting step", transits to the next step by the completion of transition condition without waiting for the operating of subroutine completion. If not formation of transition condition, transits to the right-connected step.

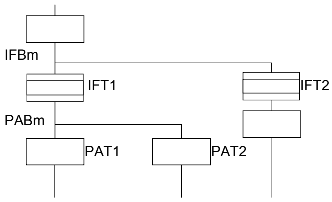
Classification	Name	Symbol (Code size (byte))	List representation	Function
Transition	WAIT Y/N		IFBm IFT1 WAIT Gn : JMP IFEm IFT2 WAIT Gn+? : JMP IFEm IFEm	• When just before is the motion control step, waits for the motion operating completion and then transits to the next step by formation of transition condition Gn (G0 to G4095). If not completion of transition condition, transits to the right-connected step. • When just before is the operation control step, transits to the next step by the completion of transition condition after operating execution. If not the completion of transition condition, transits to the right-connected step. (Same operation as Shift.) • When just before is subroutine call or starting step, waits for the operating completion of subroutine, and then transits to the next step by the completion of transition condition. If not formation of transition condition, transits to the right-connected step.
Jump	Jump	 (14)	JMP Pn	• Jumps to the specified pointer Pn (P0 to P16383) of the self program.
Pointer	Pointer	 (8)	Pn	• Indicates a jump destination pointer (label). • This pointer can be set at a step, transition, branch point or coupling point. • P0 to P16383 can be set in one program. The same No. may also be used in other programs.

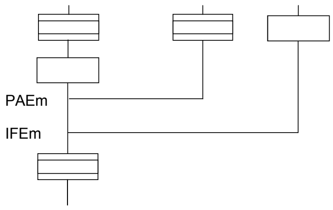
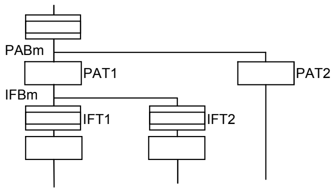
4.3 Branch and Coupling Chart List

Branch and coupling patterns which specify step and transition sequences in the Motion SFC charts are shown below.

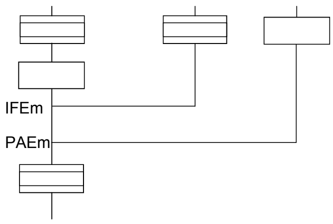
	Name (Code size (byte))	Motion SFC chart symbol	List representation	Function
Basic type	Series transition (Corresponding symbol size)		List representation corresponding to the Motion SFC chart symbols shown in Section 4.2.	• Steps and transitions connected in series are processed in order from top to bottom. • Steps and transitions need not be lined up alternately. • When a transition is omitted, unconditional shift processing is performed.
	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''	• The route which transition condition enables first is executed after executing the step or transition preceding a branch. • Selective branch destinations should always be started by transitions, all of which must be Shift or WAIT. (Using Shift and WAIT together will cause a parallel branch.)
	Selective coupling (8)			• After the route branched by a selective branch has been processed, execution shifts to a coupling point. • A coupling may be preceded and followed by either a step or a transition.
	Parallel branch (Number of branches × 22 + number of coupling points × 2 + 12)		CALL Kn PABm PAT1 CALL Fn SFT Gn' : JMP PAEm PAT2 CALL Fn' SFT Gn'' : (JMP PAEm) PAEm CALL Fn'' :	• Multiple routes (steps) connected in parallel are executed simultaneously. • Each parallel branch destination may be started by either a step or transition.
	Parallel coupling (8)			• Execution waits at the coupling point for executions of the routes branched by a parallel branch to be completed, and shifts to the next when executions of all routes are completed. • A coupling may be preceded and followed by either a step or a transition. • When this coupling is preceded by an FS step, scans are executed during waiting. After waiting is complete, scans are not executed.
	Jump transition (Corresponding symbol size)	<Normal jump> <Coupling jump>  	CALL Fn JMP Pn	1) Normal jump • After the step or transition preceding this jump transition is executed, execution shifts to the pointer Pn specified within its own program. • The jump destination may either be a step or transition. • When a jump takes place from an FS step to a transition, scans are executed during waiting for the completion of transition condition of the jump destination. 2) Coupling jump • When a jump to the other route within a parallel branch takes place after the parallel branch, a "coupling jump" takes place and execution waits at the jump destination.
		 	CALL Fn' Pn CALL Kn	

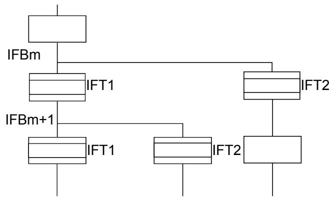
Combining the basic type branches/couplings provides the following application types, which are defined as in the basic types.

<table>
<tr><th></th><th>Name</th><th>Motion SFC chart symbol</th><th>List representation</th><th>Function</th></tr>
<tr><td rowspan="4">Appli-
cation
type</td><td>Selective branch
\|
Parallel branch</td><td></td><td rowspan="2">CALL Kn
IFBm
IFT1
SFT Gn
PABm
PAT1
CALL Fn
:
JMP PAEm
PAT2
CALL Fn'
:
(JMP PAEm)
PAEm
JMP IFEm

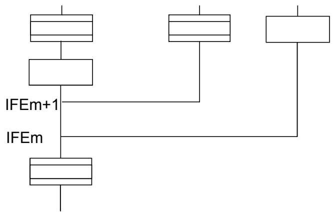
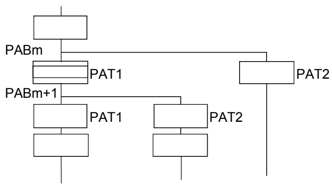
IFT2
SFT Gn'
CALL Fn''
:
(JMP IFEm)
IFEm
SFT Gn''</td><td>• After a selective branch, a parallel branch can be performed.</td></tr>
<tr><td>Parallel coupling
\|
Selective coupling</td><td></td><td>• The selective coupling point can be the same as the coupling point of a parallel coupling for selective branch → parallel branch. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a selective coupling, as shown on the left.
• In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the selective coupling point (IFEm).</td></tr>
<tr><td>Parallel branch
\|
Selective branch</td><td></td><td rowspan="2">SFT Gn
PABm
PAT1
CALL Fn
IFBm
IFT1
SFT Gn'
CALL Fn'
:
JMP IFEm
IFT2
SFT Gn''
CALL Fn''
:
(JMP IFEm)
IFEm
JMP PAEm

PAT2
CALL Fn'''
:
CALL Kn
(JMP PAEm)

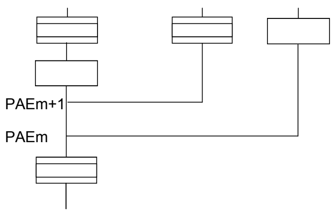
PAEm
SFT Gn'''</td><td>• After a parallel branch, a selective branch can be performed.</td></tr>
<tr><td>Selective coupling
\|
Parallel coupling</td><td></td><td>• The parallel coupling point can be the same as the coupling point of a selective coupling for parallel branch → selective branch.
Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a parallel coupling, as shown on the left.
• In this case, a pointer (Pn) cannot be set between the selective coupling point (IFEm) and the parallel coupling point (PAEm).</td></tr>
</table>

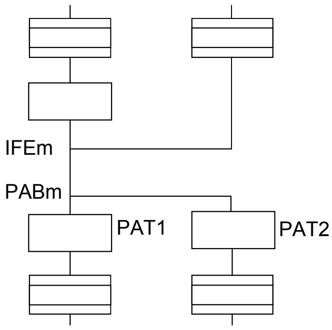
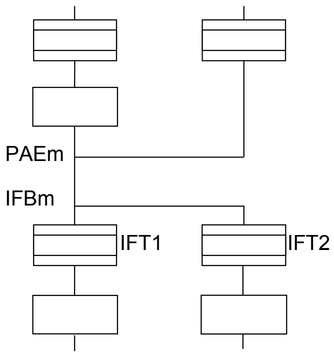
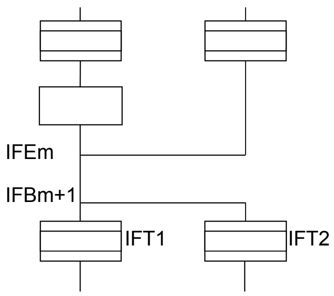
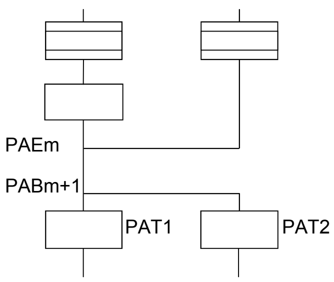
<table>
<tr><th></th><th>Name</th><th>Motion SFC chart symbol</th><th>List representation</th><th>Function</th></tr>
<tr><td rowspan="4">Appli-
cation
type</td><td>Selective branch
|
Selective branch</td><td></td><td rowspan="2">CALL Kn
IFBm
IFT1
SFT Gn
IFBm+1
IFT1
SFT Gn'
:
JMP IFEm+1
IFT2
SFT Gn''
:
(JMP IFEm+1)
IFEm+1
JMP IFEm

IFT2
SFT Gn'''
CALL Fn'
:
(JMP IFEm)

IFEm
SFT Gn''''
:</td><td>• After a selective branch, a selective branch can be performed.</td></tr>
<tr><td>Selective coupling
|
Selective coupling</td><td></td><td>• The two selective coupling points for selective branch → selective branch can be the same. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a selective coupling, as shown on the left.
• In this case, a pointer (Pn) cannot be set between the selective coupling point (IFEm+1) and the selective coupling point (IFEm).</td></tr>
<tr><td>Parallel branch
|
Parallel branch</td><td></td><td rowspan="2">CALL Kn
PABm
PAT1
SFT Gn
PABm+1
PAT1
CALL Fn'
:
JMP PAEm+1
PAT2
CALL Fn''
:
(JMP PAEm+1)
PAEm+1
JMP PAEm

PAT2
CALL Fn'''
:
CALL Kn
JMP PAEm

PAEm
SFT Gn'''
:</td><td>• After a parallel branch, a parallel branch can be performed.
• A parallel branch can be nested up to four levels.</td></tr>
<tr><td>Parallel coupling
|
Parallel coupling</td><td></td><td>• The two parallel coupling points for parallel branch parallel branch can be the same. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a parallel coupling, as shown on the left.
• In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm+1) and the parallel coupling point (PAEm).</td></tr>
</table>

	Name	Motion SFC chart symbol	List representation	Function
Appli-cation type	Selective coupling | Parallel branch		: (JMP IFEm) IFEm PABm PAT1 CALL Fn : JMP PAEm PAT2 CALL Fn' : (JMP PAEm) PAEm :	• The selective coupling point and parallel branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a parallel branch, as shown on the left. • In this case, a pointer (Pn) cannot be set between the selective coupling point (IFEm) and the parallel branch point (PABm).
	Parallel coupling | Selective branch		: JMP PAEm PAEm IFBm IFT1 SFT Gn : JMP IFEm IFT2 SFT Gn' : (JMP IFEm) IFEm :	• The parallel coupling point and selective branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a selective branch, as shown on the left. • Execution waits at the parallel coupling point and shifts to the selective branch. • In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the selective branch point (IFBm).
	Selective coupling | Selective branch		: (JMP IFEm) IFEm IFBm+1 IFT1 SFT Gn : JMP IFEm+1 IFT2 SFT Gn' : (JMP IFEm+1) IFEm+1	• The selective coupling point and selective branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a selective coupling → a selective branch, as shown on the left. • In this case, a pointer (Pn) cannot be set between the selective coupling point (IFEm) and the selective branch point (IFBm+1).
	Parallel coupling | Parallel branch		: (JMP PAEm) PAEm PABm+1 PAT1 CALL Fn : JMP PAEm+1 PAT2 CALL Fn' : (JMP PAEm+1) PAEm+1 :	• The parallel coupling point and parallel branch point can be the same. Note that in the Motion SFC chart, this type is displayed in order of a parallel coupling → a parallel branch, as shown on the left. • Execution waits at the parallel coupling point and shifts to the parallel branch. • In this case, a pointer (Pn) cannot be set between the parallel coupling point (PAEm) and the parallel branch point (PABm+1).

4.4 Motion SFC Program Name

Set the "Motion SFC program name" to the Motion SFC program No.0 to No.255 individually.
Set the Motion SFC program name within 16 characters. Specify this Motion SFC program name for a "subroutine call/start step (GSUB)" and "clear step (CLR)".
Refer to "Chapter 11 USER FILES" for the user file of the Motion SFC program.

POINT

(1) It is can be set the Motion SFC program to any of No.0 to No.255. There are no specific programs which have special roles.
(2) "$" cannot be used in the first character of the Motion SFC program name.
(3) "\ / : ; , . * ? " < > |" cannot be used in Motion SFC program name.

4.5 Steps

4.5.1 Motion control step

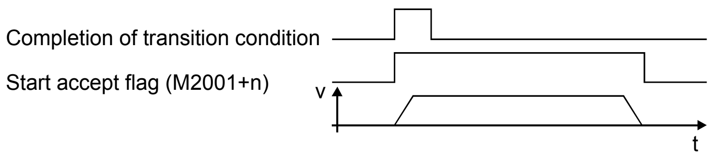
Name	Symbol	Function
Motion control step	Kn	Starts the servo program Kn. Specified range: K0 to K4095

[Operations]

(1) Turns on the start accept flag of the axis specified with the specified servo program Kn running.

(2) Starts the specified servo program Kn.

Execution timing



[Errors]

(1) When the specified servo program Kn does not exist, the Motion SFC error (error code: 16200) will occur and stops to execute the Motion SFC program at the error detection.

[Instructions]

(1) When the current value change is executed in the Motion SFC program running, specify the CHGA instruction in the servo program and call it at the motion control step.

(2) If the servo program has stopped due to a major/minor error which occurred at or during a start of the servo program specified with the motion control step, the Motion SFC program continues executing. When the Motion SFC program is stopped at error detection, provide an error detection condition at the transition (transition condition).

(3) Refer to Chapter 7 for servo programs that can be described in Motion control steps.

4.5.2 Operation control step

Name	Symbol	Function
Operation control step	Fn/FSn	Executes the operation control program Fn/FSn. Specified range: F0 to F4095/FS0 to FS4095

[Operations]

(1) Once execution type operation control step Fn
In the case of Fn, executes the specified operation control program Fn once.

(2) Scan execution type operation control step FSn
In the case of FSn, repeats the specified operation control program FSn until the next transition condition enables.

[Errors]

(1) When the specified operation control program Fn/FSn does not exist, the Motion SFC error (error code: 16201) will occur and stops to execute the Motion SFC program at the error detection.

[Instructions]

(1) Refer to Chapter 5 for operation expressions that may be described in operation control programs.

(2) If an operation or similar error occurs the operation control program running, the Motion SFC program continues executing.

4.5.3 Subroutine call/start step

Name	Symbol	Function
Subroutine call/start step	Program name	Calls/starts the Motion SFC program of the specified program name.

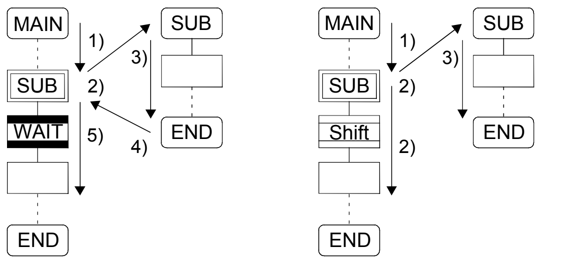
[Operations]

(1) Calls/starts the Motion SFC program of the specified program name.

(2) Control varies with the type of the transition coupled next to the subroutine call/start step.

(a) WAIT (Subroutine Call)
When the subroutine call step is executed, control transits to the specified program as shown below, and when END of the called program is executed, control returns to the call source program.

(b) Except WAIT (Subroutine Start)
When the subroutine start step is executed, control starts the specified program and then shifts to the next as shown below. Since, the start source and destination Motion SFC programs are executed in parallel. The started program ends at END execution.



WAIT (Subroutine call)　　Except WAIT (Subroutine start)

[Errors]

(1) When the specified Motion SFC program does not exist at a subroutine call/start, the Motion SFC error (error code: 16005) will occur and stops to execute the Motion SFC program at the error detection.

(2) When the called/started Motion SFC program is already starting at a subroutine call/start, the Motion SFC error (error code: 16006) will occur and stops to execute the Motion SFC program at the error detection.

(3) When the self program is started at a subroutine call/start, the Motion SFC error (error code: 16110) will occur and stops to execute the Motion SFC program at the error detection.

(4) When the subroutine to be called/started at a subroutine call/start in the Motion SFC program 2 running which was called/started from the Motion SFC program 1 is the Motion SFC program 1 (call source/start program), the Motion SFC error (error code: 16111) will occur and the call/start source Motion SFC program 2 running is stopped at the point of error detection.

[Instructions]

(1) There are no restrictions on the depth of subroutine call/start nesting.

(2) For a subroutine start, the start source Motion SFC program continues processing if the start destination Motion SFC program stops due to an error.

(3) For a subroutine call, the call source Motion SFC program stops running as soon as the call destination Motion SFC program stops due to an error.

4.5.4 Clear step

Name	Symbol	Function
Clear step	CLR Program name	Stops the Motion SFC program of the specified program name.

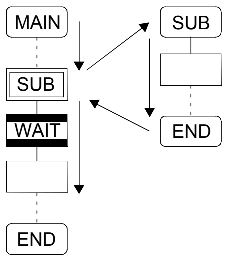
[Operations]

(1) Stops the specified Motion SFC program running.

(2) The clear-specified Motion SFC program will not start automatically after stopped if it has been set to start automatically.

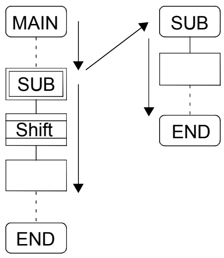
(3) The specified program may be its self program.

(4) If the specified program is being subroutine called, the subroutine program called is also stopped. (Shown below)



If the program has been "subroutine called" as shown on the left
- When the call source program (MAIN) is cleared.
  Even if the subroutine (SUB) is running, both the call source program (MAIN) and subroutine (SUB) stop running.
- When the subroutine (SUB) is running, the subroutine (SUB) stops running and execution to the call source program (MAIN).

(5) When the specified program has been subroutine started, the subroutine program started continues processing. (Shown below)



If the program has been "subroutine started" as shown on the left
- When the started program (MAIN) is cleared.
  Even if the subroutine (SUB) is running, the started program (MAIN) stops running but the started subroutine (SUB) continues processing.
- When the subroutine (SUB) is cleared
  If the subroutine (SUB) is running, only the subroutine (SUB) stops running.

(6) When the servo program started from the specified program is starting, the servo program continues processing.

(7) The servo program is executed after waiting completion of condition in "WAITON/WAITOFF + motion control step". Input the stop command of target axis in addition not to execute the servo program.

[Errors]

(1) When the Motion SFC specified with the clear step does not exist, the Motion SFC error (error code: 16203) will occur.

[Instructions]

(1) When the Motion SFC program specified with the clear step is not starting, an error does not occur specifically and this step is ignored.

(2) If the Motion SFC program running is stopped by the clear step, the output is held.

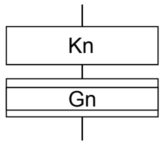
(3) Input the stop command of target axis in addition to stop an operating axis with the clear step execution.

4.6 Transitions

You can describe conditional and operation expressions at transitions. The operation expression described here is repeated until the transition condition enables, as at the scan execution type operation step.
Refer to Chapter 6 for the conditional/operation expressions that can be described in transition conditions.

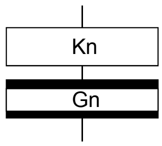
(1) Combinations with motion control steps

(a) Motion control step + Shift



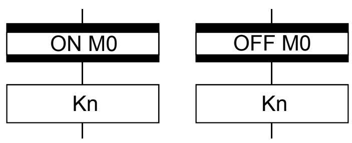
[Operations]
- Transits to the next step by formation of transition condition Gn without waiting for the operating completion of the servo program Kn started at the motion control step.

(b) Motion control step + WAIT



[Operations]
- Waits for the operating completion of the servo program Kn started at the motion control step, and then transits to the next step by formation of transition condition Gn.
- The operation completion condition of the servo program Kn is not needed in the transition condition Gn.
- An error stop of the started servo program Kn at/during a start is also regarded as an operation completion.

(c) WAITON/WAITOFF + Motion control step



[Operations]
- Prepares for the start of the motion control step next to WAITON/WAITOFF, and makes a start immediately when the specified bit device turns ON/OFF. When the motion control step is executed without being used with WAITON/WAITOFF, preparations for a start are made after the transition condition preceding the motion control step enables. This will cause a variation of delay/starting time between when the transition condition is completed and when a start is made, but a combination with WAITON/WAITOFF can eliminate the variation of the above delay/starting time.

• Specifiable bit devices

Device	Range
X	X0 to X1FFF (Note-1)
Y	Y0 to Y1FFF
M	M0 to M12287
U□\G	U□\G10000.0 to U□\G (10000+p-1).F (Note-2) (□: CPU No. (No.1: 3E0, No.2: 3E1, No.3: 3E2, No.4: 3E3) CPU No. that is lager than the number of Multiple CPU cannot be set.)
B	B0 to B1FFF
F	F0 to F2047
SM	SM0 to SM2255

(Note-1): The range of "PXn+4 to PXn+F" cannot be used (fixed at 0) for the input device (PXn+0 to PXn+F) allocated to the built-in interface in Motion CPU (DI). (n: First input No.) QDS

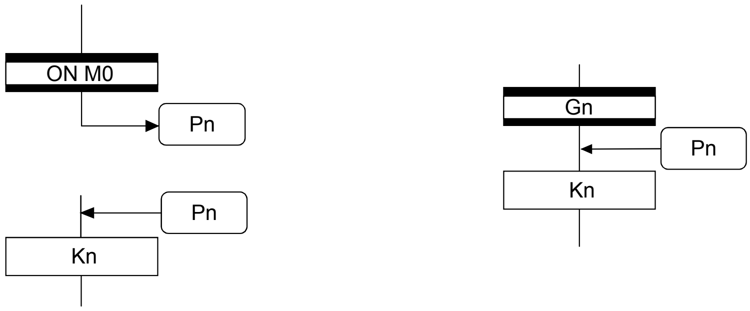
(Note-2): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for each CPU.

POINT

Refer to Chapter 2 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.

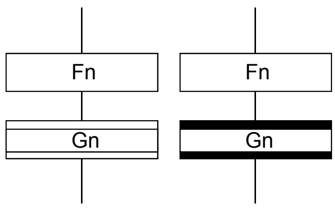
[Instructions]

- Always pair a transition with a motion control step one-for-one. If the step following WAITON/WAITOFF is not a motion control step, the Motion SFC error (error code: 16102) will occur and the Motion SFC program running will stop at the error detection.
- An error will not occur if the jump destination immediately after WAITON/WAITOFF is a motion control step. (Left below)
- A pointer may exist immediately after WAITON/WAITOFF. (Right below)



- If the servo program specified with a motion control step could not be started due to a major/minor error, the Motion SFC program continues running and execution shifts to the next, independently of the WAITON/WAITOFF bit device status. To stop the Motion SFC program at error detection, provide an error detection condition at the next transition (transition condition).
- The following instructions can be used in the motion control step used combining the WAITON/WAITOFF.
  (Linear interpolation control, circular interpolation control, helical interpolation, speed switching control, position follow-up control, constant-speed control, high speed oscillation and speed control with fixed position stop.)

(2) Combination with operation control step



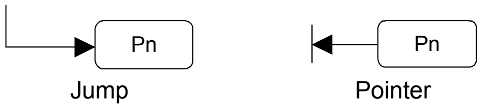
[Operations]

- At an operation control step, both Shift and WAIT perform the same operation, and after executing of the operation control program Fn, transits to the next step by formation of transition condition Gn.

(3) Combination with subroutine call/start step

Refer to Section "4.5.3 Subroutine call/start step".

4.7 Jump, Pointer

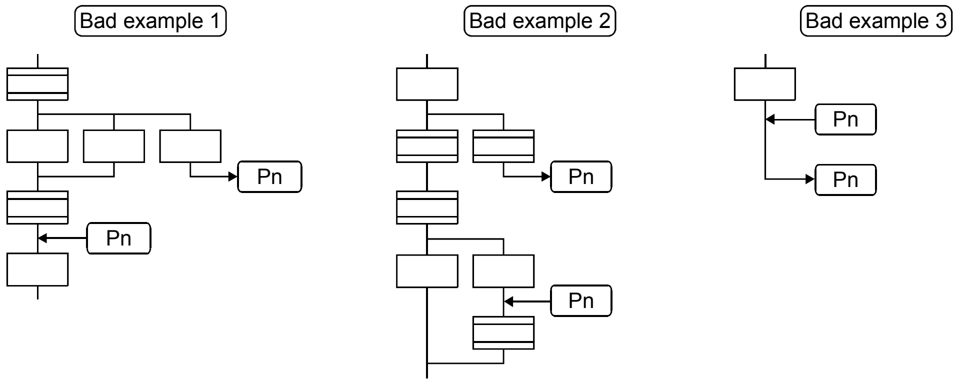


[Operations]

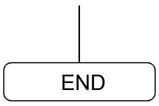
- Setting a jump will cause a jump to the specified pointer Pn of the self program.
- You can set pointers at steps, transitions, branch points and coupling points.
- You can set pointers Pn at P0 to P16383 in one program.

[Instructions]

- You cannot make a jump setting which will exit from within parallel branch-parallel coupling. Connect directly. (Bad example 1 given below)
- You cannot make a jump setting from outside parallel branch-parallel coupling to within parallel branch-parallel coupling. (Bad example 2 given below)
- You cannot make a setting where a label and a jump will continue. (Bad example 3 given below)



4.8 END



[Operations]

- Ends a program. (In this case of an event task or NMI task, operation changes with end operation setting of the program parameter. Refer to Section "9.12 Program Parameters" for details.)
- Making a subroutine call will return to the call source Motion SFC program.

[Instructions]

- END may be set a multiple number of times in one program.
- END cannot be set between a parallel branch and a parallel coupling.
- The output is held after the Motion SFC program is ended by END.

4.9 Branches, Couplings

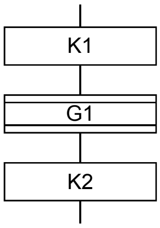
4.9.1 Series transition

Transits execution to the subsequent step or transition connected in series.

(1) To start a servo program or subroutine and shift execution to the next without waiting for operation completion

Set Shift at a transition.

In this case, the transition (shift) may be omitted. When you omitted the transition, an unconditional shift transition is performed.



Starts the servo program K1.

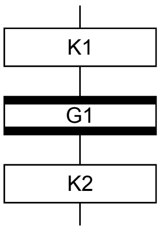
Transits to next by the completion of condition set at transition G1 without waiting for operating completion of the servo program K1.

Starts the servo program K2.

POINT
For a subroutine start, self program and a subroutine program are processed in parallel.

(2) To start a servo program or subroutine and proceed to the next step on operation completion

Set WAIT at a transition.



Starts the servo program K1.

Transits to next when the start axis stops in the servo program K1 (start accept flag turns OFF) and condition is completed set at transition G1.

Starts the servo program K2.

POINT

(1) The above start accept flag of the axis started in the next servo program K2 is not included in interlocks.
To use it as an interlock, the user should set it in the transition condition G1.

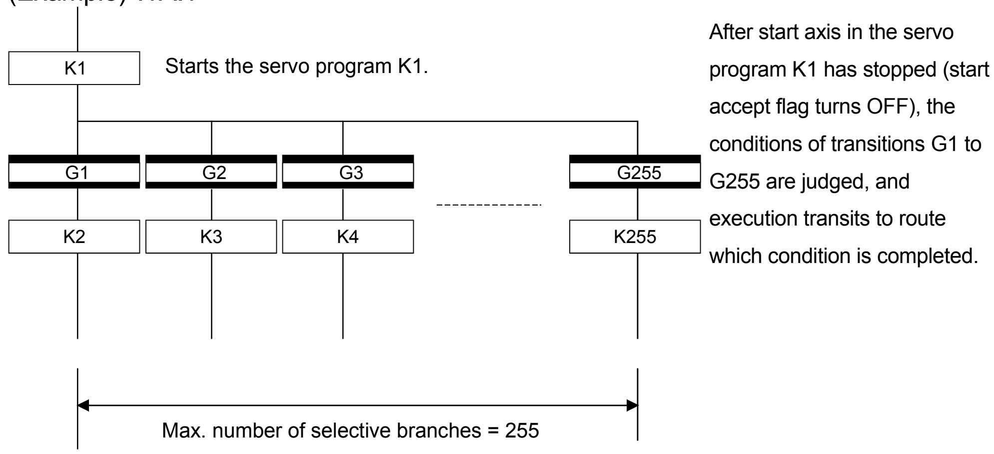
(2) WAIT must be set to proceed to the next step on operation completion.
However, when there are specifically no conditions to be set as interlocks, set "NOP (No Operation)" in the transition program (Gn).

4.9.2 Selective branch, selective coupling

(1) Selective branch

Executes only the route which condition was judged to have enabled first among the conditions of multiple transitions connected in parallel. Transitions must be all Shifts or WAITs.

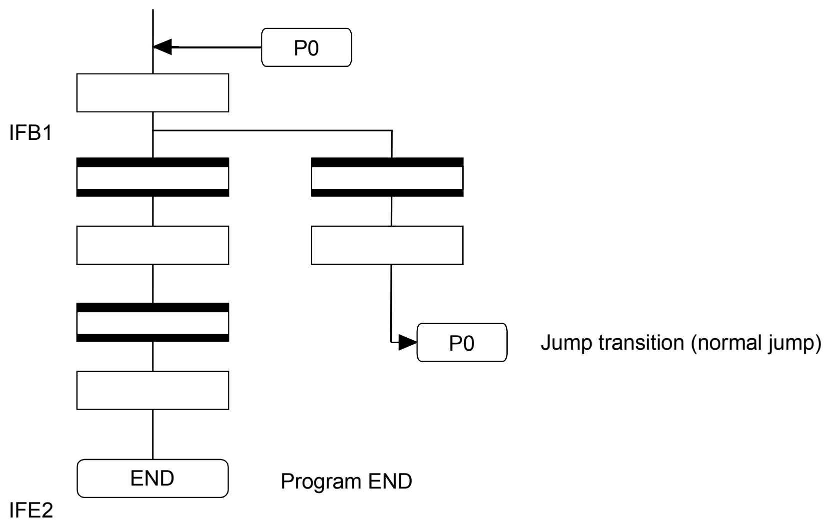
(Example) WAIT



POINT
(1) Transition condition judgment is not always executed from left to right. (2) Using Shift and WAIT together will cause a parallel branch.

(2) Selective coupling

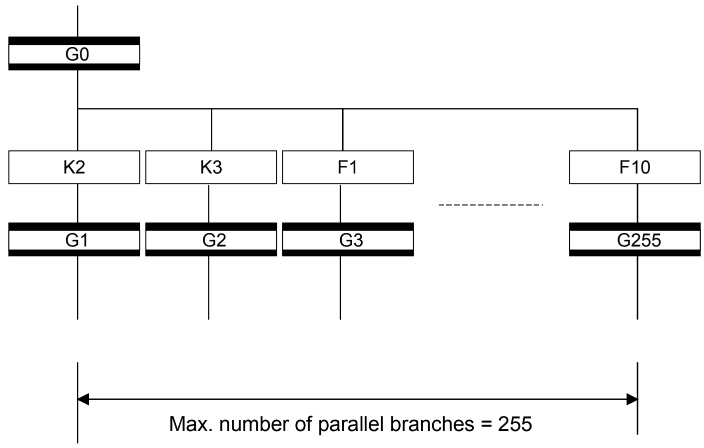
Recoupling of routes into a single route after their processing completions following a selective branch will be a selective coupling. However, you can also make a setting where no coupling will be made as shown below.



4.9.3 Parallel branch, parallel coupling

(1) Parallel branch

Multiple routes connected in parallel are executed simultaneously. Each parallel branch destination may be started by either a step or a transition.

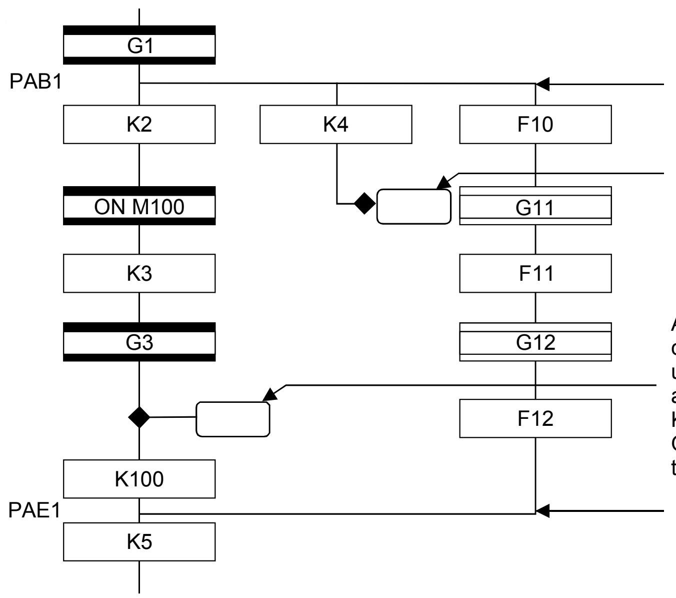


POINT

"Shift" or "WAIT" can be set to a transition preceding a parallel branch.
"WAITON" and "WAITOFF" cannot be set.

(2) Parallel coupling

A parallel branch must be coupled by a parallel coupling. A jump setting to another branch route can be made within parallel branch-parallel coupling.
In this case, a jump destination is a midway parallel coupling point (coupling jump).
You cannot set a jump to exit from within parallel branch-parallel coupling.

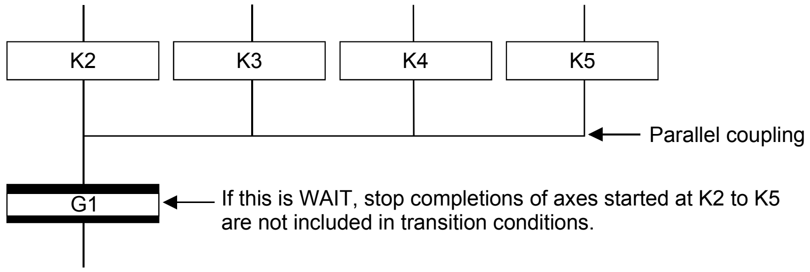


POINT

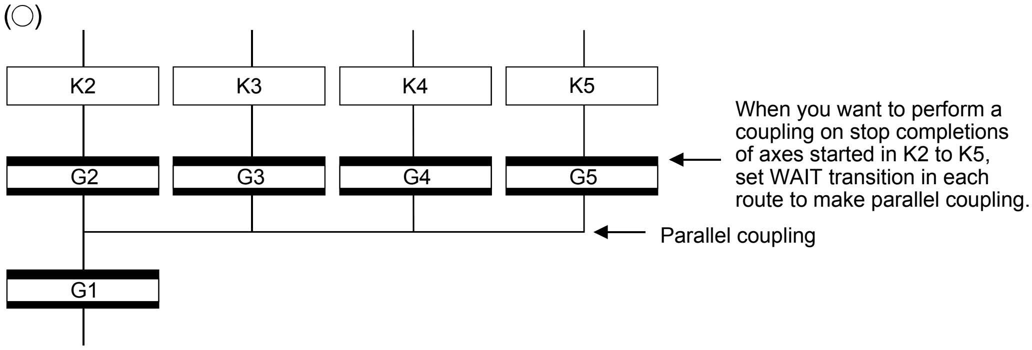
The number of parallel branches need not match that of couplings at a parallel coupling point.
(In the example of the diagram in Section 4.9.3 (2), the number of parallel branches is 3 and that of couplings is 2.)

When a WAIT transition is set right after a parallel coupling, the stop completions of the axes are not included in the waiting conditions if the parallel coupling is preceded by motion control steps. To perform a parallel coupling on stop completions, set WAIT transitions before a parallel coupling.

(×)

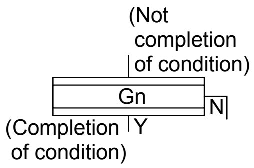
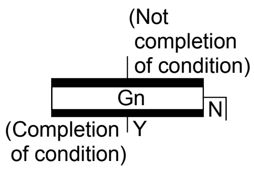


(○)



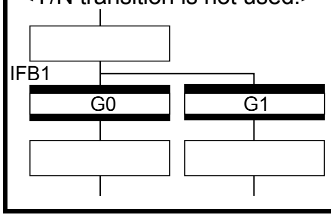
4.10 Y/N Transitions

When routes are branch at a transition condition enables and disable, "Shift Y/N transition" or "WAIT Y/N transition" will be useful.

Name	Symbol	Function
Shift Y/N transition		• When a transition condition set at Gn enables, execution shifts to the lower step. When that condition disables, execution shifts to the right-connected step. • Differences between "Shift Y/N" and "WAIT Y/N" are the same as those between "Shift" and "WAIT".
WAIT Y/N transition		

A Y/N transition is designed to describe the following two-route selective branch program easily.

<Y/N transition is not used.>



- G0 and G1 programs should be different only in acknowledgement/negation of the conditional expressions.

(Example 1)

[G 0] M0

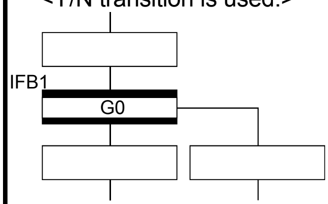
[G 1] !M0

(Example 2)

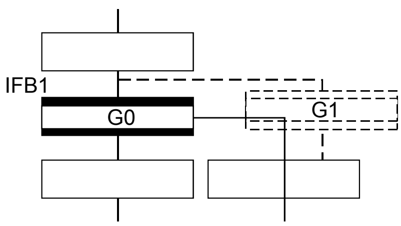
[G 0] D0!=K100

[G 1] D0=K100

<Y/N transition is used.>



- Set the G0 program shown in above (Example 1) or (Example 2) as a G0 program.
- The Motion SFC program list codes after conversion are the same as in the conventional description (different only in the Motion SFC chart representation). Therefore, "automatic search for free G number → automatic generation of program which conditional expression part is logically negated" is performed during program editing to occupy two G programs. Using "Program editor" to delete a Y/N transition does not delete the automatically generated G program (G1 below). Use "Program use list" to delete that program.



(1) Automatic free G number search feature

(a) When not set to automatic numbering
Searches for a free number forward, starting with the "set G number + 1" at the "Shift Y/N" or "WAIT Y/N" symbol.
When no free numbers are found after a search up to 4095, a search is made from 0 to the "set G number - 1".

(b) When set to automatic numbering
Searches for a free number forward (or backward) in the automatic numbering range, starting with the "automatically numbered G number + 1 (or -1)" at the "Shift Y/N" or "WAIT Y/N" symbol. (The searching method is as in the automatic numbering setting.)

(2) Automatic logical NOT program generation feature

Automatically generates a program which logically negates the conditional expression block (last block) of the transition program set at "Shift Y/N" or "WAIT Y/N".
The basic is shown below.

<Setting program (conditional expression block)>

Conditional expression//(bit conditional expression or comparison conditional expression)

⬇

<Logically negated, automatically generated program (conditional expression block)>

!Conditional expression//(bit conditional expression or comparison conditional expression)

Examples are shown below.

<Setting program (conditional expression block)>

(Example 1)

M0	//Bit device ON

(Example 2)

D0!=K100	//Data register D0 is not K100

<Logically negated, automatically generated program (conditional expression block)>

(Example 1)

!(M0)	//Bit device OFF

(Example 2)

!(D0!=K100)	//Data register D0 is K100

POINT

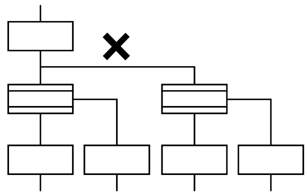
(1) Refer to Section "1.2.3 (2) Table of the operation control/transition instruction" for the instructions usable in the conditional expressions of "Shift Y/N" or "WAIT Y/N" transition programs.
(2) Set conditional expression block only to the setting program.

(3) Instructions for the Motion SFC charts

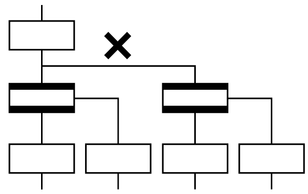
Any Motion SFC chart that will be meaningless to or conflict with the definition of Y/N transitions will result in an error at the time of editing (or Motion SFC chart conversion). Their patterns and instructions will be given below.

(a) When "Shift Y/N" or "WAIT Y/N" is connected as a selective branch or parallel branch: Error

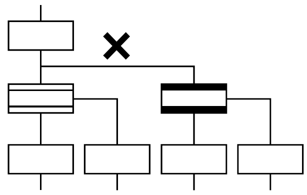
• "Shift Y/N" used as selective branch



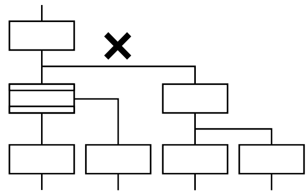
• "WAIT Y/N" used as selective branch



• "Shift Y/N" and "WAIT Y/N" used as parallel branch



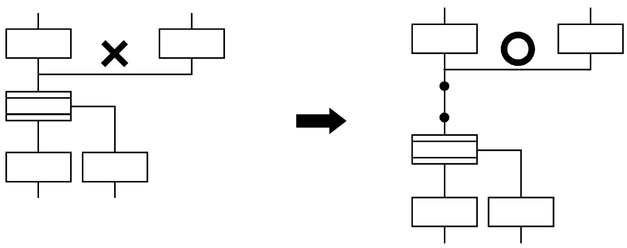
• "Shift (or WAIT) Y/N" used with other step/transition as parallel branch or selective branch



(b) When a coupling precedes "Shift Y/N" or "WAIT Y/N": Provide "coupling-branch continuation" in between.

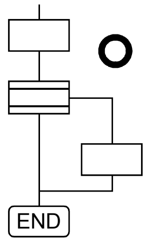
• Direct coupling with "Shift Y/N" or "WAIT Y/N" is not allowed.

• Provide "coupling-branch continuation" in between.

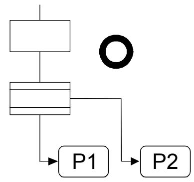


(c) The following patterns may be set.

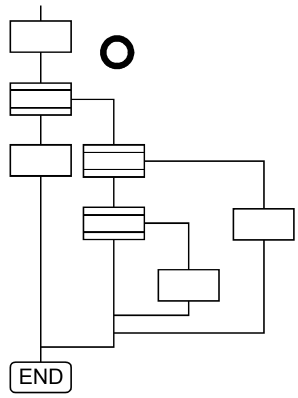
• End (END) from "Shift Y/N" or "WAIT Y/N"



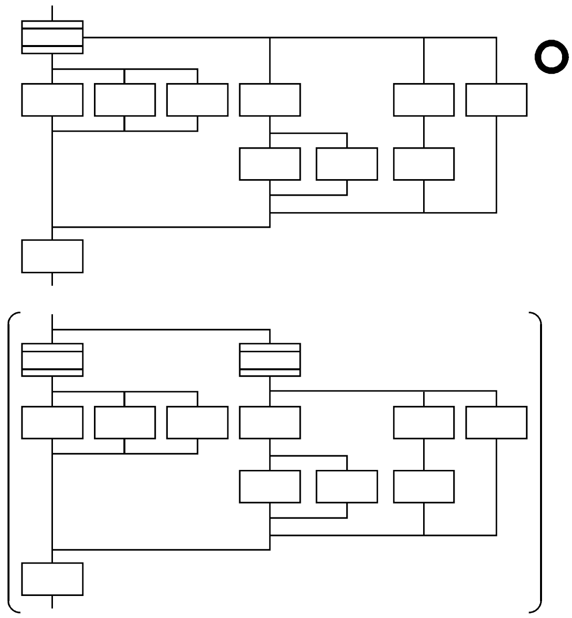
• Jump from "Shift Y/N" or "WAIT Y/N"



• Continuation from "Shift Y/N" or "WAIT Y/N" to "Shift Y/N" or "WAIT Y/N"
(selective branch-selective branch)



• When there are two or more connection lines from Y/N side of "Shift Y/N" or "WAIT Y/N", selective branch continues to selective branch or parallel branch.



4.11 Motion SFC Comments

A comment can be set to each symbol of the step/transition in the motion SFC chart. Comments are shown in the Motion SFC chart by changing the display mode to "Comment display" on the Motion SFC program edit screen.

Classification	Name	Symbol	Comment Setting
Program start/end	START	Program name	Comment setting cannot be made.
	END	END	
Step	Motion control step	Kn	Up to 80 characters Displayed in 20 characters×4 lines
	Once execution type operation control step	Fn	
	Scan execution type operation control step	FSn	
	Subroutine call/start step	Program name	
	Clear step	CLR Program name	
Transition	Shift (preread transition)	Gn	
	WAIT	Gn	
	WAITON	ON bit device	
	WAITOFF	OFF bit device	
	Shift Y/N	Gn	
	WAIT Y/N	Gn	
Jump	Jump	Pn	Up to 64 characters Displayed in 16 characters×4 lines
Pointer	Pointer	Pn	

POINT

(1) Motion SFC comments are stored into the code area of Motion CPU. The code area stores the Motion SFC chart codes, operation control (F/FS) program codes, transition (G) program codes and Motion SFC comments.
Be careful not to set too many comments to avoid code area overflow. (Refer to Section "1.2.2 (1) (b) Motion SFC Performance Specifications" for the code area sizes.)

(2) You cannot use "," in comment statements.

MEMO

5. OPERATION CONTROL PROGRAMS

5.1 Operation Control Programs

(1) Operation control programs

(a) Substitution operation expressions, motion-dedicated functions and bit device control commands can be set in operation control program.

(b) Multiple blocks in one operation control program can be set.

(c) There are no restrictions on the number of blocks that may be set in one operation control program. However, one program is within 64k bytes.

(d) The maximum number of characters in one block is 128.

(e) Transition conditions cannot be set. Transition conditions can be set only in transition programs.

(f) The bit conditional expression that logical data value (true or false) is returned in an operation control program, a comparison conditional expression can be set up only as a source (S) of device set (SET=) or device reset (RST=).

An operation control program example is shown below.

1 block

```
#0=D0+(D1+D2)*#5//Substitution expression (four arithmetic operations)
W0:F=SIN(#10F)//Substitution expression (standard function)
CHGV(K2,K10)//Motion-dedicated function
SET M100=M0+X0//Bit device control (SET=)
RST M10=!X0//Bit device control (RST=)
DIN D0,X0//Bit device control (DIN)
```

1 program

Comment

(2) Priorities of operators and functions

Operators and functions have the following priorities.
Using parentheses allows an operation sequence to be specified freely.

Priority	Item (Operator, Function)
High	Calculation within parentheses ((...))
	Standard function (SIN, COS, etc.), Type conversion (USHORT, LONG, etc.)
	Bit inversion (~), logical negation (!), sign inversion (－)
	Multiplication (*), division (/), remainder (%)
	Addition (+), subtraction (－)
	Bit left shift (<<), bit right shift (>>)
	Comparison operators: Less than (<), less than or equal to (<=), more than (>), more than or equal to (>=)
	Comparison operators: Equal to (==), not equal to (!=)
	Bit logical AND (&)
	Bit exclusive OR (^)
	Bit logical OR (|)
	Logical AND (*)
	Logical OR (+)
Low	Substitution (=)

(3) Structure of instruction

Many of the instructions usable in operation control programs can be divided into instruction and data parts.

The instruction and data parts are used for the following purposes.

- Instruction part.......... Indicates the function of that instruction.
- Data part.................. Indicates the data used in the instruction.

"Substitution: =" structure example

```
D0 = #0
```

Data part: Source (S)
Instruction part
Data part: Destination (D)

(a) Source (S)

1) The source is the data used in an operation.
2) It varies with the device specified in each instruction is shown below.
   - Bit or word device
     Specify the device which stores the data used in operation.
     The data must have been stored in the specified device until the operation is executed.
     Changing the data stored in the specified device during program execution allows changing the data used in that instruction.
   - Constant
     Specify the numerical value used in an operation.
     As the constant is set during program creation, it cannot be changed during program running.

(b) Destination (D)

1) As the destination data, after-operation data is stored.
2) Destination data is always set the device for storing the data.

(4) How to specify data

There are the following six different data usable in each instruction.

- Data usable in each instruction
  - Numerical data
    - Integer data
      - 16-bit integer type data
      - 32-bit integer type data
    - 64-bit floating-point type data
  - Bit data
  - Batch bit data
  - Logical data

(a) 16-bit integer type data

The 16-bit integer type data is 16-bit integer value data.
Word devices are used in increments of 1 point.
Data ranges are shown below.

	Decimal representation	Hexadecimal representation
Data range	K-32768 to K32767	H0000 to HFFFF

(b) 32-bit integer type data

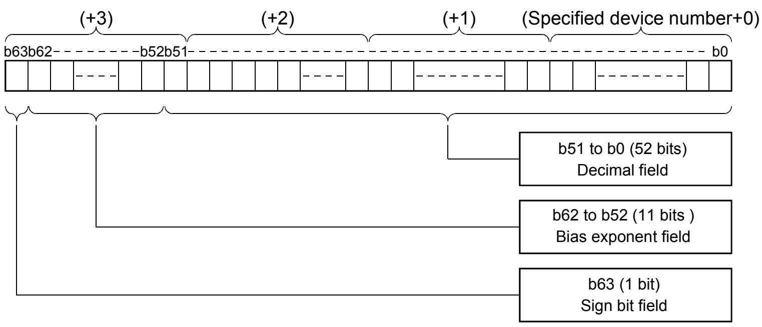
The 32-bit integer type data is 32-bit integer value data.
Word devices are used in increments of 2 points: (specified device No.), (specified device No.+1). Data ranges are shown below.

	Decimal representation	Hexadecimal representation
Data range	K-2147483648L to K2147483647L	H00000000L to HFFFFFFFFL

(c) 64-bit floating-point type data

The 64-bit floating-point type data is IEEE-formatted, 64-bit floating-point value data.
Word devices are used in increments of 4 points: (specified device No.), (specified device No.+1), (specified device No.+2), (specified device No.+3).

1) The internal bit locations are shown below.



2) The represented value is shown below. (The bias value is H3FF.)

$$(-1)^{[\text{Sign bit field}]} * (1.0+[\text{decimal field}]) * 2^{([\text{Bias exponent field}]-[\text{bias value}])}$$

3) Data ranges are shown below.

	Decimal representation	Hexadecimal representation
Data range	K-1.79E+308 to K-2.23E-308, K0.0, K2.23E-308 to K1.79E+308	H0000000000000000, H0010000000000000 to H7FE1CCF385EBC89F, H8000000000000000, H8010000000000000 to HFFE1CCF385EBC89F

4) A round-off error may be produced in a 64-bit floating-point type data operation. Especially when using 64-bit floating-point type data in a comparison operation, note that a round-off error may cause an intended operation.

Example) In the following transition program, the result of the comparison operation may not become true depending on the value of #200F due to a round-off error.

```
#100F=SQRT(#200F)
#300F=#100F*#100F
#200F==#300F
```

(d) Bit data

The bit data is the data where a contact/coil or similar device is handled in increments of 1 bit. It is used in device set (SET=) and device reset (RST=).

Example 1

$$\texttt{SET}\ \underbrace{\texttt{M0}}_{\text{Bit data}}$$

(e) Batch bit data

The batch bit data is the data where bit data is handled in increments of 16/32 points. It is used in device input (DIN) and device output (DOUT).
As indicated below, whether the bit data is handled in increments of 16 or 32 points is governed by the data type of the word device used as an input destination/output source.

	Increments of 16 points	Increments of 32 points
Program example	DIN #0, M0 DOUT M0, D0	DIN #0L, M0 DOUT M0, D0L
Used devices	(Specified device No.) to (specified device No.+15) M0 to M15 in the above program example	(Specified device No.) to (specified device No.+31) M0 to M31 in the above program example

(f) Logical data

The logical data is a value returned by a bit or comparison conditional expression and indicates whether the result is true or false.
Normally, it is used in the conditional expression of a transition program. In an operation control program, the logical data is used in a bit conditional expression set to device set (SET=) or device reset (RST=).

Example 1

$$\mathrm{SET}\ \underbrace{\mathrm{M0}}_{\text{Bit data}} = \underbrace{\mathrm{X10}}_{\text{Logical data}}$$

Example 2

$$\mathrm{RST}\ \underbrace{\mathrm{M5}}_{\text{Bit data}} = !\underbrace{\mathrm{X10} * \mathrm{M100}}_{\text{Logical data}}$$

Example 3 (transition program)

$$\underbrace{\mathrm{D0 == K100}}_{\text{Logical data}}$$

(5) Internal operation data types

For internal operations, when (S1) and (S2) differ in data type, the data of the smaller type is converted into that of the greater type before operation is performed. If the operation result is over the range of processed number in each type, an overflow will occur. However, an operation error will not occur. By converting the set data with the type converting instruction, an overflow may be able to be prevented.

(S1) Operator (S2)

The operators are "+, -, *, /, %"

The data type combinations, and internal operation data types for binary operations are shown below.

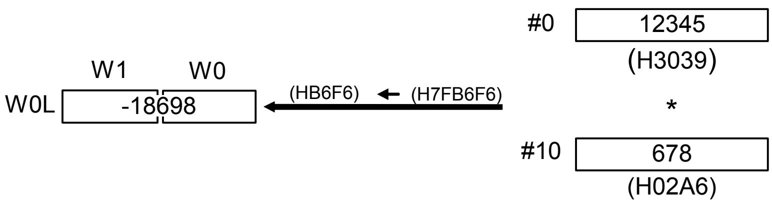
(S1) data type	(S2) data type	Internal operation data type
16-bit integer type	16-bit integer type	16-bit integer type
	32-bit integer type	32-bit integer type
	64-bit floating-point type (Note-1)	64-bit floating-point type (Note-1)
32-bit integer type	16-bit integer type	32-bit integer type
	32-bit integer type	
	64-bit floating-point type (Note-1)	64-bit floating-point type (Note-1)
64-bit floating-point type	16-bit integer type	64-bit floating-point type (Note-1)
	32-bit integer type	
	64-bit floating-point type (Note-1)	

(Note-1): Except the operator "%"

(a) Program example

1) Program which substitutes the result of multiplying #0 by #10 to W0L

W0L = #0 * #10



Since both of set data are the 16-bit integer type, the multiplier result is processed by the 16-bit integer type.

An overflow occurs, and the least 16-bit of the multiplier result is the operation result.

2) Program which substitutes the result of multiplying #0 and #10 to W0L after converting into the 32-bit integer type

```
W0L = LONG(#0) * LONG(#10)
```

Since the multiplier result is processed with the 32-bit integer type by the type converting instruction, even if the device value is the same as the program example 1), an overflow will not occur.

5.2 Device Descriptions

Word and bit device descriptions are shown below.

(1) Word device descriptions

<table>
<tr><th rowspan="4"></th><th colspan="3">Device descriptions</th><th colspan="4">Device No. (n) specified ranges</th></tr>
<tr><th rowspan="3">16-bit integer type</th><th rowspan="3">32-bit integer type ("n" is even No.)</th><th rowspan="3">64-bit floating-point type ("n" is even No.)</th><th colspan="3">Q173DSCPU/Q172DSCPU</th><th rowspan="3">Q173DCPU(-S1)/ Q172DCPU(-S1)</th></tr>
<tr><th rowspan="2">SV13</th><th colspan="2">SV22</th></tr>
<tr><th>Virtual mode switching method</th><th>Advanced synchronous control method Ver.!</th></tr>
<tr><td>Data register</td><td>Dn</td><td>DnL</td><td>DnF</td><td colspan="2">0 to 8191</td><td>0 to 19823</td><td>0 to 8191</td></tr>
<tr><td>Link register</td><td>Wn</td><td>WnL</td><td>Wn:F</td><td colspan="4">0 to 1FFF</td></tr>
<tr><td>Special register</td><td>SDn</td><td>SDnL</td><td>SDnF</td><td colspan="4">0 to 2255 (Note-1)</td></tr>
<tr><td>Motion register</td><td>#n</td><td>#nL</td><td>#nF</td><td colspan="4">0 to 12287</td></tr>
<tr><td>Multiple CPU area device</td><td>U□\Gn (Note-2)</td><td>U□\GnL (Note-2)</td><td>U□\GnF (Note-2)</td><td colspan="4">10000 to (10000+p-1) (Note-3)</td></tr>
<tr><td>Coasting timer</td><td>—</td><td>FT</td><td>—</td><td colspan="4">—</td></tr>
</table>

(Note-1): The range of "2000 to 2255" cannot be specified indirectly.

(Note-2): □ = CPU No. (CPU No.1: 3E0, CPU No.2: 3E1, CPU No.3: 3E2, CPU No.4: 3E3). A CPU No. that exceeds the number of CPUs in the Multiple CPU system cannot be set.

(Note-3): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for each CPU.

(a) For differentiation, the 32-bit floating-point type is ended by L and the 64-bit floating-point type by F (":F" for the link register).

(b) For the 32-bit integer type and 64-bit floating-point type, specify the device number with an even number. (It cannot be set as an odd number.)

(c) The coasting timer FT is incremented per 888[μs]. (The coasting timer is a 32-bit integer type.)

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(2) Bit device descriptions

	Device description	Device No. (n) specified ranges
Input relay	Xn/PXn	0 to 1FFF (Note-1)
Output relay	Yn/PYn	0 to 1FFF
Internal relay	Mn	0 to 12287
Multiple CPU area device	U□\Gn (Note-2)	10000 to (10000+p-1) (Note-3)
Link relay	Bn	0 to 1FFF
Annunciator	Fn	0 to 2047
Special relay	SMn	0 to 2255 (Note-4)

(Note-1): The range of "PXn+4 to PXn+F" cannot be used (fixed at 0) for the input device (PXn+0 to PXn+F) allocated to the built-in interface in Motion CPU (DI).
(n: First input No.) QDS

(Note-2): □ = CPU No. (CPU No.1: 3E0, CPU No.2: 3E1, CPU No.3: 3E2, CPU No.4: 3E3). A CPU No. that exceeds the number of CPUs in the Multiple CPU system cannot be set.

(Note-3): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for each CPU.

(Note-4): The range of "2000 to 2255" cannot be specified indirectly.

(a) When using the device in DIN or DOUT as batch bit data, specify "n" as a multiple of 16.

(b) When using the device in Multiple CPU area device as batch bit data, specify it as word device without making bit specification.

(3) Indirect specification of device No.

In the above word/bit device descriptions, device No. (n) can be specified indirectly.

(a) Indirect specification of device No. (n) using word device

- The word device which the device No. was specified indirectly cannot be used.
- You can use the 16-bit and 32-bit integer type word devices for indirect specification.
  The 64-bit floating-point type cannot be used.
- The word devices that can indirectly specify device No.(n) are shown below.
  - Data register (D)
  - Link register (W)
  - Motion register (#)
  - Special register (SD)

(Description examples)

Good example	Bad example
#(D10)	#(D(D5))
D(#10L)F	D(#4F)

(b) Indirect specification of device No. (n) using word device using operation expression

- Device No. can be specified indirectly by calculation expressions which use the following data and operators.

Usable data	16-bit integer type word device
	32-bit integer type word device
	16-bit integer type constant
	32-bit integer type constant
Usable operators	Addition: +
	Subtraction: －
	Multiplication: *
	Division: /
	Remainder: %
	Sign inversion: －

- The word device which the device No. is specified indirectly cannot be used.
- Only one operator may be used.

(Description examples)

Good example	Bad example
#(D10-K5)	#(D(D5)F+K20)
D(#10L%H6L)F	D(#4L<<K2)

(Note) : When you want to use the result of calculation other than the above to specify the device No. indirectly, describe it in two blocks as shown below.

```
D0=SHORT(ASIN(#0F))
W0=#(D0)
```

POINT

Refer to Chapter 2 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.

5.3 Constant Descriptions

The constant descriptions of the 16-bit integer type, 32-bit integer type and 64-bit floating-point type are shown below.

	16-bit integer type	32-bit integer type	64-bit floating-point type
Decimal representation	K-32768 to K32767	K-2147483648L to K2147483647L	K-1.79E+308 to K-2.23E-308, K0.0, K2.23E-308 to K1.79E+308
Hexadecimal representation	H0000 to HFFFF	H00000000L to HFFFFFFFFL	—

(1) The 32-bit integer type is ended by L and the 64-bit floating-point type is provided with a decimal point and exponent part (E) to denote their data types explicitly.

(2) The constant without the data type is regarded as the applicable minimum type.

(3) The constant in decimal representation is headed by K and the one in hexadecimal representation by H.
K can be omitted.

(4) The 64-bit floating-point type cannot be represented in hexadecimal.

F/FS	G
○	○

5.4 Binary Operations

5.4.1 Substitution : =

Format	(D)=(S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—
(D)	—	○	○	○	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Word device/constant/calculation expression to be substituted	Data type of (D)
(D)	Word device which will store the operation result	

[Functions]

(1) The data value specified with (S) is substituted to the specified word device at (D).

(2) When (S) and (D) differ in data type, the data at (S) is converted into the data type of (D) and the resultant data is substituted.
(When (D) is a 16- or 32-bit integer type and (S) is a 64-bit floating-point type, the fraction part of (S) is discarded.)

[Errors]

(1) An operation error will occur if:
- The data at (S) is outside the data type range of (D); or
- (D) or (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which substitutes the D0 value to #0

```
#0 = D0
```

#0 [123] ← D0 [123]

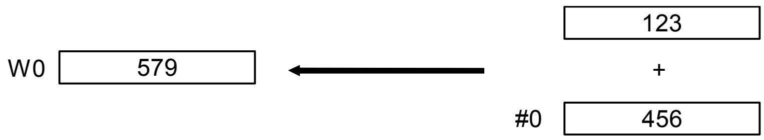
(2) Program which substitutes K123456.789 to D0L

```
D0L = K123456.789
```

The 64-bit floating-point type is converted into the 32-bit integer type and the result is substituted.

(3) Program which substitutes the result of adding K123 and #0 to W0

```
W0 = K123 + #0
```



F/FS	G
○	○

5.4.2 Addition : +

Format	(S1)+(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Augend data	Data type of (S1) or (S2) which is greater
(S2)	Addend data	

[Functions]

(1) The data specified with (S2) is added to the data specified with (S1).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

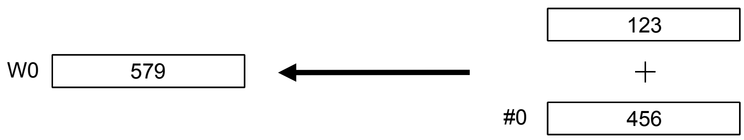
[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

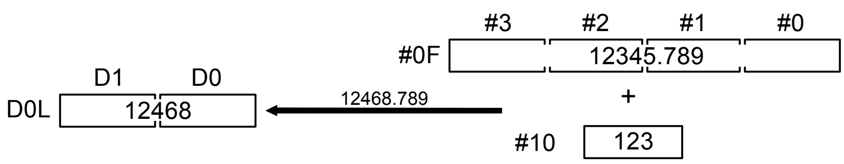
(1) Program which substitutes the result of adding K123 and #0 to W0

```
W0 = K123 + #0
```



(2) Program which substitutes the result of adding #0F and #10 to D0L

```
D0L = #0F + #10
```



The 64-bit floating-point type data are used for addition, and the result is converted into the 32-bit integer type and then substituted.

F/FS	G
○	○

5.4.3 Subtraction : −

Format	(S1)−(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Minuend data	Data type of (S1) or (S2) which is greater
(S2)	Subtracted data	

[Functions]

(1) The data specified with (S2) is subtracted from the data specified with (S1).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

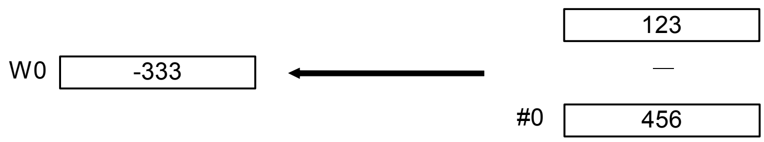
[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

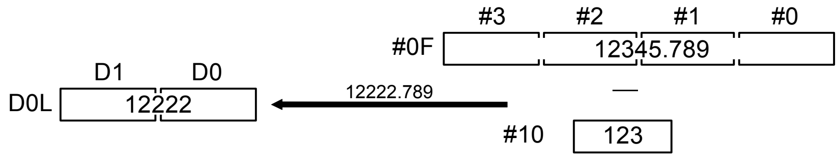
(1) Program which substitutes the result of subtracting #0 from K123 to W0

```
W0 = K123−#0
```



(2) Program which substitutes the result of subtracting #10 from #0F to D0L

```
D0L = #0F−#10
```



The 64-bit floating-point type data are used for subtraction, and the result is converted into the 32-bit integer type and then substituted.

F/FS	G
○	○

5.4.4 Multiplication : ∗

Format	(S1)∗(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Multiplicand data	Data type of (S1) or (S2) which is greater
(S2)	Multiplier data	

[Functions]

(1) The data specified with (S1) is multiplied by the data specified with (S2).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

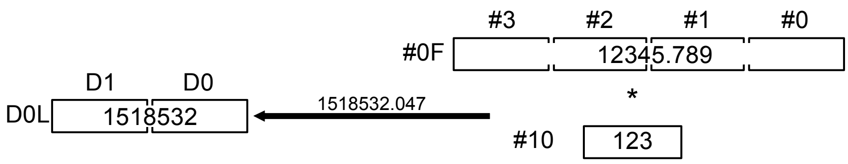
[Program examples]

(1) Program which substitutes the result of multiplying K123 by #0 to W0

```
W0 = K123*#0
```

(2) Program which substitutes the result of multiplying #0F by #10 to D0L

```
D0L = #0F*#10
```



The 64-bit floating-point type data are used for multiplication, and the result is converted into the 32-bit integer type and then substituted.

F/FS	G
○	○

5.4.5 Division : /

Format	(S1)/(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Dividend data	Data type of (S1) or (S2) which is greater
(S2)	Divisor data	

[Functions]

(1) The data specified with (S1) is divided by the data specified with (S2) to find a quotient.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

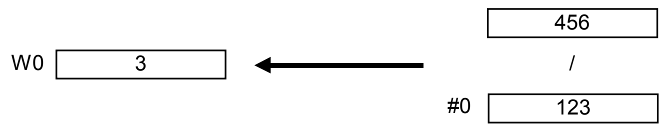
[Errors]

(1) An operation error will occur if:
- (S2) is 0; or
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

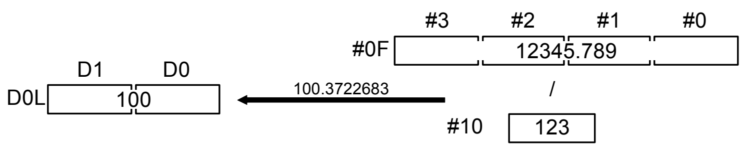
(1) Program which divides K456 by #0 and substitutes a quotient to W0

```
W0 = K456 / #0
```



(2) Program which divides #0F by #10 and substitutes a quotient to D0L

```
D0L = #0F / #10
```



The 64-bit floating-point type data are used for division, and the quotient is converted into the 32-bit integer type and then substituted.

F/FS	G
○	○

5.4.6 Remainder : %

Format	(S1)%(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Dividend data	Data type (integer type) of (S1) or (S2) which is greater (Integer type)
(S2)	Divisor data	

[Functions]

(1) The data specified with (S1) is divided by the data specified with (S2) to find a remainder.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S2) is 0; or
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which divides K456 by #0 and substitutes a remainder to W0

```
W0 = K456 % #0
```

456

W0 87 ← %

#0 123

F/FS	G
○	○

5.5 Bit Operations

5.5.1 Bit inversion (Complement) : ~

Format	~(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data											
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose bits will be inverted	Data type of (S) (Integer type)

[Functions]

(1) The bit inverted value of the data specified with (S) is found.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the bit inverted value of #0 and substitutes the value to D0

```
D0 = ~#0
```

	b15 ··· b0			b15 ··· b0
D0	1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1	←	#0	0 0 1 0 0 1 0 1 0 0 1 1 0 1 0 0

F/FS	G
○	○

5.5.2 Bit logical AND : &

Format	(S1)&(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ANDed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

(1) The bit-by-bit logical product of the data specified with (S1) and the data specified with (S2) is found.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

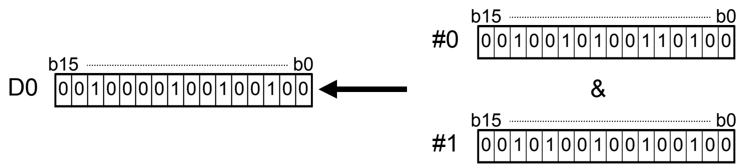
[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which ANDs #0 and #1 and substitutes the result to D0

```
D0 = #0 & #1
```



F/FS	G
○	○

5.5.3 Bit logical OR : |

Format	(S1) I (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ORed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

(1) The bit-by-bit logical add of the data specified with (S1) and the data specified with (S2) is found.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

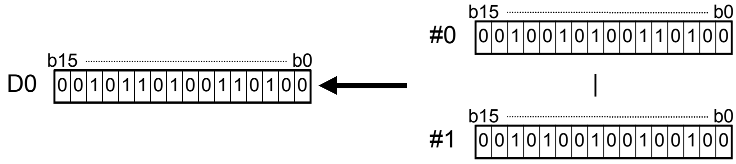
[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which ORs #0 and #1 and substitutes the result to D0

```
D0 = #0 I #1
```



F/FS	G
○	○

5.5.4 Bit exclusive logical OR : ^

Format	(S1)^(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be EXCLUSIVE ORed bit-by-bit	Data type of (S1) or (S2) which is greater (Integer type)
(S2)		

[Functions]

(1) The bit-by-bit exclusive logical add of the data specified with (S1) and the data specified with (S2) is found.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before operation is performed. At this time, note that signed data is converted.

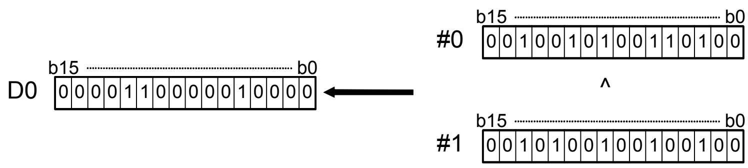
[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which EXCLUSIVE ORs #0 and #1 and substitutes the result to D0

```
D0 = #0 ^ #1
```



F/FS	G
○	○

5.5.5 Bit right shift : >>

Format	(S1) >> (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data to be right-shifted	Data type of (S1) (Integer type)
(S2)	Number of right shifts	

[Functions]

(1) The data specified with (S1) is shifted to the right by the number of times specified with (S2).

(2) If the most significant bit of (S1) is 1, 1 enters the most significant bit of the right shift result.
If the most significant bit of (S1) is 0, 0 enters the most significant bit of the right shift result.

(3) When (S1) is a 16-bit integer type and (S2) is a negative number or not less than 16, the result is 0.

(4) When (S1) is a 32-bit integer type and (S2) is a negative number or not less than 32, the result is 0.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which shifts #0 two bit positions to the right and substitutes the result to D0

```
D0 = #0 >> K2
```

	b15															b0
D0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1

←

	b15															b0
#0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0	0

F/FS	G
○	○

5.5.6 Bit left shift : <<

Format	(S1) << (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data to be left-shifted	Data type of (S1) (Integer type)
(S2)	Number of left shifts	

[Functions]

(1) The data specified with (S1) is shifted to the left by the number of times specified with (S2).

(2) 0 enters the least significant bit of the left shift result.

(3) When (S1) is a 16-bit integer type and (S2) is a negative number or not less than 16, the result is 0.

(4) When (S1) is a 32-bit integer type and (S2) is a negative number or not less than 32, the result is 0.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which shifts #0 one bit position to the left and substitutes the result to D0

```
D0 = #0 << K1
```

```
     b15 ........................... b0         b15 ........................... b0
D0   |0|1|0|0|1|0|1|0|0|1|1|0|1|0|0|0|  ←  #0   |0|0|1|0|0|1|0|1|0|0|1|1|0|1|0|0|
```

F/FS	G
○	○

5.5.7 Sign inversion (Complement of 2) : −

Format	−(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose sign will be inverted	Data type of (S)

[Functions]

(1) The sign-inverted value of the data specified with (S) is found.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which substitutes the sign-inverted value of #0 to D0

```
D0 =−#0
```

D0 [-123] ← #0 [123]

F/FS	G
○	○

5.6 Standard Functions

5.6.1 Sine : SIN

Format	SIN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which SIN (sine) operation will be performed	Floating-point type

[Functions]

(1) SIN (sine) operation is performed on the data specified with (S).

(2) The data specified with (S) is in an angle (degree) unit.

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which performs the SIN operation of D0 and substitutes the result to #0F

```
#0F = SIN(D0)
```

```
        #3    #2    #1    #0
#0F  [ 0.707106781186 55 ]   <-----   D0 [ 45 ]
```

F/FS	G
○	○

5.6.2 Cosine : COS

Format	COS(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which COS (cosine) operation will be performed	Floating-point type

[Functions]

(1) COS (cosine) operation is performed on the data specified with (S).

(2) The data specified with (S) is in an angle (degree) unit.

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which performs the COS operation of D0 and substitutes the result to #0F

```
#0F = COS(D0)
```

	#3	#2	#1	#0			
#0F	0.5				←	D0	60

F/FS	G
○	○

5.6.3 Tangent : TAN

Format	TAN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Angle data on which TAN (tangent) operation will be performed	Floating-point type

[Functions]

(1) TAN (tangent) operation is performed on the data specified with (S).

(2) The data specified with (S) is in an angle (degree) unit.

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range; or
- (S) is 90+(180∗n). ("n" is an integer)

[Program examples]

(1) Program which performs the TAN operation of D0 and substitutes the result to #0F

```
#0F = TAN(D0)
```

	#3	#2	#1	#0			
#0F	0.57735026918963				←	D0	30

F/FS	G
○	○

5.6.4 Arcsine : ASIN

Format	ASIN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	SIN value data on which SIN $^{-1}$ (arcsine) operation will be performed	Floating-point type

[Functions]

(1) SIN $^{-1}$ (arcsine) operation is performed on the SIN value data specified with (S) to find an angle.

(2) The SIN value specified with (S) must be within the range -1.0 to 1.0.

(3) The operation result is in an angle (degree) unit.

(4) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is outside the range -1.0 to 1.0; or
- (S) is an indirectly specified device and its device number is outside the range.

[Program examples]

(1) Program which performs the SIN $^{-1}$ (arcsine) operation of D0 and substitutes the result to #0F

```
#0F = ASIN(D0)
```

	#3	#2	#1	#0			
#0F	90.0				←	D0	1

F/FS	G
○	○

5.6.5 Arccosine : ACOS

Format	ACOS(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	COS value data on which COS $^{-1}$ (arccosine) operation will be performed	Floating-point type

[Functions]

(1) COS $^{-1}$ (arccosine) operation is performed on the COS value data specified with (S) to find an angle.

(2) The COS value specified with (S) must be within the range -1.0 to 1.0.

(3) The operation result is in an angle (degree) unit.

(4) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is outside the range -1.0 to 1.0; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which performs the COS $^{-1}$ (arccosine) operation of D0F and substitutes the result to #0F

```
#0F = ACOS(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	60.0				←	D0F	0.5			

F/FS	G
○	○

5.6.6 Arctangent : ATAN

Format	ATAN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	TAN value data on which TAN $^{-1}$ (arctangent) operation will be performed	Floating-point type

[Functions]

(1) TAN $^{-1}$ (arctangent) operation is performed on the TAN value data specified with (S) to find an angle.

(2) The operation result is in an angle (degree) unit.

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which performs the TAN $^{-1}$ (arctangent) operation of D0F and substitutes the result to #0F

```
#0F = ATAN(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	45.0				←	D0F	1.0			

F/FS	G
○	○

5.6.7 Square root : SQRT

Format	SQRT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which square root operation will be performed	Floating-point type

[Functions]

(1) The square root of the data specified with (S) is found.

(2) Only a positive number may be specified with (S). (Operation cannot be performed with a negative number.)

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

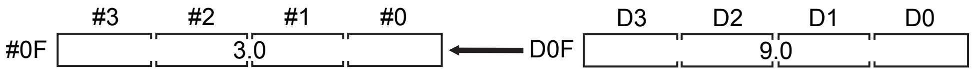
[Errors]

(1) An operation error will occur if:
- (S) is a negative number; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the square root of D0F and substitutes the result to #0F

```
#0F = SQRT(D0F)
```



F/FS	G
○	○

5.6.8 Natural logarithm : LN

Format	LN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which natural logarithm operation will be performed	Floating-point type

[Functions]

(1) The base e natural logarithm of the data specified with (S) is found.

(2) Only a positive number may be specified with (S). (Operation cannot be performed with a negative number.)

(3) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is 0 or a negative number; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the natural logarithm of D0F and substitutes the result to #0F

```
#0F = LN(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	2.302585092994				←	D0F	10.0			

F/FS	G
○	○

5.6.9 Exponential operation : EXP

Format	EXP(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which exponential operation will be performed	Floating-point type

[Functions]

(1) Exponential operation is performed on the base e data specified with (S).

(2) If (S) is an integer type, it is converted into a floating-point type before operation is performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which performs exponential operation of D0F and substitutes the result to #0F

```
#0F = EXP(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	442413.39200892				←	D0F	13.0			

F/FS	G
○	○

5.6.10 Absolute value : ABS

Format	ABS(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data on which absolute value conversion will be performed	Data type of (S)

[Functions]

(1) The absolute value of the data specified with (S) is found.

[Errors]

(1) An operation error will occur if:
- (S) is 16-bit integer type and other than -32767 to 32767.
- (S) is 32-bit integer type and other than -2147483647 to 2147483647.
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the absolute value of D0F and substitutes the result to #0F

```
#0F = ABS(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	33.0				←	D0F	-33.0			

F/FS	G
○	○

5.6.11 Round-off : RND

Format	RND(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded off	Data type of (S)

[Functions]

(1) The rounded-off fractional portion value of the data specified with (S) is found.

(2) If (S) is a negative number, the absolute value of (S) is found and its fractional portion is rounded off and signed.

(3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the rounded-off fractional portion value of D0F and substitutes the result to #0F

```
#0F = RND(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F	34.0				←	D0F	33.54			

(2) Program which finds the rounded-off fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)

```
#0F = RND(D4F)
```

	#3	#2	#1	#0			D7	D6	D5	D4
#0F	-33.0				←	D0F	-33.44			

F/FS	G
○	○

5.6.12 Round-down : FIX

Format	FIX(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded down	Data type of (S)

[Functions]

(1) The largest integer not greater than the data specified with (S) is found.

(2) If the (S) value is positive, the absolute value will be smaller, and if it is negative, the absolute value will be greater.

(3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the rounded-down fractional portion value of D0F and substitutes the result to #0F

```
#0F = FIX(D0F)
```

	#3	#2	#1	#0		D3	D2	D1	D0
#0F		33.0			← D0F		33.54		

(2) Program which finds the rounded-down fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)

```
#0F = FIX(D4F)
```

	#3	#2	#1	#0		D7	D6	D5	D4
#0F		-34.0			← D4F		-33.54		

F/FS	G
○	○

5.6.13 Round-up : FUP

Format	FUP(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data whose fractional portion will be rounded up	Data type of (S)

[Functions]

(1) The smallest integer not less than the data specified with (S) is found.

(2) If the (S) value is positive, the absolute value will be greater, and if it is negative, the absolute value will be smaller.

(3) If (S) is an integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which finds the rounded-up fractional portion value of D0F and substitutes the result to #0F

```
#0F = FUP(D0F)
```

	#3	#2	#1	#0			D3	D2	D1	D0
#0F		34.0			←	D0F		33.54		

(2) Program which finds the rounded-up fractional portion value of D4F and substitutes the result to #0F (when D4F is a negative number)

```
#0F = FUP(D4F)
```

	#3	#2	#1	#0			D7	D6	D5	D4
#0F		-33.0			←	D4F		-33.54		

F/FS	G
○	○

5.6.14 BCD → BIN conversion : BIN

Format	BIN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	BCD data which will be converted into BIN data	Data type of (S) (Integer type)

[Functions]

(1) The BCD data specified with (S) is converted into BIN data.

(2) If (S) is a 16-bit integer type, the data range is 0 to 9999.

(3) If (S) is a 32-bit integer type, the data range is 0 to 99999999.

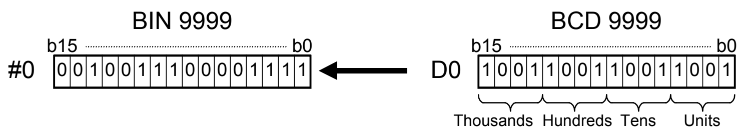
[Errors]

(1) An operation error will occur if:
- A value other than 0 to 9 is in any digit of (S); or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the BCD data of D0 into BIN data and substitutes the result to #0

```
#0 = BIN(D0)
```



F/FS	G
○	○

5.6.15 BIN → BCD conversion : BCD

Format	BCD(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	BIN data which will be converted into BCD data	Data type of (S) (Integer type)

[Functions]

(1) The BIN data specified with (S) is converted into BCD data.

(2) If (S) is a 16-bit integer type, the data range is 0 to 9999.

(3) If (S) is a 32-bit integer type, the data range is 0 to 99999999.

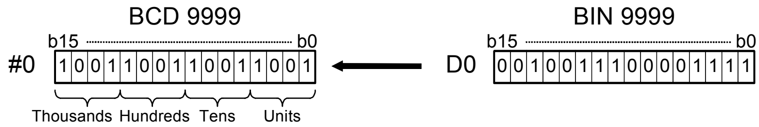
[Errors]

(1) An operation error will occur if:
- The data is other than 0 to 9999 when (S) is a 16-bit integer type;
- The data is other than 0 to 99999999 when (S) is a 32-bit integer type; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the BIN data of D0 into BCD data and substitutes the result to #0

```
#0 = BCD(D0)
```



F/FS	G
○	○

5.7 Type Conversions

5.7.1 Signed 16-bit integer value conversion : SHORT

Format	SHORT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 16-bit integer value	16-bit integer type

[Functions]

(1) The data specified with (S) is converted into a signed 16-bit integer value.

(2) The data range of (S) is -32768 to 32767.

(3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.

(4) If (S) is a 16-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range -32768 to 32767; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0L into a signed 16-bit integer value and substitutes the result to #0

```
#0 = SHORT(D0L)
```

```
                                      D1      D0
#0  | K-30000 |  <-----------   D0L | K-30000L     |
     (H8AD0)                         (HFFFF8AD0)
```

F/FS	G
○	○

5.7.2 Unsigned 16-bit integer value conversion : USHORT

Format	USHORT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 16-bit integer value	16-bit integer type

[Functions]

(1) The data specified with (S) is converted into an unsigned 16-bit integer value.

(2) The data range of (S) is 0 to 65535.

(3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.

(4) If (S) is a 16-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range 0 to 65535; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0L into an unsigned 16-bit integer value and substitutes the result to #0

```
#0 = USHORT(D0L)
```

```
                                      D1    D0
#0  [K-5536]  ←————————  D0L [  K60000L  ]
    (HEA60)                  (H0000EA60)
       ↑
Unsigned value is K60000
```

POINT

It is converted into a large data type to operate the binary operations with a different data type. Therefore, USHORT does not become valid.
The target binary operations are shown below.

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Remainder (%)
- Bit logical AND (&)
- Bit logical OR (|)
- Bit exclusive logical OR (^)

[Example] W0:F=#0F+USHORT(D0L)

USHORT does not become valid.

64-bit floating point type

F/FS	G
○	○

5.7.3 Signed 32-bit integer value conversion : LONG

Format	LONG(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 32-bit integer value	32-bit integer type

[Functions]

(1) The data specified with (S) is converted into a signed 32-bit integer value.

(2) The data range of (S) is -2147483648 to 2147483647.

(3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.

(4) If (S) is a 32-bit integer type, its value is returned unchanged, with no conversion processing performed.

[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range -2147483648 to 2147483647; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0 into a signed 32-bit integer value and substitutes the result to #0L

```
#0L = LONG(D0)
```

```
       #1      #0
#0L  [  K-1L      ]  <------------  D0 [ K-1 ]
      (HFFFFFFFF)                      (HFFFF)
```

F/FS	G
○	○

5.7.4 Unsigned 32-bit integer value conversion : ULONG

Format	ULONG(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 32-bit integer value	32-bit integer type

[Functions]

(1) The data specified with (S) is converted into an unsigned 32-bit integer value.

(2) The data range of (S) is 0 to 4294967295.

(3) When (S) is a 64-bit floating-point type, its fractional portion is rounded down before conversion is made.

(4) If (S) is a 32-bit integer type, its value is returned unchanged, with no conversion processing performed.

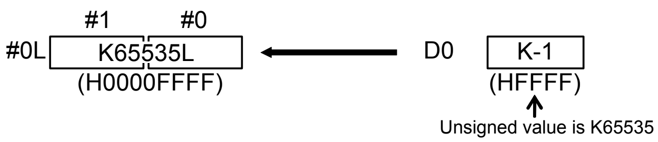
[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range 0 to 4294967295; or
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0 into an unsigned 32-bit integer value and substitutes the result to #0L

```
#0L = ULONG(D0)
```



POINT

It is converted into a large data type to operate the binary operations with a different data type. Therefore, ULONG does not become valid.
The target binary operations are shown below.

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Remainder (%)
- Bit logical AND (&)
- Bit logical OR (|)
- Bit exclusive logical OR (^)

[Example] W0:F=#0F+ULONG(D0L)

ULONG does not become valid.

64-bit floating point type

F/FS	G
○	○

5.7.5 Signed 64-bit floating-point value conversion : FLOAT

Format	FLOAT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into signed 64-bit floating-point value	64-bit floating-point type

[Functions]

(1) The data specified with (S) is converted into a signed 64-bit floating-point value.

(2) If (S) is a 64-bit floating-point type, its value is returned unchanged, with no conversion processing performed.

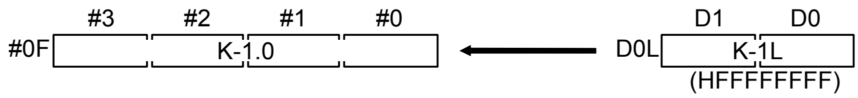
[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0L into a signed 64-bit floating-point value and substitutes the result to #0F

```
#0F = FLOAT(D0L)
```



F/FS	G
○	○

5.7.6 Unsigned 64-bit floating-point value conversion : UFLOAT

Format	UFLOAT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into unsigned 64-bit floating-point value	64-bit floating-point type

[Functions]

(1) The data specified with (S) is converted into an unsigned 64-bit floating-point value.

(2) If (S) is a 64-bit floating-point type, its value is returned unchanged, with no conversion processing performed.

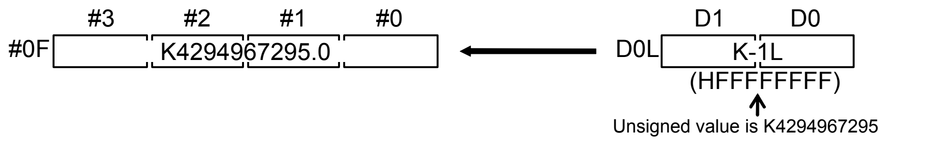
[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which converts the data of D0L into an unsigned 64-bit floating-point value and substitutes the result to #0F

```
#0F = UFLOAT(D0L)
```



F/FS	G
○	○

5.7.7 Floating-point value conversion 32-bit into 64-bit : DFLT Ver.!

Format	DFLT (S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data											
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	—	○ (Note-1)	—	—	—	—	—	—	—	—

○ : Usable

(Note-1) : The data is handled as the 32-bit integer type on the program, but store the 32-bit floating-point data in the device.

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into 64-bit floating-point value	64-bit floating-point type

[Functions]

(1) The 32-bit floating-point (single precision real number) value stored in the device specified with (S) is converted into the 64-bit floating-point (double precision real number) value.
Convertible data ranges are shown below.

-3.40×10^{38} to -1.18×10^{-38}, 0.0, 1.18×10^{-38} to 3.40×10^{38} (single precision real number)

(2) The 64-bit floating-point type is used as the data of floating-point type in the Motion SFC program. Use this instruction to input the data of 32-bit floating-point type from the external devices.

[Errors]

(1) An operation error will occur if:
- The (S) data is not a valid 32-bit floating-point type.

[Program examples]

(1) Program which converts the 32-bit floating-point value data of D2000L into 64-bit floating-point value data and substitutes the result to #0F

```
#0F = DFLT(D2000L)
```

```
      #3     #2     #1     #0                 D2001  D2000
#0F [      |    K-1.0     |      ]  <-----  D2000L [   K-1.0   ]
```

Ver.! : Refer to Section 1.3 for the software version that supports this function.

F/FS	G
○	○

5.7.8 Floating-point value conversion 64-bit into 32-bit : SFLT Ver.!

Format	SFLT(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	—	—	○	—	—	—	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be converted into 32-bit floating-point value	32-bit floating-point type

[Functions]

(1) The 64-bit floating-point (double precision real number) value stored in the device specified with (S) is converted into the 32-bit floating-point (single precision real number) value. Convertible data ranges are shown below.

-3.40×10^{38} to -1.18×10^{-38}, 0.0, 1.18×10^{-38} to 3.40×10^{38} (single precision real number)

(2) The 64-bit floating-point type is used as the data of floating-point type in the Motion SFC program. Use this instruction to output the data into the external devices that cannot use the 64-bit floating-point type.

POINT

The number of effective digits of 32-bit floating-point value data is approx. 7 digits. Data in the seven digits or later of conversion result by SFLT instruction may not match the (S) data.

[Errors]

(1) An operation error will occur if:
- The (S) data is not a valid 64-bit floating-point type.
- The (S) data after convert is outside the range of 32 bit floating-point type.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Program examples]

(1) Program which converts the 64-bit floating-point value data of #0F into 32-bit floating-point value data and substitutes the result to D2000L

```
D2000L = SFLT (#0F)
```

	D2001	D2000			#3	#2	#1	#0
D2000L	K-1.0		←	#0F		K-1.0		

F/FS	G
○	○

5.8 Bit Device Statuses

5.8.1 ON (Normally open contact) : (None)

Format	(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Bit device used in bit conditional expression	Logical type (true/false)

[Functions]

(1) True is returned when the bit device specified with (S) in a bit conditional expression is ON (1), or false is returned when that bit device is OFF (0).

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when either of M0 and X0 is ON (1)

```
SET M100 = M0 + X0
```

M0 $\boxed{0}$ (False)

M100 $\boxed{1}$ $\xleftarrow{\text{(True)}}$ +

X0 $\boxed{1}$ (True)

F/FS	G
○	○

5.8.2 OFF (Normally closed contact) : !

Format	!(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Bit device used in bit conditional expression	Logical type (true/false)

[Functions]

(1) True is returned when the bit device specified with (S) in a bit conditional expression is OFF (0), or false is returned when that bit device is ON (1).

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which resets M100 when M0 is OFF (0)

```
RST M100 = !M0
```

M100 $\boxed{0}$ ← !M0 $\boxed{0}$ (True)

F/FS	G
○	○

5.9 Bit Device Controls

5.9.1 Device set : SET

Format	SET(D)=(S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

(Note-1) : PX is write-disabled and cannot be used at (D).
(Note-2) : M2001 to M2032 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Bit data for device set	Bit logical type (true/false)
(S)	Condition data which determines whether device set will be performed or not	

[Functions]

(1) If the data specified with (S) is true, the bit data specified with (D) is set.

(2) (S) can be omitted.
At this time, the format is "SET(D)" and device set is made unconditionally.

(3) When this instruction is set as a transition condition in the last block of a transition program, whether the data specified with (S) is true or false is returned as logical type data. In this case, (S) cannot be omitted.

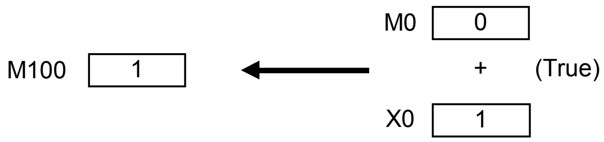
[Errors]

(1) An operation error will occur if:
- (D) or (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

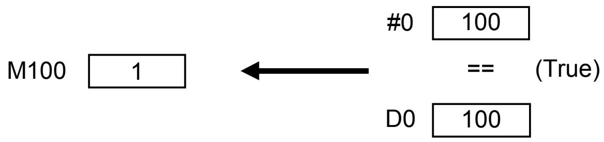
(1) Program which sets M100 when either of M0 and X0 is 1

```
SET M100 = M0 + X0
```



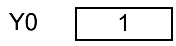
(2) Program which sets M100 when #0 is equal to D0

```
SET M100 = #0 = = D0
```



(3) Program which sets Y0 unconditionally

```
SET Y0
```



F/FS	G
○	○

5.9.2 Device reset : RST

Format	RST(D)=(S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

(Note-1) : PX is write-disabled and cannot be used at (D).
(Note-2) : M2001 to M2032 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Bit data for device reset	Bit logical type (true/false)
(S)	Condition data which determines whether device reset will be performed or not	

[Functions]

(1) If the data specified with (S) is true, the bit data specified with (D) is reset.

(2) (S) can be omitted.
At this time, the format is "RST(D)" and device reset is made unconditionally.

(3) When this instruction is set as a transition condition in the last block of a transition program, whether the data specified with (S) is true or false is returned as logical type data. In this case, (S) cannot be omitted.

[Errors]

(1) An operation error will occur if:
- (D) or (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which resets M100 when either of M0 and X0 is 1

```
RST M100 = M0 + X0
```

		M0	0
M100	0 ←	+	(True)
		X0	1

(2) Program which resets M100 when #0 is equal to D0

```
RST M100 = #0 != D0
```

		#0	100
M100	0 ←	!=	(True)
		D0	200

(3) Program which resets Y0 unconditionally

```
RST Y0
```

Y0 0

F/FS	G
○	○

5.9.3 Device output : DOUT

Format	DOUT(D), (S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	—	○	○	—	○	○	○	—	○	—	—

○ : Usable

(Note-1) : PX and special relay cannot be used at (D).
(Note-2) : Range including M2000 to M2127 cannot be used at (D).

[Setting data]

Setting data	Description	Data type of result
(D)	Output destination bit data	Batch bit
(S)	Output source data	

[Functions]

(1) The data specified with (S) is output to the bit data specified with (D).

(2) Specify a multiple of 16 as the device No. of the bit data specified with (D).

(3) If the type of (S) is a 16-bit integer type, 16 points of the (S) data, starting at the least significant bit, are output in order to the bit devices headed by the one specified with (D).

(4) If the type of (S) is a 32-bit integer type, 32 points of the (S) data, starting at the least significant bit, are output in order to the bit devices headed by the one specified with (D).

[Errors]

(1) An operation error will occur if:
- (D) or (S) is an indirectly specified device and its device No. is outside the range.
- (D) is an indirectly specified device and its device No. is not a multiple of 16.

[Program examples]

(1) Program which outputs the data of D0 to Y0-YF

```
DOUT Y0, D0
```

YF ··· Y0		D0	b15 ··· b0
0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1	←	D0	0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1

F/FS	G
○	○

5.9.4 Device input : DIN

Format	DIN(D), (S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	—	○	○	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Input destination data	Data type of (D) (Integer type)
(S)	Input source bit data	

[Functions]

(1) The bit data specified with (S) is input to the data specified with (D).

(2) Specify a multiple of 16 as the device No. of the bit data specified with (S).

(3) If the type of (D) is a 16-bit integer type, 16 points of the (D) data, starting at the least significant bit, are input in order to the bit devices headed by the one specified with (S).

(4) If the type of (D) is a 32-bit integer type, 32 points of the (D) data, starting at the least significant bit, are input in order to the bit devices headed by the one specified with (S).

[Errors]

(1) An operation error will occur if:
- (D) or (S) is an indirectly specified device and its device No. is outside the range.
- (S) is an indirectly specified device and its device No. is not a multiple of 16.

[Program examples]

(1) Program which inputs the data of X0-XF to D0

```
DIN D0, X0
```

	b15 ... b0		XF ... X0
D0	0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1	←	0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1

F/FS	G
○	○

5.9.5 Bit device output : OUT

Format	OUT(D)=(S)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Bit device for device output	Bit logical type (true/false)
(S)	Condition data which determines device output	

[Functions]

(1) If the data specified with (S) is true, the bit data specified with (D) is set, and if the data specified with (S) is false, the bit data specified with (D) is reset.

(2) When this instruction is set as a transition condition in the last block of a transient program, whether the data specified with (S) is true or false is returned as logical type data.

(3) In this case, (S) cannot be omitted.

[Errors]

(1) An operation error will occur if:
- (D) or (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when M0 is ON (1) and program which resets M100 when M0 is OFF (0)

```
OUT M100 = M0
```

(2) Program which sets M100 when M0 and M1 are both on and resets M100 except it

```
OUT M100 = M0 * M1
```

(3) Program which sets M100 when D0 is equal to D2000 and resets M100 when D is not equal to D2000

```
OUT M100 = (D0 == D2000)
```

F/FS	G
○	○

5.10 Logical Operations

5.10.1 Logical acknowledgement : (None)

Format	(S)

Number of basic steps	—

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be logically acknowledged	Logical type (true/false)

[Functions]

(1) Whether the logical type data specified with (S) is true or false is returned unchanged. (Logical acknowledgement)

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when either of M0 and X0 is ON (1)

```
SET M100 = M0 + X0
```

M0 0 (False)

M100 1 ← (True)

+

X0 1 (True)

F/FS	G
○	○

5.10.2 Logical negation : !

Format	! (S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Data which will be logically negated	Logical type (true/false)

[Functions]

(1) The data specified with (S) is logically negated.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when "either of M0 and X0 is not ON (1)" (when M0 and X0 are both OFF (0))

```
SET M100 = ! (M0 + X0)
```

M0 0 (False)

M100 1 ← (True) ! (False) +

X0 0 (False)

F/FS	G
○	○

5.10.3 Logical AND : *

Format	(S1) * (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	○	—	—	—	—	—	—	—	—	○	○
(S2)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ANDed	Logical type (true/false)
(S2)		

[Functions]

(1) The data specified with (S1) and the data specified with (S2) are ANDed.

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when M0 and X0 are both 1

```
SET M100 = M0 * X0
```

M0 [1]

M100 [1] ← * (True)

X0 [1]

F/FS	G
○	○

5.10.4 Logical OR : +

Format	(S1)+(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	○	—	—	—	—	—	—	—	—	○	○
(S2)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be ORed	Logical type (true/false)
(S2)		

[Functions]

(1) The data specified with (S1) and the data specified with (S2) are ORed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which sets M100 when either of M0 and X0 is 1

```
SET M100 = M0 + X0
```

M0 0

M100 1 + (True)

X0 1

F/FS	G
○	○

5.11 Comparison Operations

5.11.1 Equal to : ==

Format	(S1)==(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The data specified with (S1) and the data specified with (S2) are compared, and the result is true if they are equal.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 and D0 are equal or not

```
#0 == D0
```

#0 100

(True) ← ==

D0 100

F/FS	G
○	○

5.11.2 Not equal to : !=

Format	(S1)!=(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The data specified with (S1) and the data specified with (S2) are compared, and the result is true if they are not equal.

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 and D0 are unequal or not

```
#0 != D0
```

#0 100

(True) ← !=

D0 20

F/FS	G
○	○

5.11.3 Less than : <

Format	(S1)<(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The result is true if the data specified with (S1) is less than the data specified with (S2).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 is less than D0 or not

```
#0 < D0
```

#0 | 10 |

(True) ← <

D0 | 20 |

F/FS	G
○	○

5.11.4 Less than or equal to : <=

Format	(S1)<=(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The result is true if the data specified with (S1) is less than or equal to the data specified with (S2).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 is less than or equal to D0 or not

```
#0 <= D0
```

#0 | 10

(True) ← <=

D0 | 20

F/FS	G
○	○

5.11.5 More than : >

Format	(S1)>(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The result is true if the data specified with (S1) is greater than the data specified with (S2).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 is greater than D0 or not

```
#0 > D0
```

#0 400

(True) ← >

D0 20

F/FS	G
○	○

5.11.6 More than or equal to : >=

Format	(S1)>=(S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which will be compared	Logical type (true/false)
(S2)		

[Functions]

(1) The result is true if the data specified with (S1) is greater than or equal to the data specified with (S2).

(2) When (S1) and (S2) differ in data type, the data of the smaller data type is converted into that of the greater type before comparison is performed.

[Errors]

(1) An operation error will occur if:
- (S1) or (S2) is an indirectly specified device and its device No. is outside the range.

[Program examples]

(1) Program which compares whether #0 is greater than or equal to D0 or not

```
#0 >= D0
```

#0 | 400 |

(True) ← >=

D0 | 20 |

F/FS	G
○	○

5.12 Motion-Dedicated Functions

5.12.1 Speed change request : CHGV

Format	CHGV((S1), (S2))

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which speed change request will be given	—
(S2)	Specified speed	

[Functions]

(1) A speed change is made shown below.
  (a) The speed change accepting flag (M2061 to M2092) correspond to the axis specified with (S1) is turned ON.
  (b) The speed of the axis specified with (S1) is changed to the speed specified with (S2).
  (c) The speed change accepting flag is turned OFF.

(2) The axis No. that may be set at (S1) is within the following range.

Q173DSCPU	Q173DCPU(-S1)	Q172DSCPU	Q172DCPU(-S1)
1 to 32		1 to 16	1 to 8

(3) For interpolation control, set any one of the interpolation axes to (S1). When linear interpolation control is exercised, a speed change varies as described below with the positioning speed designation method set in the servo program.

Positioning speed designation method	Operation
Vector speed designation	Speed change is made so that the vector speed becomes the speed specified with (S2).
Longest axis designation	Speed change is made so that the longest axis speed becomes the speed specified with (S2).
Reference axis speed designation	Speed change is made so that the reference axis speed becomes the speed specified with (S2).

(4) Operation varies with the sign of the specified speed set at (S2).

Sign of specified speed	Operation
Positive	Speed change
0	Temporary stop
Negative	Return

(5) The specified speed that may be set at (S2) is within the following range.

(a) Real mode

	mm		inch		degree		pulse	
	Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
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 (Note-1)	0 to 2147483647	pulse/s
Return 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 (Note-1)	-1 to -2147483647	pulse/s

(Note-1): When the "speed control 10 $\times$ multiplier setting for degree axis" is set to "valid" in the fixed parameter, the unit is "$\times 10^{-2}$ [degree/min]".

(b) Virtual mode

	pulse	
	Setting range	Unit
Speed change request	0 to 2147483647	pulse/s
Return request	-1 to -2147483647	pulse/s

(6) The speed changed by CHGV instruction is effective only on the servo program during starting.

(7) The speed change does not executed for the axis specified with (S1) during deceleration stop.

(8) The speed change does not executed for the axis specified with (S1) during speed-torque control. QDS

(9) Acceleration/deceleration time at speed change can be changed by setting the acceleration/deceleration time change parameter of the axis specified with (S1). QDS Ver.!
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for acceleration/deceleration time change parameter and acceleration/deceleration time change function.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(10) By specifying a negative speed and making a speed change request during the start, allows the axis to start deceleration at that point and return in the opposite direction upon completion of deceleration.
The following operations by the servo instruction are shown below.

Control mode	Servo instruction	Operation
Linear control	ABS−1 INC−1 ABS−2 INC−2 ABS−3 INC−3 ABS−4 INC−4	On completion of deceleration, the axis reverses its moving direction, returns to the positioning starting point at the absolute value of the specified speed, and stops (waits) there. For circular interpolation, the axis returns in the circular path.
Circular interpolation control	ABS circular INC circular	
Fixed-pitch feed	FEED−1 FEED−2 FEED−3	
Constant-speed control	CPSTART1 CPSTART2 CPSTART3 CPSTART4	On completion of deceleration, the axis reverses its moving direction, returns to the preceding point at the absolute value of the specified speed, and stops (waits) there.
Speed control (Ⅰ)	VF VR	On completion of deceleration, the axis reverses its moving direction at the absolute value of the specified speed. The axis does not stop until a stop instruction is input.
Speed control (Ⅱ)	VVF VVR	
Speed-position switching control	VPF VPR VPSTART	The axis cannot return. The speed change request is regarded as a normal speed change request. Minor error [305] (Note) will occur and the axis will be controlled at the speed limit value.
Position follow-up control	PFSTART	
Speed control with fixed position stop	PVF PVR	
Speed switching control	VSTART	
JOG operation		
High-speed oscillation	OSC	A speed change cannot be made. Minor error [310] (Note) will occur.
Home position return	ZERO	A speed change cannot be made. Minor error [301] (Note) will occur.

(Note) : Minor error [301] : A speed change was made during home position return.
Minor error [305] : The setting speed is outside the range of 0 to speed limit value.
Minor error [310] : A speed change was made during high-speed oscillation.

[Controls]

(a) If a speed change is made to a negative speed, control is executed with the control mode during the start as indicated in the above table.

(b) The returning command speed is the absolute value of a new speed.

(c) When the axis is waiting at the return position
  1) Signal states
    • Start accept flag (M2001+n) .... ON (unchanged from before execution of CHGV instruction)
    • Positioning start complete (M2400+20n) .... ON (unchanged from before execution of CHGV instruction)
    • Positioning complete (M2401+20n)....................... OFF
    • In-position (M2402+20n).. ON
    • Command in-position (M2403+20n)...................... OFF
    • Speed change "0" accepting flag (M2240+n) .... ON
  2) Make a speed change to a positive speed for a restart.
  3) Turn on the stop command to end the positioning.
  4) A negative speed change made again will be ignored.

(d) While the axis is reversion in the speed control mode
  1) Make a speed change to a positive speed to change the travel direction again.
  2) Turn ON the stop command to make a stop.
  3) A speed change is made in the opposite direction if a negative speed change is made again.

(e) A speed change to a negative speed will not be made for the axis which set the stroke limit as invalid.

[Errors]

(1) An operation error will occur and a speed change will not be made if:
  • The specified axis No. of (S1) is outside the range.
  • (S2) is an indirectly specified device and its device No. is outside the range.

(2) A minor error will occur and a speed change will not be made if:
  • The axis specified with (S1) is home position return. (Minor error: 301)
  • A speed change to a negative speed was made for the axis which set the stroke limit as invalid. (Minor error: 310)

POINT

If the speed change is executed for the axis specified with (S1) during deceleration, the speed change is ignored. An error will not occur in this case.

(3) A minor error will occur and the axis to be controlled at the speed limit value if:
- The absolute value of the speed specified with (S2) is greater than the speed limit value. (Minor error: 305)

POINT
If the absolute value of a negative new speed is higher than the speed specified with the servo program during constant-speed control, return control is exercised at the speed specified in the program (speed clamp control for a speed change during constant-speed control). At this time, an error will not occur.

[Program examples]

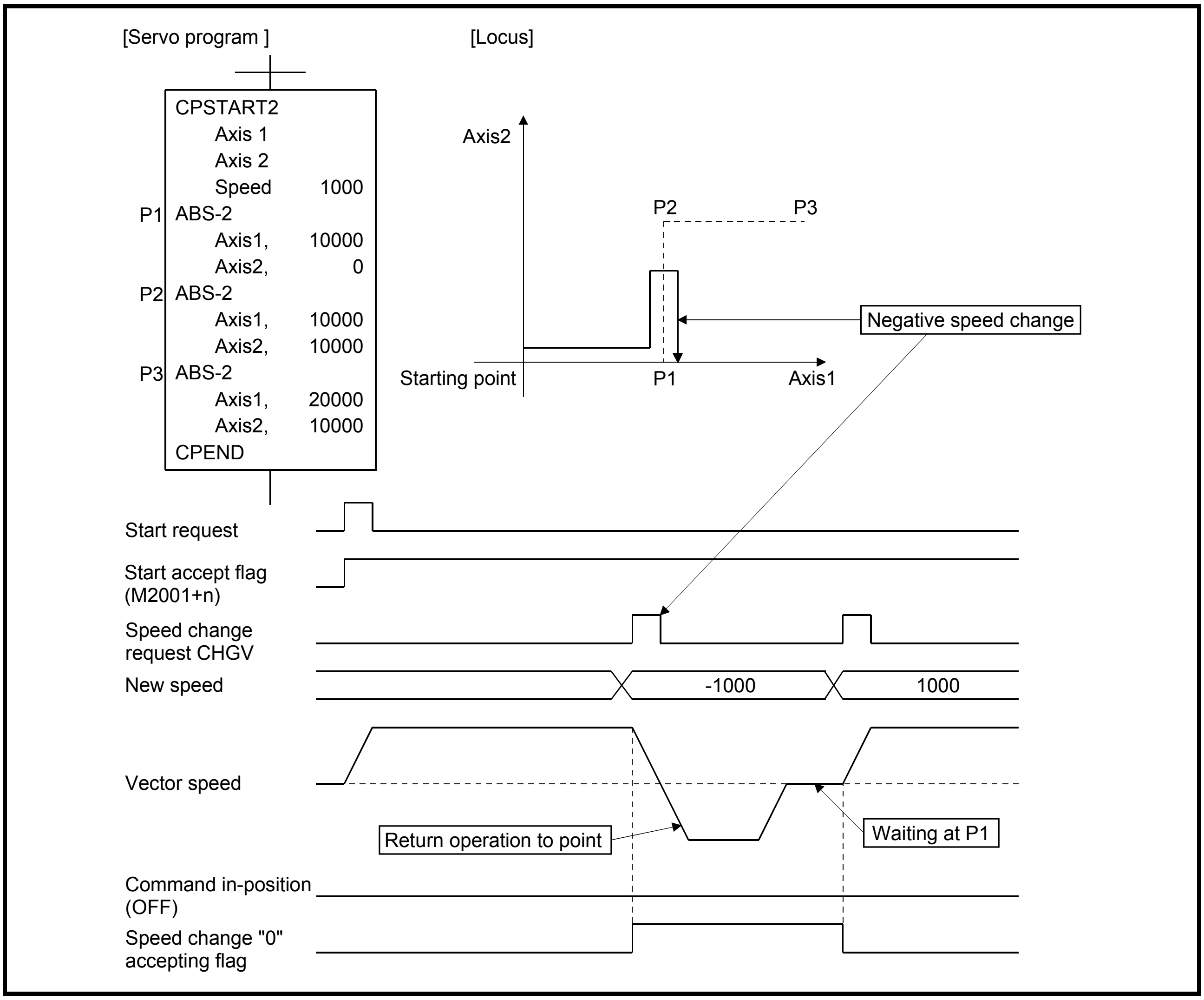
(1) Program which changes the positioning speed of axis 2

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

(2) Return program which changes the positioning speed of axis 1 to a negative value

```
CHGV(K1,K－1000)
```

The following operation will be performed when a return request is made in constant-speed control.



If a speed change to a negative speed is made during execution of positioning to P2 as shown above, the axis returns to P1 along the program specified locus and waits at P1.

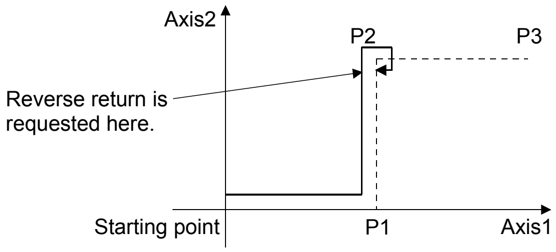
POINT

- Precautions at speed change
  (1) A speed change may be invalid if the speed change is executed until the "positioning start complete signal" status changes to ON at servo program start request . When making a speed change at almost the same timing as a start, create a program to execute speed change after the "positioning start complete signal" has turned on.

  (2) When the reverse return is requested during stop in the state of FIN waiting using the M-code FIN signal wait function in constant-speed control, it will be ignored.

  (3) In the example of previous page, if reverse return is requested before P2 and the axis passes through P2 during deceleration, it return to P2.

  (4) There will be a delay of time equivalent to an operation cycle at the maximum in the response time from when the CHGV instruction is executed until the speed begins to change actually.



F/FS	G
○	○

5.12.2 Command generation axis speed change request : CHGVS (SV22 advanced synchronous control only) QDS Ver.!

Format	CHGVS((S1), (S2))

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which speed change request will be given	—
(S2)	Specified speed	

[Functions]

(1) A speed change is made shown below.
   (a) The [St.346] Command generation axis speed change accepting flag (M9811+20n) correspond to the axis specified with (S1) is turned ON.
   (b) The speed of the axis specified with (S1) is changed to the speed specified with (S2).
   (c) The speed change accepting flag is turned OFF.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(2) The axis No. that may be set at (S1) is within the following range.

Q173DSCPU	Q172DSCPU
1 to 32	1 to 16

(3) For interpolation control, set any one of the interpolation axes to (S1). When linear interpolation control is exercised, a speed change varies as described below with the positioning speed designation method set in the servo program.

Positioning speed designation method	Operation
Vector speed designation	Speed change is made so that the vector speed becomes the speed specified with (S2).
Longest axis designation	Speed change is made so that the longest axis speed becomes the speed specified with (S2).
Reference axis speed designation	Speed change is made so that the reference axis speed becomes the speed specified with (S2).

(4) Operation varies with the sign of the specified speed set at (S2).

Sign of specified speed	Operation
Positive	Speed change
0	Temporary stop
Negative	Return

(5) The specified speed that may be set at (S2) is within the following range.

	mm		inch		degree		pulse	
	Setting range	Unit	Setting range	Unit	Setting range	Unit	Setting range	Unit
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 (Note-1)	0 to 2147483647	pulse/s
Return 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 (Note-1)	-1 to -2147483647	pulse/s

(Note-1): When the "speed control 10 × multiplier setting for degree axis" is set to "valid" in the command generation axis parameter, the unit is "$\times 10^{-2}$ [degree/min]".

(6) The speed changed by CHGVS instruction is effective only on the servo program during starting.

(7) The speed change does not executed for the axis specified with (S1) during deceleration stop.

(8) Acceleration/deceleration time at speed change can be changed by setting the acceleration/deceleration time change parameter of the axis specified by (S1).
Refer to the "Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)" for acceleration/deceleration time change parameter.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for acceleration/deceleration time change function.

(9) By specifying a negative speed and making a speed change request during the start, allows the axis to start deceleration at that point and return in the opposite direction upon completion of deceleration.
The following operations by the servo instruction are shown below.

Control mode	Servo instruction	Operation
Linear control	ABS-1 INC-1 ABS-2 INC-2 ABS-3 INC-3 ABS-4 INC-4	On completion of deceleration, the axis reverses its moving direction, returns to the positioning starting point at the absolute value of the specified speed, and stops (waits) there. For circular interpolation, the axis returns in the circular path.
Circular interpolation control	ABS circular INC circular	
Fixed-pitch feed	FEED-1 FEED-2 FEED-3	
Constant-speed control	CPSTART1 CPSTART2 CPSTART3 CPSTART4	On completion of deceleration, the axis reverses its moving direction, returns to the preceding point at the absolute value of the specified speed, and stops (waits) there.
Speed control (I)	VF VR	On completion of deceleration, the axis reverses its moving direction at the absolute value of the specified speed. The axis does not stop until a stop instruction is input.
Position follow-up control	PFSTART	The axis cannot return. The speed change request is regarded as a normal speed change request. Minor error [305] (Note) will occur and the axis will be controlled at the speed limit value.
Speed control with fixed position stop	PVF PVR	
JOG operation		

(Note): Minor error [305] : The setting speed is outside the range of 0 to speed limit value.

[Controls]

(a) If a speed change is made to a negative speed, control is executed with the control mode during the start as indicated in the above table.

(b) The returning command speed is the absolute value of a new speed.

(c) When the axis is waiting at the return position
1) Signal states
- [St.345] Command generation axis start accept flag (M9810+20n) .... ON (unchanged from before execution of CHGVS instruction)
- [St.340] Command generation axis positioning start complete (M9800+20n) .... ON (unchanged from before execution of CHGVS instruction)
- [St.341] Command generation axis positioning complete (M9801+20n) .... OFF
- [St.342] Command generation axis command in-position (M9803+20n) .... OFF
- [St.347] Command generation axis speed change "0" accepting flag (M9812+20n).... ON

2) Make a speed change to a positive speed for a restart.
3) Turn on the stop command to end the positioning.
4) A negative speed change made again will be ignored.

(d) While the axis is reversion in the speed control mode
1) Make a speed change to a positive speed to change the travel direction again.
2) Turn ON the stop command to make a stop.
3) A speed change is made in the opposite direction if a negative speed change is made again.

(e) A speed change to a negative speed will not be made for the axis which set the stroke limit as invalid.

[Errors]

(1) An operation error will occur and a speed change will not be made if:
- The specified axis No. of (S1) is outside the range.
- (S2) is an indirectly specified device and its device No. is outside the range.

(2) A minor error will occur and a speed change will not be made if:
- A speed change to a negative speed was made for the axis which set the stroke limit as invalid. (Minor error: 310)

POINT

If the speed change is executed for the axis specified with (S1) during deceleration, the speed change is ignored. An error will not occur in this case.

(3) A minor error will occur and the axis to be controlled at the speed limit value if:
- The absolute value of the speed specified with (S2) is greater than the speed limit value. (Minor error: 305)

POINT
If the absolute value of a negative new speed is higher than the speed specified with the servo program during constant-speed control, return control is exercised at the speed specified in the program (speed clamp control for a speed change during constant-speed control). At this time, an error will not occur.

[Program examples]

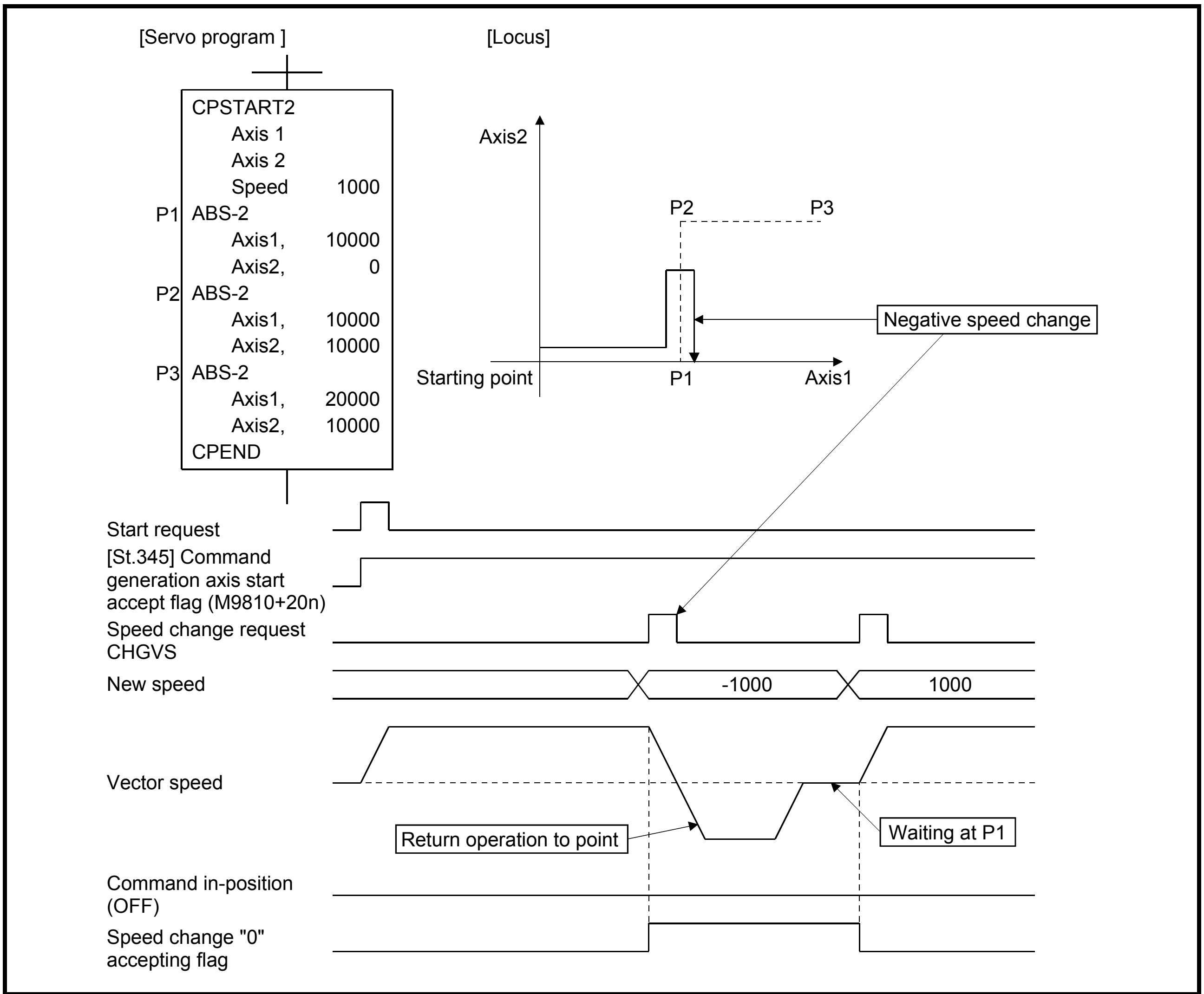
(1) Program which changes the positioning speed of axis 2

```
CHGVS(K2,K10)
```

(2) Return program which changes the positioning speed of axis 1 to a negative value

```
CHGVS(K1,K－1000)
```

The following operation will be performed when a return request is made in constant-speed control.



If a speed change to a negative speed is made during execution of positioning to P2 as shown above, the axis returns to P1 along the program specified locus and waits at P1.

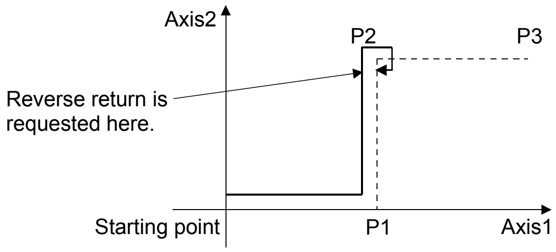
POINT

- Precautions at speed change
  (1) A speed change may be invalid if the speed change is executed until the "positioning start complete signal" status changes to ON at servo program start request . When making a speed change at almost the same timing as a start, create a program to execute speed change after the "positioning start complete signal" has turned on.

  (2) When the reverse return is requested during stop in the state of FIN waiting using the M-code FIN signal wait function in constant-speed control, it will be ignored.

  (3) In the example of previous page, if reverse return is requested before P2 and the axis passes through P2 during deceleration, it return to P2.

  (4) There will be a delay of time equivalent to an operation cycle at the maximum in the response time from when the CHGVS instruction is executed until the speed begins to change actually.



F/FS	G
○	○

5.12.3 Torque limit value change request : CHGT

Format	CHGT((S1), (S2))

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which torque limit value change request will be given	—
(S2)	Specified torque limit value	

[Functions]

(1) The torque limit value of the axis specified with (S1) is changed to the torque limit value specified with (S2) for the positive direction and negative direction.

(2) In the real mode, any axis that has completed a servo startup can be changed in torque limit value any time, independently of the status, starting, stopping, servo ON or servo OFF.

(3) The axis No. that may be set at (S1) is within the following range.

Q173DSCPU	Q173DCPU(-S1)	Q172DSCPU	Q172DCPU(-S1)
1 to 32		1 to 16	1 to 8

(4) The torque limit value that may be set at (S2) is within the range 1 to 1000[%].

(5) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for relation between torque limit value specified with servo program and torque limit value change instruction.

[Errors]

(1) An operation error will occur and a torque limit value change will not be made if:
- The specified axis No. at (S1) is outside the range; or
- (S2) is an indirectly specified device and its device No. is outside the range.

(2) A minor error will occur and a torque limit value change will not be made if:
- The torque limit value specified with (S2) is outside the range 1 to 1000[%] (Minor error: 311); or
- The CHGT instruction is executed for any axis that has not yet been started (Minor error: 312).

[Program examples]

(1) Program which changes the torque limit value of axis 2 to 10[%]

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

POINT

(1) CHGT instruction is invalid (ignored) during the virtual mode. When changing the torque limit value during operation in the virtual mode, set the "torque limit value setting device" in the output module parameter of the mechanical system program.
(2) There will be a delay of time equivalent to an operation cycle at the maximum in the time from when the CHGT instruction is executed until the torque limit value is transferred to servo amplifier actually.

F/FS	G
○	○

5.12.4 Torque limit value individual change request : CHGT2

Format	CHGT2((S1), (S2), (S3))

Number of basic steps	5

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—
(S3)	—	○	○	—	—	○	○	—	○	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Axis No. to which torque limit value change request will be given	—
(S2)	Positive direction torque limit value (×0.1[%])	
(S3)	Negative direction torque limit value (×0.1[%])	

[Functions]

(1) The torque limit value of the axis specified with (S1) is changed to the positive direction torque limit value specified with (S2) and negative direction torque limit value specified with (S3).
The positive direction torque limit value restricts the forward rotation (CCW) driving torque and reverse rotation (CW) regenerative torque of the servo motor, and negative direction torque limit value restricts the reverse rotation (CW) driving torque and forward rotation (CCW) regenerative torque of the servo motor.

(2) Any axis that has completed a servo startup can be changed in torque limit value any time, independently of the status, starting, stopping, servo ON or servo OFF.

(3) When the CHGT2 instruction is executed to the mechanical system output module in the virtual mode, set 300[%] as the torque limit value of output module. When the torque limit value of output module is indirectly specified with a device, a minor error (error code: 6260) will occur, and the individual change of torque limit value is not executed.

(4) The axis No. that may be set at (S1) is within the following range.

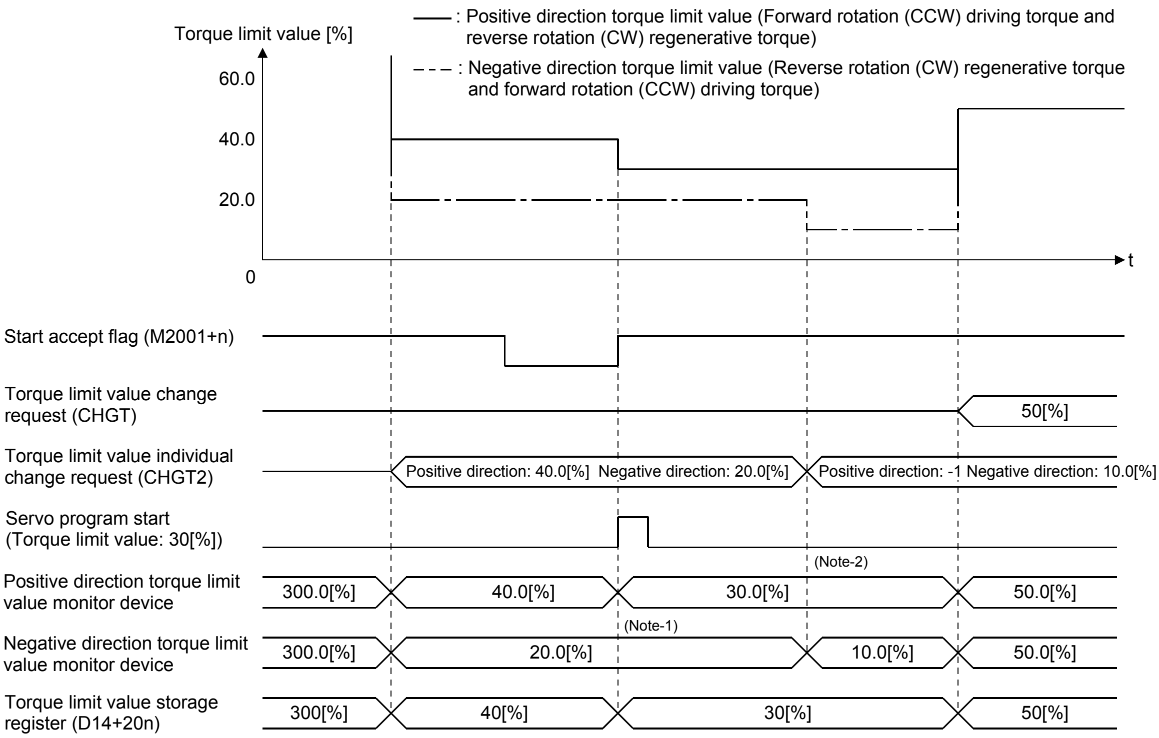
Q173DSCPU	Q172DSCPU
1 to 32	1 to 16

(5) (S2) and (S3) cannot be omitted. When only either torque limit value is changed, set "-1" as the setting data not to change.

(6) The torque limit value that may be set at (S2) and (S3) is within the range 1 to 10000 (×0.1[%]).

(7) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for relation between torque limit value specified with servo program and torque limit value change request instruction.

Operation for combining of CHGT2 and CHGT instruction is shown below.



(Note-1): The torque limit value specified with servo program is cramped with the negative direction torque limit value changed by CHGT2.
(Note-2): The torque limit value is not changed so that "-1" is set as the positive direction torque limit value of CHGT2.

(8) During speed-torque control, do not change the torque limit value to higher value than torque limit value in speed-torque control set in the speed-torque control data of servo data setting. If the either value of (S2) or (S3) specified with CHGT2 instruction is higher than torque limit value in speed-torque control, a minor error (error code: 319) will occur, and the individual change of torque limit value is not executed.

(9) The positive direction torque limit value and negative direction torque limit value can be monitored by setting the positive direction torque limit value monitor device and negative direction torque limit value monitor device in the expansion parameter of servo data setting.

[Errors]

(1) An operation error will occur and a torque limit value change will not be made if:
- The specified axis No. at (S1) is outside the range; or
- (S2) or (S3) is an indirectly specified device and its device No. is outside the range.

(2) A minor error will occur and a torque limit value change will not be made if:
- The torque limit value specified with (S2) or (S3) is outside the range of 0.1 to 1000.0[%] (Minor error: 311); or
- The CHGT2 instruction is executed for any axis that has not yet been started (Minor error: 312); or
- When the CHGT2 instruction is executed for any axis during speed-torque control, the value of (S2) or (S3) is greater than the torque limit value in speed-torque control (Minor error: 319); or
- In the virtual mode, the CHGT2 instruction is executed for any axis that the torque limit value of output module is indirectly specified with a device (Minor error: 6260).

[Program examples]

(1) Program which changes the torque limit value of axis 2 to positive direction 20.0[%] and to negative direction 10.0[%]

```
CHGT2(K2, K200, K100)
```

POINT

There will be a delay of time equivalent to an operation cycle at the maximum in the time from when the CHGT instruction is executed until the torque limit value is transferred to servo amplifier actually.

F/FS	G
○	○

5.12.5 Target position change request : CHGP

Format	CHGP((S1), (S2), (S3))

Number of basic steps	6

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(S3)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

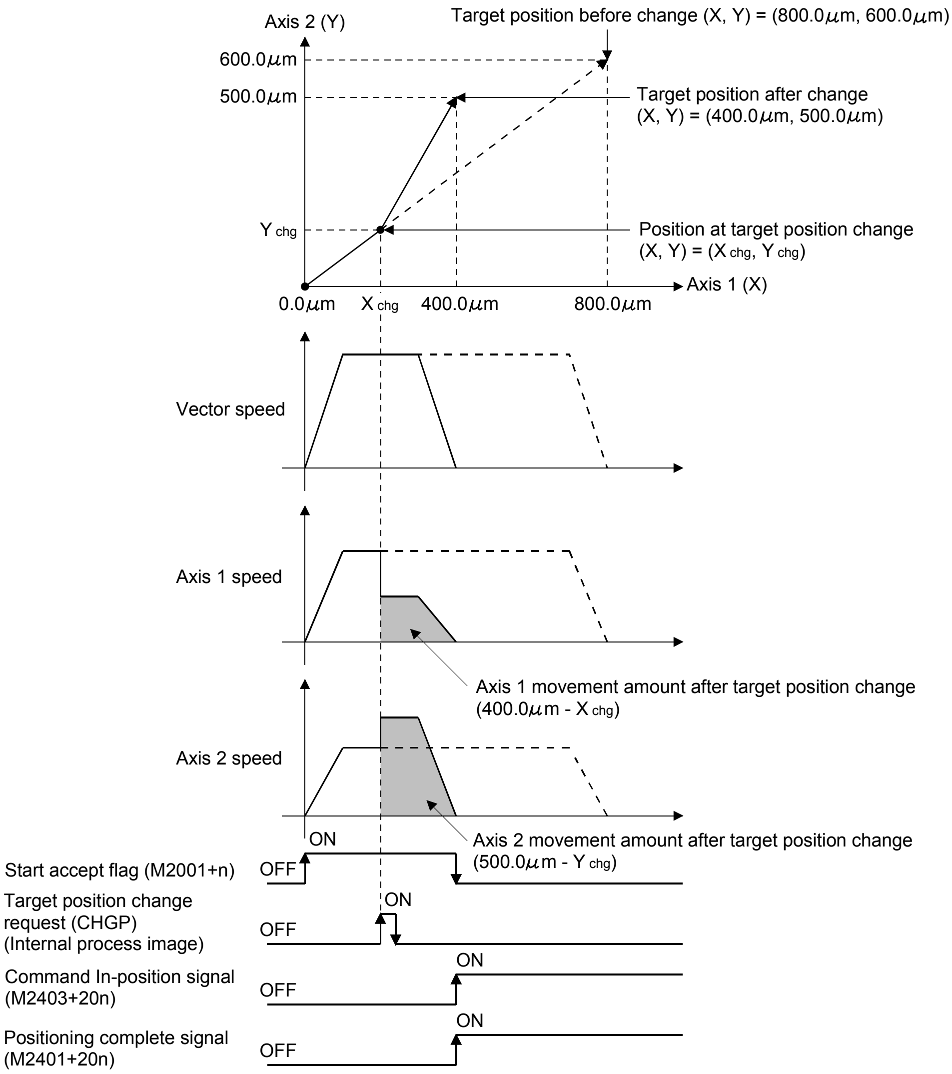
Setting data	Description	Data type of result
(S1)	Axis No. to which target position change request will be given	—
(S2)	Specified method of changed address 0: Address method 1: Movement method	
(S3)	Starting No. of device to store target position change value	

POINT

The CHGP instruction cannot be used for the command generation axis under advanced synchronous control.

[Overview]

The target position is changed during positioning instruction execution by target position change request. New target position can be set by the absolute address or relative movement amount from feed current value at target position change request. Operation for executing target position change request to (X, Y) = (400.0μm, 500.0μm) by absolute address setting during linear interpolation control from positioning start position (X, Y) = (0.0μm, 0.0μm) to (X, Y) = (800.0μm, 600.0μm) is shown below.



[Functions]

(1) The target position of the axis specified with (S1) is changed. The new target position is calculated by a value stored in the device specified with (S3) using the method specified with (S2).

POINT

(1) CHGP instruction is enabled to only starting axis.
(2) The target position is not changed when the specified axis is during deceleration stop.
(3) There will be a delay of time equivalent to an operation cycle at the maximum from when the CHGP instruction is executed until the target position is changed actually.
(4) When the CHGP instruction is executed at servo program start request (positioning start complete signal (M2400+20n) is OFF), the target position change becomes disabled. Create the program to execute the target position change after positioning start complete signal ON to change the target position at same timing with servo program start.

(2) The axis No. that may be set at (S1) is within the following range.
For interpolation control, set any one of the interpolation axes to (S1).

Q173DSCPU	Q172DSCPU
1 to 32	1 to 16

(3) The target position by setting of (S2) is shown below.
  (a) When "0" (address method) is set to (S2), the target position is the target position change value stored in the device specified with (S3).
  (b) When "1" (movement method) is set to (S2), the target position is the position that the movement for target position change value stored in the device specified with (S3) is executed from the feed current value at CHGP instruction execution.

POINT

When "1" (movement method) is set to (S2) and the CHGP instruction is executed with a normal task, a dispersion may occur for new target position depending on a dispersion of instruction accept timing. Execute the CHGP instruction in the fixed cycle task same as operation cycle to inhibit a dispersion.

(4) Set the starting device No. to store a target position change value at (S3).
Set an even number as first device, and set a target position change value as follows.

Offset	Name	Setting range				
		mm	inch	pulse	degree	
					Address method	Movement method
+0	Target position change value 1	-2147483648 to 2147483647 ($\times10^{-1}$[μm])	-2147483648 to 2147483647 ($\times10^{-5}$[inch])	-2147483648 to 2147483647 ([pulse])	0 to 35999999 ($\times10^{-5}$[degree])	-2147483648 to 2147483647 ($\times10^{-5}$[degree])
+1						
+2	Target position change value 2					
+3						
+4	Target position change value 3					
+5						
+6	Target position change value 4					
+7						

(a) Set the positioning address and movement amount according to the setting of (S2) for target position change value.

(b) Set the axis No. among interpolation axes in ascending order for target position change value.
(Example) When the target position change request is executed during INC-3 instruction execution.

```
[K100]  INC-3
          Axis     3,   3000pulse
          Axis     4,   4000pulse
          Axis     1,   4000pulse
          Speed         10000pulse/s
```

The axis No. for target position change value 1 to 4 are as follows.

Target position change value 1	Setting of axis No.1
Target position change value 2	Setting of axis No.3
Target position change value 3	Setting of axis No.4
Target position change value 4	Not necessary to set

(5) The following operations by the servo instruction at CHGP instruction execution are shown below.

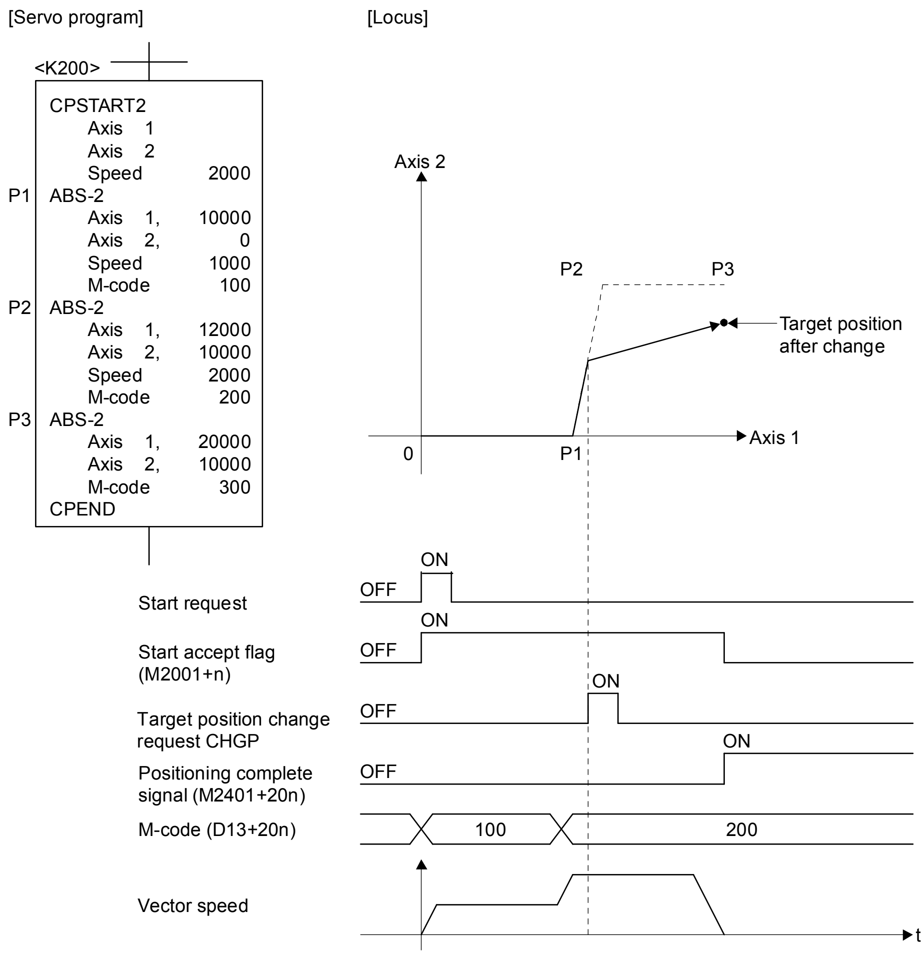
Control mode	Servo instruction	Operation
Linear control	ABS−1 INC−1 ABS−2 INC−2 ABS−3 INC−3 ABS−4 INC−4	The positioning is executed from current feed value during execution to new target position with linear interpolation control.
Fixed-pitch feed	FEED−1 FEED−2 FEED−3	
Circular interpolation control	ABS circular INC circular	The target position change is ignored, and a minor error [330] will occur.
Helical interpolation control	ABS helical INC helical	
Constant-speed control	CPSTART1 CPSTART2 CPSTART3 CPSTART4	The positioning is executed from current feed value during execution to new target position with linear interpolation control. The positioning to a remaining point is not executed. (Refer to this section (9).)
Speed control (Ⅰ)	VF VR	The target position change is ignored, and a minor error [330] will occur.
Speed control (Ⅱ)	VVF VVR	
Speed-position switching control	VPF VPR VPSTART	
Position follow-up control	PFSTART	
Speed control with fixed position stop	PVF PVR	
Speed switching control	VSTART	
JOG operation		
Speed-torque control		
High-speed oscillation	OSC	
Home position return	ZERO	

(6) Operation after execution of CHGP instruction are as follows.
- Automatic decelerating flag (M2128+n) turns ON with automatic deceleration processing to new target position.
- Command in-position signal (M2403+20n) turns ON when the absolute value of difference between new target position and current feed value becomes "command in-position range" or less.
- Positioning complete signal (M2401+20n) turns ON with command output completion to new target position.

(7) After execution of CHGP instruction, the vector speed does not change, but each axis speed changes according to new target position. Therefore, each axis speed may change rapidly depending on new target position.

(8) When the reference axis speed designation or longest axis reference designation is set in the linear interpolation control, an operation is as follows.
- The longest axis is not selected again at target position change. The longest axis before change is used continuously.
- The positioning speed is calculated depending on the movement amount for each axis new target position.
- When the movement amount of reference axis or longest axis depending on the target position change becomes 0, a minor error (error code: 264) will occur and deceleration stop is executed.

(9) The positioning is executed to new target position with CHGP instruction during constant speed control. The positioning to a point since executing point at target position change request is not executed.

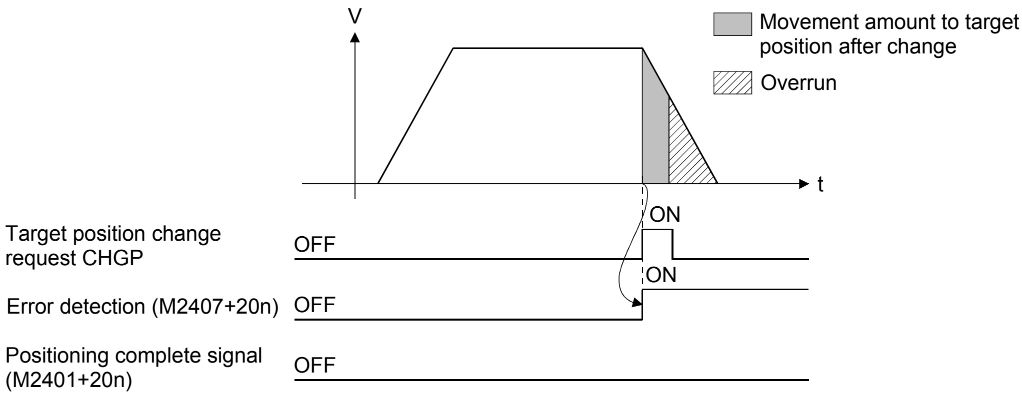


POINT

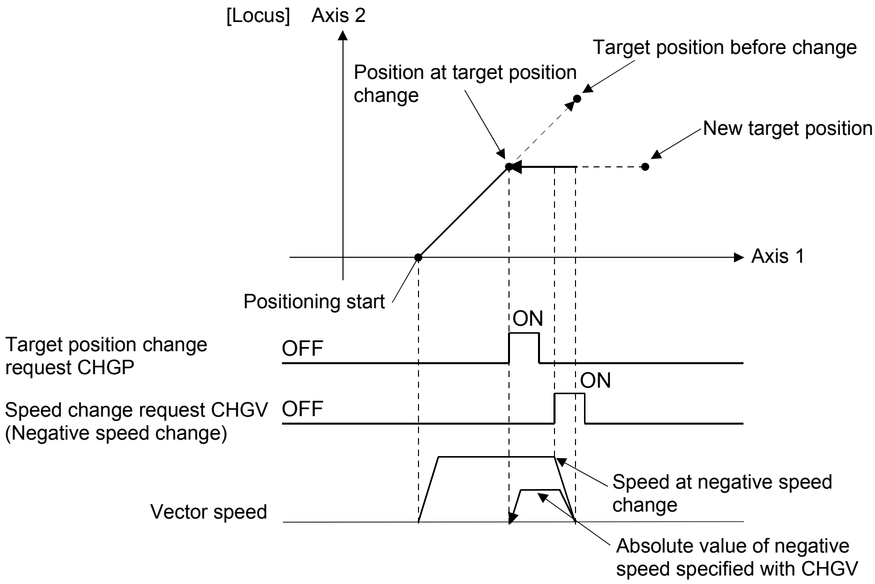
(1) The positioning is executed with the setting items of executing point for CHGP instruction.
(2) The linear interpolation control is executed in all axes specified with CPSTART for CHGP instruction. Set target position for all axes specified with CPSTART.
(3) When the CHGP instruction is executed during positioning to a point of circular interpolation or helical interpolation in the constant speed control, the target position change is executed at the same time of positioning to a point of linear interpolation.

(10) When the target position change of address method to the axis of control unit [degree] is executed, operation is as follows.
- The positioning to new address is executed in the current direction.
- Set "0 to 35999999×10^{-5} [degree] as new address at the address method. If the outside of range is set, a minor error (error code: 260) will occur and deceleration stop is executed.

(11) Operation for the movement amount to new target position is less than deceleration distance required to deceleration stop from speed during control by execution of CHGP instruction is as follows.
- A minor error (error code: 261) will occur and deceleration stop is executed at execution of CHGP instruction.
- The difference between movement amount to the deceleration stop and movement amount to new target position is overrun.
- The positioning complete signal (M2401+20n) does not turn ON.



(12) When the negative speed change is executed after execution of CHGP instruction, a deceleration is executed to speed 0. Then, it returns to position at target position change (CHGP instruction accept), and stops (waits) there.



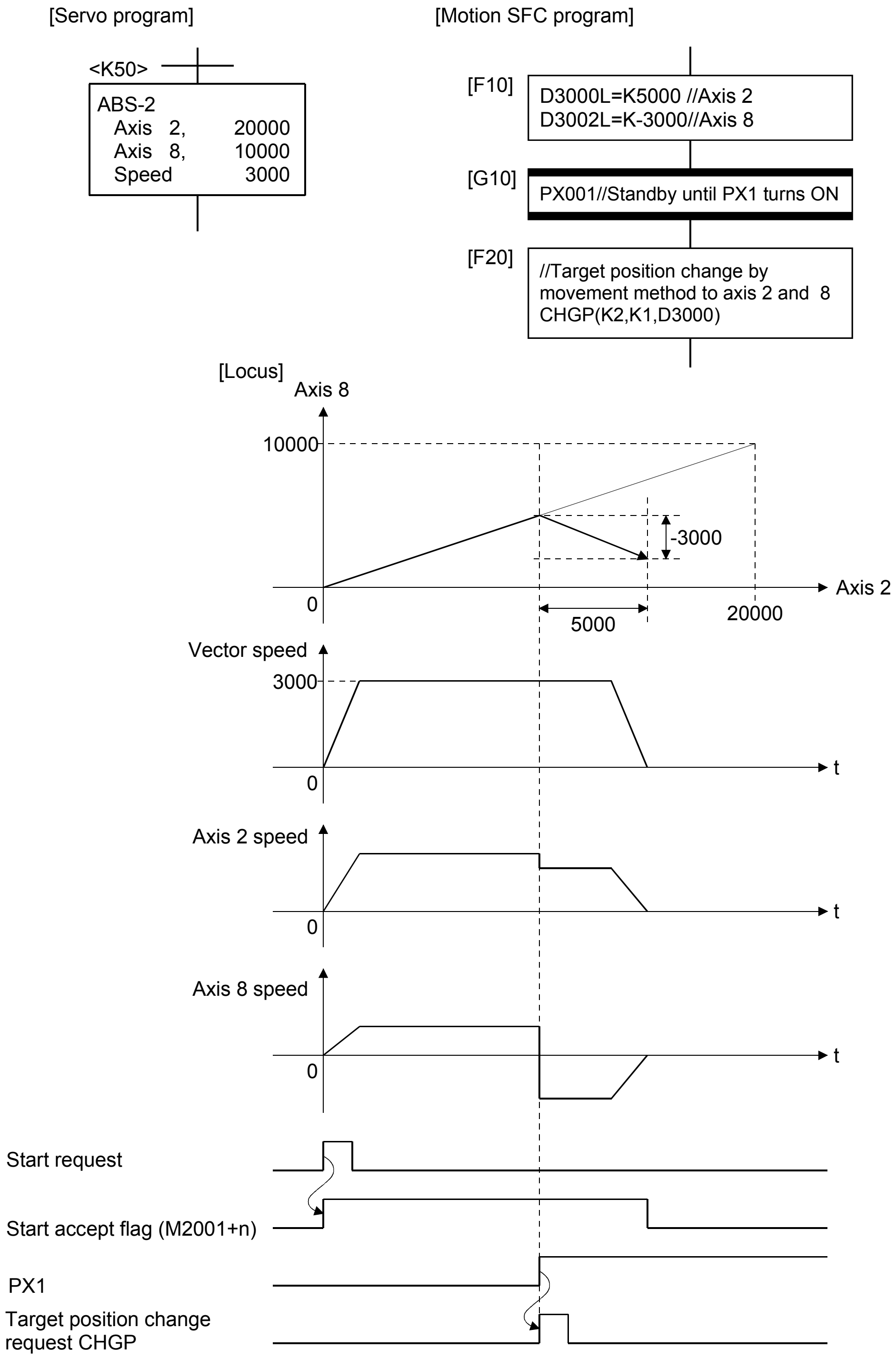
[Errors]

(1) An operation error will occur and a target position change will not be made if:
- The specified axis No. at (S1) is outside the range; or
- Except 0 to 1 is set at (S2); or
- Except even-numbered device is set at (S3); or
- The device No. (S3) to (S3) +7 is outside the range.

(2) A minor error will occur and a target position change will not be made if:
- During home position return for target axis (Minor error: 330); or
- During execution of servo program that does not correspond with the target position change for target axis (Minor error: 330); or
- New target position exceeded the stroke limit range (Minor error: 262); or
- FIN acceleration/deceleration or advanced S-curve acceleration/deceleration as acceleration/deceleration method is set (Minor error: 263); or
- When the reference axis speed designation or longest axis designation is set in the linear interpolation control, the movement amount for reference axis or longest axis after target position change becomes 0 (Minor error: 264); or
- When the target position change of address method to the axis of control unit [degree] is executed, new address is outside the range of 0 to 35999999×10^{-5} [degree] (Minor error: 260); or
- The movement amount to new target position is less than deceleration distance required to deceleration stop from speed during control (Minor error: 261).

[Program examples]

(1) Program which executes the target position change by movement method to axis 2 and axis 8 during positioning by ABS-2.



F/FS	G
○	○

5.13 Other Instructions

5.13.1 Event task enable : EI

Format	EI

Number of basic steps	1

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

(1) The execution of an event task is enabled.

(2) This instruction is usable with a normal task only.

[Errors]

(1) An operation error will occur if:
- This instruction is used with other than a normal task.

[Program examples]

(1) Enables the execution of an event task.

```
EI
```

F/FS	G
○	○

5.13.2 Event task disable : DI

Format	DI	Number of basic steps	1

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

(1) The execution of an event task is disabled.

(2) If an external interrupt or PLC interrupt occurs after execution of the DI instruction, the corresponding event task is executed once at the execution of the EI instruction. (If two or more external interrupts or PLC interrupts occur during DI, the corresponding event task is executed only once at the execution of the EI instruction.)

(3) During DI, a fixed-cycle event task is not executed.

(4) The execution of an NMI task cannot be disabled.

(5) The DI status is established at power-on or reset of the Multiple CPU system. EI/DI status does not change by the ON/OFF of PLC ready flag (M2000).

[Errors]

(1) An operation error will occur if:
- This instruction is used with other than a normal task.

[Program examples]

(1) Program which disables the execution of an event task.

```
DI
```

F/FS	G
○	○

5.13.3 No operation : NOP

Format	NOP

Number of basic steps	1

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

(1) This is a no-operation instruction and does not affect the preceding operations.

[Errors]

(1) There are no operation errors.

F/FS	G
○	○

5.13.4 Block transfer : BMOV

Format	BMOV(D), (S), (n)

Number of basic steps	6

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	○	—	—	—	—
(S)	○	○	—	—	—	—	○	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Transfer destination device starting No.	—
(S)	Transfer source device starting No.	
(n)	Number of words to be transferred	

[Functions]

(1) The contents for n words from device specified with (S) are batch-transferred to the n words from device specified with (D).

(2) Data can be transferred if the devices of the transfer source and destination overlap.
Data are transferred from devices, starting with the one at (S), for transfer of data from devices of larger numbers to those of smaller numbers, or starting with the one at (S)+(n-1) for transfer of data from devices of smaller numbers to those of larger numbers.

(3) When in SV22 virtual mode, specifying Nn (cam No.) at (D) or (S) enables batch-transfer of cam data.
In the Motion controller, the cam data of same cam No. must already have been registered.
The number of transferred words specified with (n) should match the resolution of the specified cam No.

At cam data write

The cam data storage area is rewritten.
- Transfer of data to the cam data area is also executed during cam operation. Be careful not to perform write while operation is being performed with the same cam No.

At cam data read

The cam data storage area is rewritten.
- The cam data in the currently set status are read.

POINT

The BMOV instruction cannot be used for the cam data write/read under the SV22 advanced synchronous control.
Use the CAMWR/CAMWR2 instruction (Cam data write) or the CAMRD instruction (Cam data read). (Refer to Section 5.18.)

(4) The word devices that may be set at (D), (S) and (n) are shown below.

Setting data	Word devices (Note-2)					Bit devices (Note-2), (Note-3)							Cam No. specification
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn	Nn (Note-1)
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-4)	○ (Note-4)	○
(S)	○	○	○	○	○	○	—	○	○	○	○ (Note-4)	○ (Note-4)	○
(n)	○	○	—	○	○	—	—	—	—	—	—	—	—

(Note-1) : "Nn" indicates the cam No.
(Note-2) : The device No. cannot be specified indirectly.
(Note-3) : Specify a multiple of 16 as the device number of bit data.
(Note-4) : PX/PY cannot be set.

(5) The cam No. that may be set as "Nn" is within the following range.

Q173D(S)CPU/Q172D(S)CPU
1 to 64
101 to 164
201 to 264
301 to 364

[Errors]

(1) An operation error will occur if:
- The cam data of cam No. specified with (D) or (S) are not yet registered to the Motion controller
- The resolution of cam No. specified with (D) or (S) differs from the number of transferred words specified with (n)
- (S) to (S)+(n-1) is outside the device range
- (D) to (D)+(n-1) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (S) to (S)+(n-1)
- PX/PY is set in (D) to (D)+(n-1)

when (n) specified is a word device

(2) When conversion of Motion SFC program is made in MT Developer2, an error will occur if:
- (S) to (S)+(n-1) is outside the device range
- (D) to (D)+(n-1) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (S) to (S) + (n-1)
- PX/PY is set in (D) to (D) + (n-1)

when (n) specified is a constant

- (S) is a bit device and the device number is not a multiple of 16
- (D) is a bit device and the device number is not a multiple of 16

[Program examples]

(1) Program which batch-transfers a contents for 5 words from D0 to all data for 5 words from #10

```
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

(2) Program which batch-transfers a contents for 2048 words from #0 to the data area of cam No.2 (resolution 2048)

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

	Cam data of cam No.2			
0th stroke ratio	H0000	Batch transfer ←	#0	H0000
First stroke ratio	H0005		#1	H0005
Second stroke ratio	H000A		#2	H000A
:	:		:	:
2047th stroke ratio	H0000		#2047	H0000

> **POINT**
> Cam stroke ratio is set within 0 to 7FFFH.

(3) Program which batch-transfers a contents for 4 words from X0 to all data for 4 words from #20

```
BMOV #20, X0, K4
```

	b15 ... b0			XF ... X0
#20	0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1	Batch transfer ←	XF ... X0	0 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1
#21	1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1		X1F ... X10	1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1
#22	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1		X2F ... X20	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
#23	1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0		X3F ... X30	1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0

F/FS	G
○	○

5.13.5 Same data block transfer : FMOV

Format	FMOV(D), (S), (n)

Number of basic steps	6

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	○	—	—	—	—
(S)	○	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Transfer destination device starting No.	—
(S)	Device No. which transfer data or data to be transferred are stored.	
(n)	Number of words to be transferred	

[Functions]

(1) The data specified with (S) or contents of device are transferred a part for (n)words of data to the device specified with (D).

(2) The devices that may be set at (D), (S) and (n) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(S)	○	○	○	○	○	○	—	○	○	○	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.
(Note-2) : Specify a multiple of 16 as the device number of bit data.
(Note-3) : PX/PY cannot be set.

[Errors]

(1) An operation error will occur if:
- (D) to (D)+(n-1) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (D) to (D)+(n-1)

$\Big\}$ When (n) specified is a word device

(2) When conversion of Motion SFC program is made in program editing of MT Developer2, an error will occur if:
- (D) to (D)+(n-1) is outside the device range
- (S) is outside the device range
- (n) is 0 or a negative number
- PX/PY is set in (S)
- PX/PY is set in (D) to (D) + (n-1)

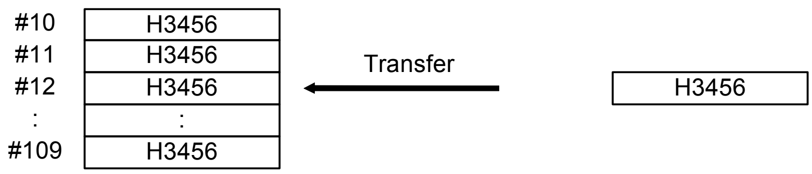
When (n) specified is a constant

- (S) is a bit device and the device number is not a multiple of 16
- (D) is a bit device and the device number is not a multiple of 16

[Program examples]

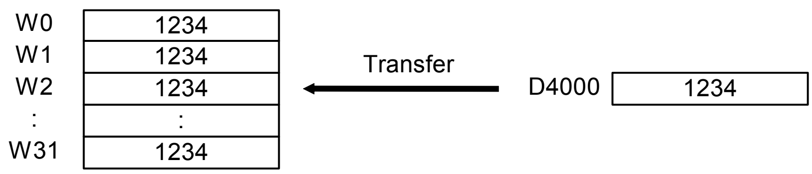
(1) Program which sets 3456H to all data for 100 words from #10

```
FMOV #10,H3456,K100
```



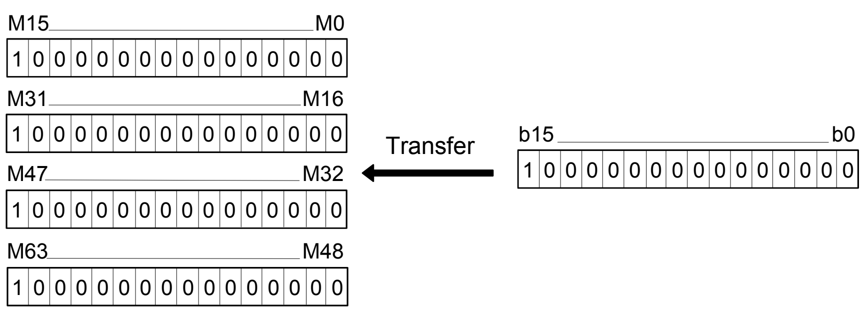
(2) Program which sets a content of D4000 to all data for 50 words from W0

```
FMOV W0,D4000,K50
```



(3) Program which sets 8000H to all data for 4 words from M0

```
FMOV M0, H8000, K4
```



F/FS	G
○	○

5.13.6 Write device data to CPU shared memory of the self CPU: MULTW

Format	MULTW(D), (S), (n), (D1)

Number of basic steps	8

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	—	○	—	—	—	○	—	—	—	—	—
(S)	○	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—
(D1)	○	—	—	—	—	—	—	—	—	—	—

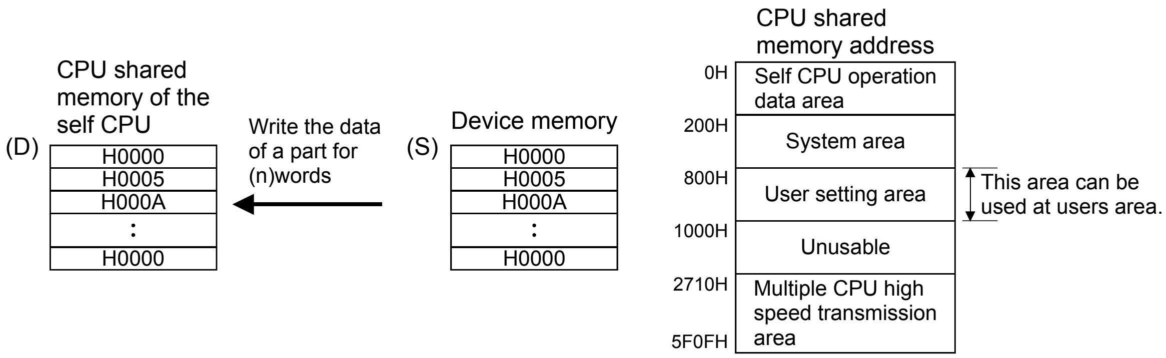
○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	The CPU shared memory address of self CPU of the writing destination device. (800H to FFFH)	—
(S)	Start device No. which writing data are stored.	
(n)	Number of words to be written (1 to 256)	
(D1)	Self CPU device is made to turn on by the writing completion.	

[Functions]

(1) A part for (n)words of data since the device specified with (S) of the self CPU module are written to since the CPU shared memory address specified with (D) of the self CPU module. After writing completion of the device data, the complete bit device specified with (D1) turns on.



(2) Do resetting of the complete bit device by the user program.

(3) Another MULTW instruction cannot be processed until MULTW instruction is executed and a complete bit device is turned on. When MULTW instruction is executed again, before MULTW instruction is executed and complete bit device is turned on, the MULTW instruction executed later becomes an error.

(4) The word devices that may be set at (D), (S), (n) and (D1) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	—	—	—	—	—	—	—
(S)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D1)	—	—	—	—	—	○	○	○	○	—	○ (Note-4)	○ (Note-4)

(Note-1) : The device No. cannot be specified indirectly.
(Note-2) : Specify a multiple of 16 as the device number of bit data.
(Note-3) : PX and PY cannot be set.
(Note-4) : PX can be set. PY cannot be set.

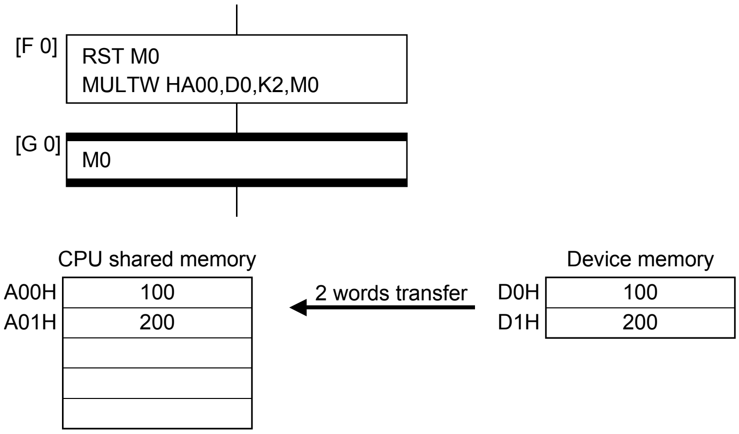
(5) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the number of words (n) to be written.

[Errors]

(1) An operation error will occur if:
- Number of words (n) to be written is outside the range of 1 to 256.
- The CPU shared memory address (D) of self CPU of the writing destination device is outside the range (800H to FFFH) of the CPU shared memory address.
- The CPU shared memory address (D) of self CPU of the writing destination device + number of words (n) to be written is outside the range (800H to FFFH) of the CPU shared memory address.
- Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
- MULTW instruction was executed again before MULTW instruction is executed and complete bit device is turned on.
- (D1) is a write-disabled device.
- (S) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (S) to (S)+(n-1).

[Program examples]

(1) 2 words from D0 is written in the CPU shared memory to since A00H, and transits to next step after confirmation of writing completion.



F/FS	G
○	○

5.13.7 Read device data from CPU shared memory: MULTR

Format	MULTR(D), (S1), (S2), (n)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

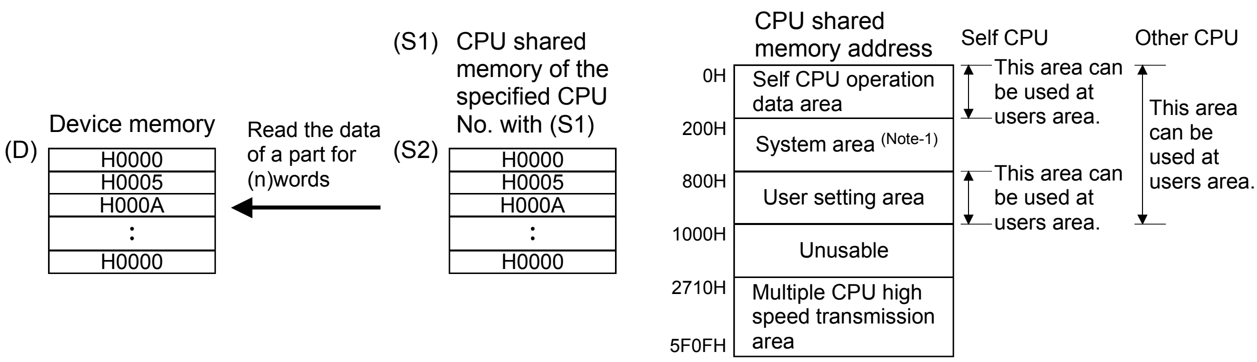
○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Start device No. which stores the reading data.	—
(S1)	First I/O No. of the PLC CPU/Motion CPU which it will be read. (CPU No.1 : 3E0H, CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H)	
(S2)	The CPU shared memory first address of the data which it will be read. (000H to FFFH)	
(n)	Number of words to be read (1 to 256)	

[Functions]

(1) A part for (n)words of data of the target CPU specified with (S1) are read from the address specified with (S2) of the CPU shared memory, and are stored since the device specified with (S2).



(Note-1): This area cannot be read when the target CPU is self CPU.

(2) The word devices that may be set at (D), (S), (n) and (D1) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(S)	○	○	—	○	○	—	—	—	—	—	—	—
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D1)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.
(Note-2) : Specify a multiple of 16 as the device number of bit data.
(Note-3) : PX and PY cannot be set.

(3) When data are read normally from the target CPU specified with (S1), the reading complete flag SM528 to SM531 (CPU No.1 : SM528, CPU No.2 : SM529, CPU No.3 : SM530, CPU No.4 : SM531) corresponding to the target CPU turns on. If data cannot be read normally, the reading complete flag of the target CPU does not turn on.

(4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the Number of words (n) to be read.

(5) When multiple MULTR instructions are executed to the same CPU simultaneously, the reading complete flag SM528 to SM531 turns on/as a result of MULTR that it is executed at the end.

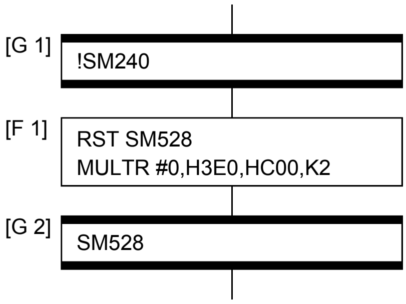
(6) Reset the reading complete flag (SM528 to SM531) using the user program.

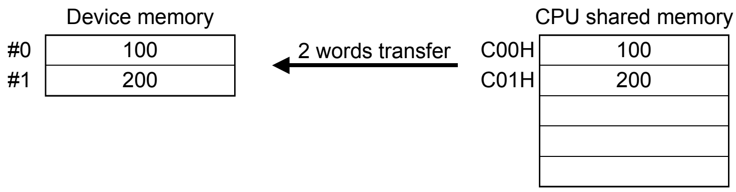
[Errors]

(1) An operation error will occur if:
- Number of words (n) to be read is outside the range of 1 to 256.
- The CPU shared memory first address (S2) of the data which it will be read is outside the range (000H to FFFH) of the CPU shared memory address.
- The CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is outside the range (000H to FFFH) of the CPU shared memory address.
- Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
- Except 3E0H/3E1H/3E2H/3E3H is set at (S1).
- The CPU which reads is resetting.
- The errors are detected in the CPU which read.
- (D) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (D) to (D)+(n-1).

[Program examples]

(1) It checks that a CPU No.1 is not resetting, 2 words is read to since #0 from the CPU shared memory C00H of CPU No.1, and transits to next step after reading completion.





F/FS	G
○	○

5.13.8 Write device data to intelligent function module : TO

Format	TO(D1), (D2), (S), (n)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D1)	—	○	—	—	—	○	—	—	—	—	—
(D2)	—	○	—	—	—	○	—	—	—	—	—
(S)	○	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

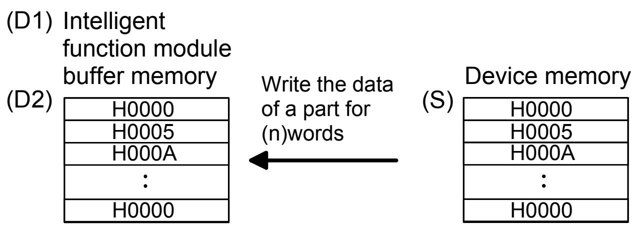
○ : Usable

[Setting data]

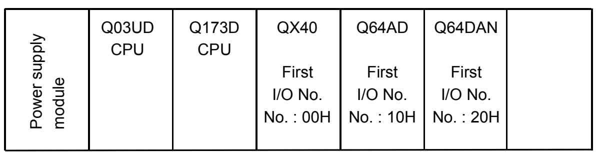
Setting data	Description	Data type of result
(D1)	First I/O No. of the intelligent function module (000H to FF0H)	—
(D2)	First address of the buffer memory which writes data.	
(S)	Start device No. which writing data are stored.	
(n)	Number of words to be written (1 to 256)	

[Functions]

(1) (n) words of data from the device specified with (S) are written to the address specified with (D2) and after of the buffer memory in the intelligent function module controlled by the self CPU specified with (D1).



(2) First I/O No. of the module set by system setting is specified by (D1).



(D1) sets 20H by the system setting when a TO instruction is executed in the D/A conversion module (Q64DAN).

(3) The word devices that may be set at (D1), (D2), (S) and (n) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D1)	○	○	—	○	○	—	—	—	—	—	—	—
(D2)	○	○	—	○	○	—	—	—	—	—	—	—
(S)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.
(Note-2) : Specify a multiple of 16 as the device number of bit data.
(Note-3) : PX and PY cannot be set.

(4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the number of words (n) to be written.

(5) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for intelligent function modules that can be used as the Motion CPU control module.

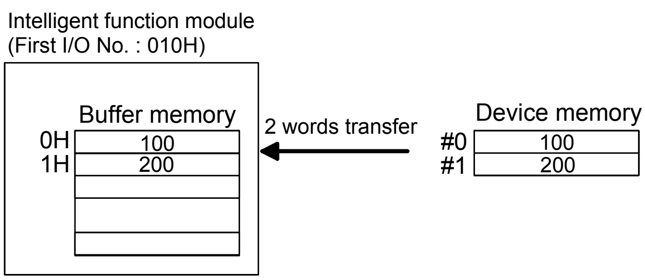
[Errors]

(1) An operation error will occur if:
- Number of words (n) to be written is outside the range of 1 to 256.
- Motion CPU cannot communicate with intelligent function module at the instruction execution.
- Abnormalities of the intelligent function module were detected at the instruction execution.
- I/O No.s specified with (D1) differ from the intelligent function module controlled by the self CPU.
- The address specified with (D2) is outside the buffer memory range.
- Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
- (S) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (S) to (S)+(n-1).

[Program examples]

(1) 2 words from #0 are written to buffer memory address 0H of the intelligent function module (First I/O No. : 010H).

```
TO H010, H0, #0, K2
```



F/FS	G
○	○

5.13.9 Read device data from intelligent function module : FROM

Format	FROM(D), (S1), (S2), (n)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

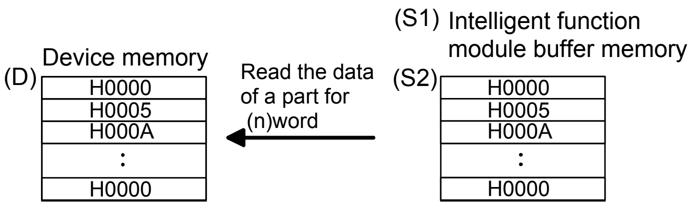
○ : Usable

[Setting data]

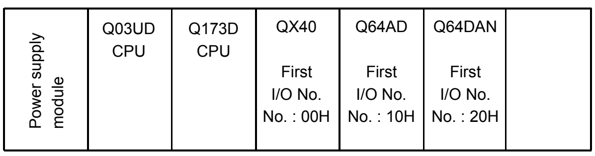
Setting data	Description	Data type of result
(D)	Start device No. which stores the reading data.	—
(S1)	First I/O No. of the intelligent function module (000H to FF0H)	
(S2)	First address of the buffer memory which it will be read.	
(n)	Number of words to be read (1 to 256)	

[Functions]

(1) (n) words of data are read from the address specified with (S2) of the buffer memory in the intelligent function module controlled by the self CPU specified with (S1), and are stored since the device specified with (S2).



(2) First I/O No. of the module set by system setting is specified by (S1).



(S1) sets 10H by the system setting when a FROM instruction is executed in the A/D conversion module (Q64AD).

(3) The word devices that may be set at (D), (S1), (S2) and (n) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(S1)	○	○	—	○	○	—	—	—	—	—	—	—
(S2)	○	○	—	○	○	—	—	—	—	—	—	—
(n)	○	○	—	○	○	—	—	—	—	—	—	—

(Note-1) : The device No. cannot be specified indirectly.
(Note-2) : Specify a multiple of 16 as the device number of bit data.
(Note-3) : PX and PY cannot be set.

(4) Adjust an executive task, the number of transfer word referring to the operation processing time so that this instruction may not obstruct the execution of the motion operation because processing time becomes long in argument to the Number of words (n) to be read.

(5) Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for intelligent function modules that can be used as the Motion CPU control module.

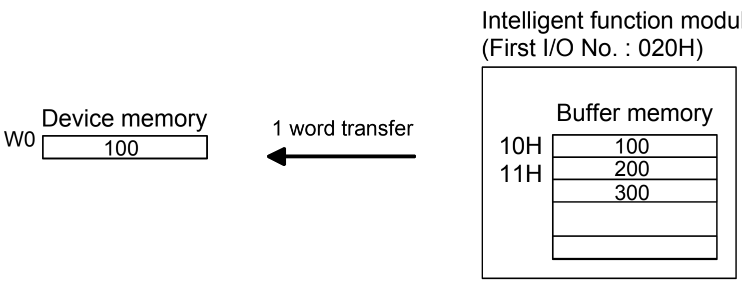
[Errors]

(1) An operation error will occur if:
- Number of words (n) to be read is outside the range of 1 to 256.
- Motion CPU cannot communicate with intelligent function module at the instruction execution.
- Abnormalities of the intelligent function module were detected at the instruction execution.
- I/O No.s specified with (S1) differ from the intelligent function module controlled by the self CPU.
- The address specified with (S2) is outside the buffer memory range.
- Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
- (D) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (D) to (D)+(n-1).

[Program examples]

(1) 1 word is read from the buffer memory address 10H of the intelligent function module (First I/O No. : 020H), and is stored in W0.

```
FROM W0, H020, H10, K1
```



F/FS	G
○	○

5.13.10 Write buffer memory data to head module : RTO QDS Ver.!

Format	RTO(D1), (D2), (D3), (S), (n), (D4)

Number of basic steps	11

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D1)	—	○	—	—	—	○	—	—	—	—	—
(D2)	—	○	—	—	—	○	—	—	—	—	—
(D3)	—	○	—	—	—	○	—	—	—	—	—
(S)	○	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—
(D4)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

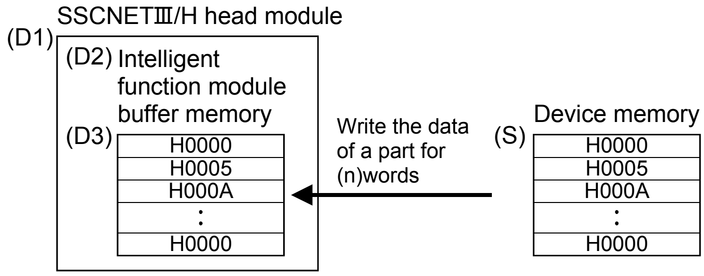
[Setting data]

Setting data	Description	Data type of result
(D1)	Axis No. of the target SSCNETⅢ/H head module (1 to 8)	—
(D2)	First I/O No. of the intelligent function module which writes data. (00 to FEH: First 2 digits when I/O No. is 3 digits)	
(D3)	First address of the buffer memory of the intelligent function module which writes data.	
(S)	Start device No. which writing data are stored.	
(n)	Number of words to be written (1 to 240)	
(D4)	Complete devices (D4+0): Self CPU device is made to turn ON by the writing completion. (D4+1): Self CPU device is made to turn ON by the writing abnormal completion. ("D4+0" also turns ON at the abnormal completion)	

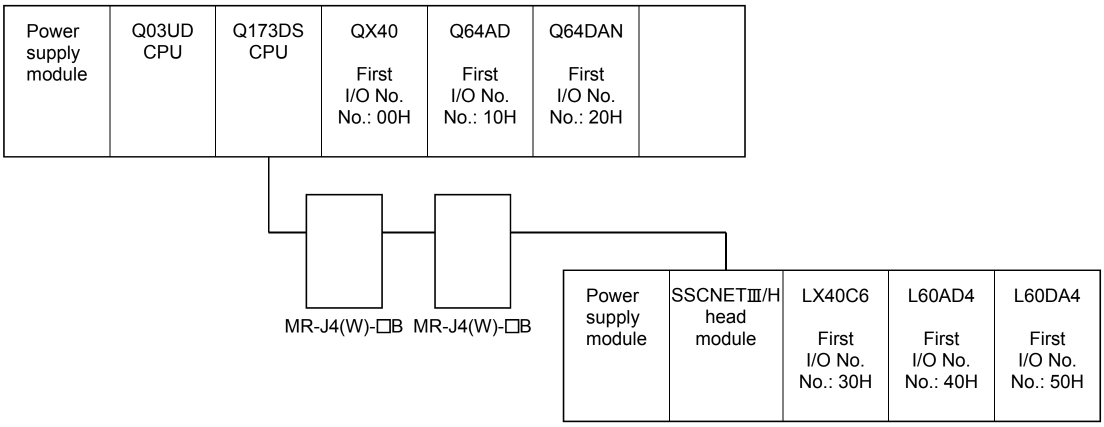
Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Functions]

(1) (n) words of data from the device specified with (S) are written to the address specified with (D3) and after of the buffer memory in the intelligent function module specified with (D2).
The intelligent function module is mounted to the target SSCNETⅢ/H head module specified with (D1).
After writing completion of the device data, the complete bit device specified with (D4) turns ON.



(2) First I/O No. of the module mounted to the SSCNETⅢ/H head module set by system setting is specified by (D2).



(D2) sets 05H by the system setting when a RTO instruction is executed in the D/A conversion module (L60DA4).

(3) The word devices that may be set at (D1), (D2), (D3), (S), (n), and (D4) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D1)	○	○	—	○	○	—	—	—	—	—	—	—
(D2)	○	○	—	○	○	—	—	—	—	—	—	—
(D3)	○	○	—	○	○	—	—	—	—	—	—	—
(S)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D4)	—	—	—	—	—	○	○	○	○	—	○	○

(Note-1): The device No. cannot be specified indirectly.
(Note-2): Specify a multiple of 16 as the device number of bit data.
(Note-3): PX and PY cannot be set.

(4) The following modules can be used.
- Analogue input (L60AD4, L60AD4-2GH)
- Analogue output (L60DA4)
- High-speed counter (LD62, LD62D)

(5) Do resetting of the complete bit device by the user program.

(6) Another RTO instruction cannot be processed until RTO instruction is executed and a complete bit device is turned ON. When RTO instruction is executed again, before RTO instruction is executed and complete bit device is turned ON, the RTO instruction executed later becomes an error.

(7) To write 3 words or more of data to the SSCNETⅢ/H head module, 2 words of writing data are written each time and then repeated. This means that data specified with 3 words or more is not written at the same timing. For data that needs to be written at the same timing use the refresh of the cyclic transmission of the SSCNETⅢ/H head module word device (Buffer memory of SSCNETⅢ/H head module and intelligent function module).
Refer to the manual of SSCNETⅢ/H head module for details.

[Errors]

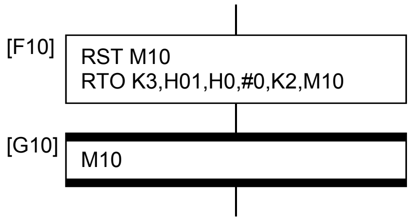
(1) An operation error will occur if:
- Number of words (n) to be written is outside the range of 1 to 240.
- The target SSCNETⅢ/H head module axis number specified with (D1) is outside the range of 1 to 8.
- The target SSCNETⅢ/H head module is not connected at instruction execution.
- Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
- (S) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (S) to (S)+(n-1).
- RTO instruction was executed again before the RTO instruction is executed and complete bit device is turned ON.

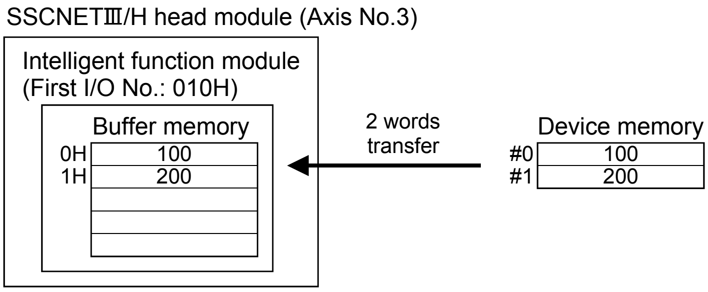
(2) Abnormal completion (D4+1) of complete device turns ON if:
- Abnormalities of the target SSCNETⅢ/H head module were detected at the instruction execution.
- First I/O No. of the intelligent function module specified with (D2) differ from the intelligent function module.
- The first address of the buffer memory of the intelligent function module specified with (D3) is outside the buffer memory range.

[Program examples]

(1) 2 words from #0 are written to buffer memory address 0H of the intelligent function module (First I/O No.: 010H) on the 3rd axis of the SSCNETⅢ/H head module.

RTO K3, H01, H0, #0, K2, M10





F/FS	G
○	○

5.13.11 Read buffer memory data from head module: RFROM QDS Ver.!

Format	RFROM(D), (S1), (S2), (S3), (n), (D1)

Number of basic steps	11

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	○	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(S3)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—
(D1)	○	—	—	—	—	—	—	—	—	—	—

○ : Usable

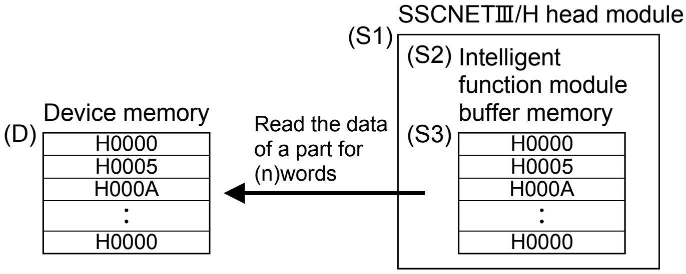
[Setting data]

Setting data	Description	Data type of result
(D)	Start device No. which stores the reading data.	—
(S1)	Axis No. of the target SSCNETⅢ/H head module (1 to 8)	
(S2)	First I/O No. of the intelligent function module which data to be read are stored. (00 to FEH: First 2 digits when I/O No. is 3 digits)	
(S3)	First address of the buffer memory of the intelligent function module which stores the data to be read.	
(n)	Number of words to be read (1 to 240)	
(D1)	Complete devices (D1+0): Self CPU device is made to turn ON by the reading completion. (D1+1): Self CPU device is made to turn ON by the reading abnormal completion. ("D1+0" also turns ON at the abnormal completion)	

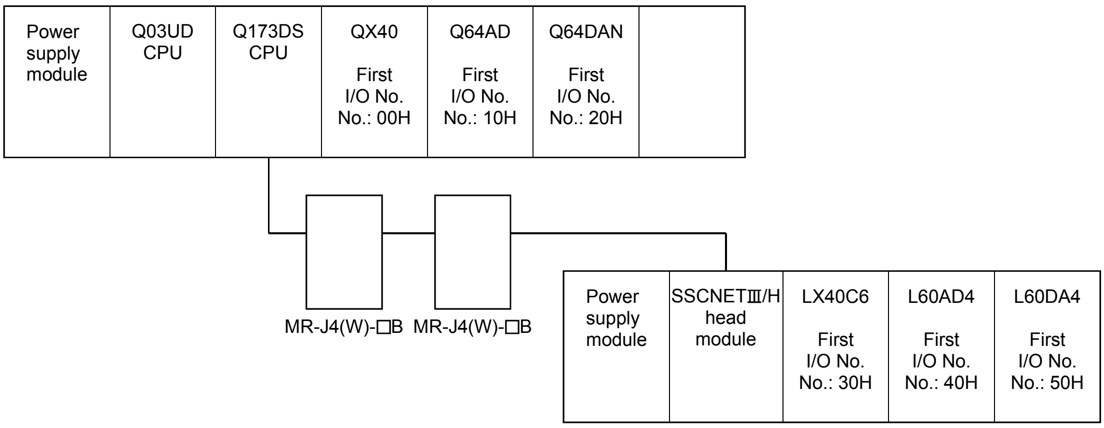
Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Functions]

(1) (n) words of data are read from the address specified with (S3) of the buffer memory in the intelligent function module specified with (S2).
The intelligent function module is mounted to the target SSCNETⅢ/H head module specified with (S1).
The data is written to device specified with (D) and after.



(2) First I/O No. of the module mounted to the SSCNETⅢ/H head module set by system setting is specified by (S2).



(S2) sets 04H by the system setting when a RFROM instruction is executed in the A/D conversion module (L60AD4).

(3) The word devices that may be set at (D), (S1), (S2), (S3), (n), and (D1) are shown below.

Setting data	Word devices (Note-1)					Bit devices (Note-1), (Note-2)						
	Dn	Wn	SDn	U□\Gn	#n	Mn	U□\Gn.m	Bn	Fn	SMn	Xn	Yn
(D)	○	○	—	○	○	○	—	○	○	—	○ (Note-3)	○ (Note-3)
(S1)	○	○	—	○	○	—	—	—	—	—	—	—
(S2)	○	○	—	○	○	—	—	—	—	—	—	—
(S3)	○	○	—	○	○	—	—	—	—	—	—	—
(n)	○	○	—	○	○	—	—	—	—	—	—	—
(D1)	—	—	—	—	—	○	○	○	○	—	○ (Note-3)	○ (Note-3)

(Note-1): The device No. cannot be specified indirectly.
(Note-2): Specify a multiple of 16 as the device number of bit data.
(Note-3): PX and PY cannot be set.

(4) The following modules can be used.
- Analogue input (L60AD4, L60AD4-2GH)
- Analogue output (L60DA4)
- High-speed counter (LD62, LD62D)

(5) Do resetting of the complete bit device by the user program.

(6) Another RFROM instruction cannot be processed until RFROM instruction is executed and a complete bit device is turned ON. When RFROM instruction is executed again, before RFROM instruction is executed and complete bit device is turned ON, the RFROM instruction executed later becomes an error.

(7) To read 5 words or more of data from the SSCNETⅢ/H head module, 4 words of read data are read each time and then repeated. This means that data specified with 5 words or more is not read at the same timing. For data that needs to be read at the same timing use the refresh of the cyclic transmission of the SSCNETIII/H head module word device (Buffer memory of SSCNETⅢ/H head module and intelligent function module).
Refer to the manual of SSCNETⅢ/H head module for details.

[Errors]

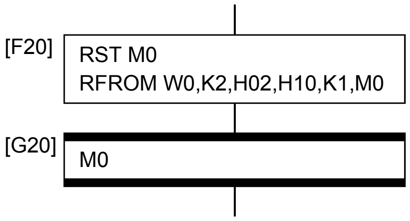
(1) An operation error will occur if:
- Number of words (n) to be read is outside the range of 1 to 240.
- The target SSCNETⅢ/H head module axis number specified with (S1) is outside the range of 1 to 8.
- The target SSCNETⅢ/H head module is not connected at instruction execution.
- Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
- (D) is a bit device and device number is not a multiple of 16.
- PX/PY is set in (D) to (D)+(n-1).
- RFROM instruction was executed again before the RFROM instruction is executed and complete bit device is turned ON.

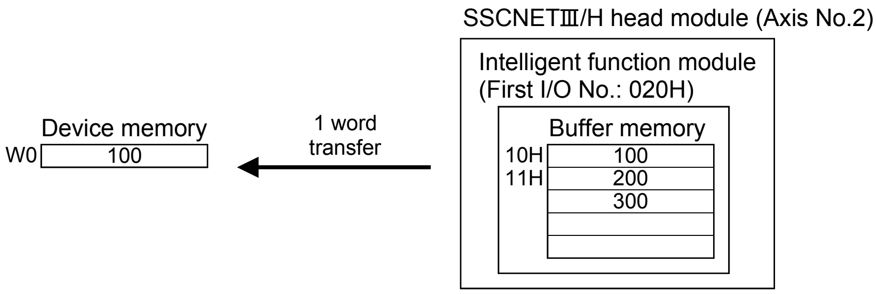
(2) Abnormal completion (D1+1) of complete device turns ON if:
- Abnormalities of the target SSCNETⅢ/H head module were detected at the instruction execution.
- First I/O No. of the intelligent function module specified with (S2) differ from the intelligent function module.
- The first address of the buffer memory of the intelligent function module specified with (S3) is outside the buffer memory range.

[Program examples]

(1) 1 word is read from the buffer memory address 10H of the intelligent function module (First I/O No. : 020H) on the 2nd axis of the SSCNETⅢ/H head module, and is stored in W0.

RFROM W0, K2, H02, H10, K1, M0





F/FS	G
—	○

5.13.12 Time to wait : TIME

Format	TIME(S)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	○	—	—	○	○	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Waiting time (0 to 2147483647)[ms]	Logical type (true/false)

[Functions]

(1) A wait state continues for the time specified with (S). The result is false when the elapsed time is less than the preset time, or the result is true and execution transits when the preset time has elapsed.

(2) When a 16-bit integer type word device is used to specify any of 32768 to 65535[ms] at (S), convert it into an unsigned 32-bit integer value with ULONG. (Refer to the program example.)

[Errors]

(1) An operation error will occur if:
- (S) is an indirectly specified device and its device No. is outside the range. ; or
- The data (device data at indirect specification) specified with (S) is outside the range of 0 to 2147483647.

[Program examples]

(1) Program which sets a wait of 60 seconds (when constant is specified)

```
TIME K60000
```

(2) Program for a case where there may be a wait of 32768 to 65535[ms] for 16-bit integer type indirect designation (#0)

```
TIME ULONG(#0)
```

(3) Program which SETS (RSTs) a bit device when the specified time has elapsed

```
SET M100 = TIME K60000
```

POINT

(1) When the waiting time setting is indirectly specified with a word device, the value imported first is used as the device value for exercising control. The set time cannot be changed if the device value is changed during a wait state.
(2) The TIME instruction is equivalent to a conditional expression, and therefore may be set on only the last line of a transition (G) program.
(3) When the transition program (Gn) of the same number having the TIME instruction setting is used in multiple Motion SFC programs, avoid running them at the same time. (If they are run simultaneously, the waiting time in the program run first will be illegal.)
(4) Another transition program (Gn) can executed a time of instruction by multiple Motion SFC program simultaneously. (Multi active step less than 256.)
(5) While time by TIME instruction waits, the wait time can not be stopped.

F/FS	G
○	○

5.14 Comment Statement : //

Format	//

Number of basic steps	—

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
—	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

(1) A character string from after // to a block end is a comment.

[Errors]

(1) There are no operation errors.

[Program examples]

(1) Example which has commented a substitution program.

```
D0=D1//Substitutes the D0 value (16-bit integer data) to D1.
```

F/FS	G
○	○

5.15 Vision System Dedicated Function Ver.!

5.15.1 Open line : MVOPEN

Format	MVOPEN (S1), (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision system (camera) No. to log on (1 to 8)	—
(S2)	Timeout until vision system logon is completed (1 to 32767) [×10ms]	

[Functions]

(1) The vision system specified with (S1) is logged on.

(2) The Motion SFC program execution transits to the next block without waiting for the completion on the vision system logon. After process completion, the status storage device value set in the Ethernet communication line setting parameter is 20 (reception enable). (Refer to APPENDIX 3.2.)

(3) (S2) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 8.
- The (S2) data is outside the range of 1 to 32767.
- MVOPEN is executed again for the vision system that has been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.
- The logon is not completed when the specified timeout time has elapsed.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Program examples]

(1) Program which logs on the vision system of the vision system (camera) 3

```
MVOPEN K3
```

F/FS	G
○	○

5.15.2 Load a program : MVLOAD

Format	MVLOAD(S1), (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision program No. to load (1 to 32)	—
(S2)	Timeout time until the job loading completed (1 to 32767) [×10ms]	

[Functions]

(1) The job of the vision program No. specified with (S1) is loaded to the vision system (The process of developing a job file stored in the vision system into the memory in the vision system, and making it an active job). And the status is changed to on-line.

(2) The Motion SFC program execution transits to the next block without waiting for the process completion. After process completion, the status storage device value set in the Ethernet communication line setting parameter is 20 (reception enabled). When the job loading is executed successfully and the vision system becomes online, the status storage device value set in the vision program operation setting parameter specified with (S1) is 1 (Online after the job loading completed). (Refer to APPENDIX 3.2.)

(3) When the job of the vision program No. specified with (S1) has been loaded to the vision system, the job is forced to reload. When job contents have been changed by In-Sight® Explorer, etc., save the job in advance to prevent from losing.

(4) (S2) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).
The process time is changed according to job contents in the vision system. Set the timeout time according to the vision system and the job contents.

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 32.
- The (S2) data is outside the range of 1 to 32767.
- The vision system which is used in the vision program specified with (S1) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system and the job.
- The job loading is not completed when the specified timeout time has elapsed.

[Program examples]

(1) Program which loads the job of the vision program No.2

```
MVLOAD K2
```

F/FS	G
○	○

5.15.3 Send an image acquisition trigger : MVTRG

Format	MVTRG(S1), (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision system (camera) No. issues a trigger (1 to 8)	—
(S2)	Timeout time until execution result is received from vision system (1 to 32767) [×10ms]	

[Functions]

(1) The job is executed if a trigger is issued to the vision system specified with (S1) and the result is stored in the image data storage device set in the vision program operation setting parameter.

(2) The Motion SFC program execution transits to the next block without waiting for the process completion. After the job is ended in the vision system and the sending of image data (created by vision processing) by TCP/IP protocol is completed, the status storage device value set in the Ethernet communication line setting parameter is 40 (Image data reception completed).
When the read value is set in the vision system parameters, the data is stored in the read value storage device, and the status storage device value set in the Ethernet communication line setting parameter is 50 (value cell reception completed). (Refer to APPENDIX 3.2.)

(3) (S2) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 8.
- The (S2) data is outside the range of 1 to 32767.
- The vision system specified with (S1) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.
- The tag specified with reading value cell or the spreadsheet data is not an integer value.
- The current process is not completed when the specified timeout time has elapsed.

[Program examples]

(1) Program which issues a trigger to the vision system (camera) 1

```
MVTRG K1
```

F/FS	G
○	○

5.15.4 Start a program : MVPST

Format	MVPST(S1), (S2)

Number of basic steps	4

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision program No. to start (1 to 32)	—
(S2)	Timeout time until execution result is received from vision system (1 to 32767) [×10ms]	

[Functions]

(1) The following process is executed.
The job of the vision program No. specified with (S1) is loaded to the vision system (The process of developing a job file stored in the vision system into the memory in the image system, and making it an active job). And the status is changed to on-line. Then, the job is executed if a trigger is issued, and the result is stored in the image data storage device set in the vision program operation setting parameter.

(2) The Motion SFC program execution transits to the next block without waiting for the process completion. After the job is ended in the vision system and the sending of vision data (created by vision processing) by TCP/IP protocol is completed, the status storage device value set in the Ethernet communication line setting parameter is 40 (Image data reception completed).
When the read value is set in the vision system parameters, the data is stored in the read value storage device, and the status storage device value set in the Ethernet communication line setting parameter is 50 (value cell reception completed). (Refer to APPENDIX 3.2.)

(3) When the job of the vision program No. specified with (S1) has been loaded to the vision system, the following process is executed without reloading. The job is executed if a trigger is issued to the vision system, and the result is stored in the image data storage device set in the vision program operation setting parameter.

(4) (S2) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).
The process time is changed according to job contents in the vision system. Set the timeout time according to the vision system and the job contents.

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 32.
- The (S2) data is outside the range of 1 to 32767.
- The vision system which is used in the vision program specified with (S1) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system and the job.
- The tag specified with reading value cell or the spreadsheet data is not an integer value.
- The current process is not completed when the specified timeout time has elapsed.

[Program examples]

(1) Program which executes the job of the vision program No.20

```
MVPST K20
```

F/FS	G
○	○

5.15.5 Input data : MVIN

Format	MVIN(S1), (S2), (D), (S3)

Number of basic steps	8 or more

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○ (Note-2)	—	—	—	—	—	—	—	—	—
(D)	—	—	○ (Note-3)	○ (Note-3)	—	—	—	—	—	—	—
(S3)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.
(Note-2): Specify the start device which stores the character string data. The character string can be specified directly.
(Note-3): Data is the same format as the job set in the vision system. (If the format is different, the data is converted to the type specified with (D).)

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision system (camera) No. to which data will be read (1 to 8)	—
(S2)	Spreadsheet cell or tag to which data will be read	
(D)	Device which will store the reading data	
(S3)	Timeout time until data is read from vision system (1 to 32767) [×10ms]	

[Functions]

(1) The numerical value of spreadsheet cell or tag specified with (S2) is stored in the device specified with (D) from the vision system specified with (S1).

POINT

The operation error (error code: 18023) will occur if data of spreadsheet cell or tag specified with (S2) is not a numerical value (character string, etc.).
Use MVCOM instruction (Refer to Section 5.15.9.).

(2) The Motion SFC program execution transits to the next block without waiting for the process completion. After process completion, the status storage device value set in the Ethernet communication line setting parameter is 20 (reception enabled). (Refer to APPENDIX 3.2.)

(3) In (S2), write directly the spreadsheet cell or tag as a 32 one-byte character or less character string enclosed with double quotation, or set the head of a device in which a 32 one-byte character or less character string is stored.

Designation methods of the character string are shown below.

Setting with cell	The spreadsheet row (A to Z) and line (0 to 399) are arranged and written. (Example) When cell is A5, "A5" is set.
Setting with tag	The Symbolic tag name is written in the original state. (Example) When tag is Job.Pass_count, "Job.Pass_count" is set.

(4) The numerical value read from the vision system is stored with the following format.

Numerical data format of spreadsheet cell or tag	Data format stored in (D)	Number of points
Integer value	32-bit integer value type	Consecutive 2 points
Floating-point value	64-bit floating-point type	Consecutive 4 points

(5) (S3) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 8.
- The number of character string of spreadsheet cell or tag specified with (S2) outside the range of 1 to 32 bytes.
- The spreadsheet cell or tag specified with (S2) does not exist.
- The data of spreadsheet cell or tag specified with (S2) is not a numerical value.
- The (S3) data is outside the range of 1 to 32767.
- The vision system specified with (S1) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.
- The reading data is not completed when the specified timeout time has elapsed.

[Program examples]

(1) Program which stores the numerical value stored in the tag "pattern_1.fixture.score" of the vision system (camera) 1 to D3000 or later

```
MVIN K1, "pattern_1.fixture.score", D3000F
```

(2) Program which stores the numerical value from the tag indicated by a character string stored in D100 or later, to D2000 or later for the vision system (camera) 3

```
MVIN K3, D100, D2000L
```

F/FS	G
○	○

5.15.6 Output data : MVOUT Ver.!

Format	MVOUT(S1), (S2), (S3), (S4)

Number of basic steps	8 or more

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○ (Note-2)	—	—	—	—	—	—	—	—	—
(S3)	—	○ (Note-3)	○ (Note-3)	○ (Note-3)	—	○ (Note-3)	○ (Note-3)	○ (Note-3)	—	—	—
(S4)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.
(Note-2): Specify the start device which stores the character string data. The character string can be specified directly.
(Note-3): Data is the same format as data to be transferred. The character string can be specified directly.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision system (camera) No. to transfer data (1 to 8)	—
(S2)	Spreadsheet cell or tag to transfer data	
(S3)	Data to be transferred	
(S4)	Timeout time until data is transferred to vision system (1 to 32767) [×10ms]	

[Functions]

(1) Data specified with (S3) is transferred to spreadsheet cell or tag specified with (S2) of vision system specified with (S1).

(2) The Motion SFC program execution transits to the next block without waiting for the process completion. After process completion, the status storage device value set in the Ethernet communication line setting parameter is 20 (reception enabled). (Refer to APPENDIX 3.2.)

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(3) In (S2), write directly the spreadsheet cell or tag as a 32 one-byte character or less character string enclosed with double quotation, or set the head of a device in which a 32 one-byte character or less character string is stored.

Designation methods of the character string are shown below.

Setting with cell	The spreadsheet row (A to Z) and line (0 to 399) are arranged and written. (Example) When cell is A5, "A5" is set.
Setting with tag	The Symbolic tag name is written in the original state. (Example) When tag is Job.Pass_count, "Job.Pass_count" is set.

(4) In (S3), set the head of a device that store data to be transferred to spreadsheet cell or tag.

Data type specified with (S3)	Number of points	Setting example
16-bit integer value type	1 point	D1000
32-bit integer value type	Consecutive 2 points	D2000L
64-bit floating-point type	Consecutive 4 points	D3000F

Also, the character string of constants or 99 one-byte character or less character string can be specified directly.

Data type specified with (S3)	Setting example
16-bit integer value type	K12345
32-bit integer value type	K12345678L
64-bit floating-point type	K1234.5
Character string	"MITSUBISHI"

POINT

If the floating-point data is transferred to the vision system, it is handled as 32-bit floating-point data. The number of effective digits is approx. 7 digits. Data in the seven digits or later may not match the (S3) data.

(5) (S4) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 8.
- The number of character string of spreadsheet cell or tag specified with (S2) outside the range of 1 to 32 bytes.
- The spreadsheet cell or tag specified with (S2) does not exist.
- The data type of spreadsheet cell or tag specified with (S2) is different from data format specified with (S3).
- The (S3) data is outside the range.
- The (S4) data is outside the range of 1 to 32767.
- The vision system specified with (S1) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.
- The reading data is not completed when the specified timeout time has elapsed.

[Program examples]

(1) Program which transfers the floating-point value stored in D3000F to the tag "Calib_1.World_Point0.X" of the vision system (camera) 1

```
MVOUT K1, "Calib_1.World_Point0.X", D3000F
```

F/FS	G
○	○

5.15.7 Reset a status storage device : MVFIN

Format	MVFIN(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S)	Vision system (camera) No. to reset the status storage device (1 to 8)	—

[Functions]

(1) The status storage device value set in the Ethernet communication line setting parameter specified with (S) is 20 (reception enabled). (Refer to APPENDIX 3.2.)

(2) When a trigger is issued to the vision system, the status storage device is reset by MVFIN instruction in advance, and the process completion for trigger needs to be detected.

[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range of 1 to 8.
- The vision system specified with (S) has not been logged on.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.
- The status storage device value set in the Ethernet communication line setting parameter is not 20 (reception enabled), 40 (Image data reception completed), or 50 (value cell reception completed).

[Program examples]

(1) Program which resets the status storage device for the vision system (camera) 1

```
MVFIN K1
```

F/FS	G
○	○

5.15.8 Close line : MVCLOSE

Format	MVCLOSE(S)

Number of basic steps	2

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S)	Vision system (camera) No. to log off (1 to 8)	—

[Functions]

(1) The vision system specified with (S) is logged off (disconnected).
The status storage device value set in the Ethernet communication line setting parameter is 0 (not connected). (Refer to APPENDIX 3.2.)

(2) MVCLOSE instruction is not operated for the vision system which is not logged on (not connected).

[Errors]

(1) An operation error will occur if:
- The (S) data is outside the range of 1 to 8.
- The vision system parameter setting (Refer to APPENDIX 3.2.) is different from the setting on the vision system.

[Program examples]

(1) Program which logs off the vision system of the vision system (camera) 1

```
MVCLOSE K1
```

F/FS	G
○	○

5.15.9 Send a command for native mode : MVCOM

Format	MVCOM(S1), (S2), (D), (S3), (S4)

Number of basic steps	9 or more

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○ (Note-2)	—	—	—	—	—	—	—	—	—
(D)	—	○	—	—	—	—	—	—	—	—	—
(S3)	—	○	—	—	—	○	—	—	—	—	—
(S4)	—	○	—	—	—	○	—	—	—	—	—

○ : Usable

(Note-1): Special register (SD) cannot be used.
(Note-2): Specify the start device which stores the character string data. The character string can be specified directly.

[Setting data]

Setting data	Description	Data type of result
(S1)	Vision system (camera) No. to which Native Mode command will be sent (1 to 8)	—
(S2)	Native Mode command character string	
(D)	Start device which stores return value	
(S3)	Mode setting for return value conversion	
(S4)	Timeout time until data is read from vision system (1 to 32767) [×10ms]	

[Functions]

(1) Native Mode command specified with (S2) is sent to the vision system specified with (S1), and the return value is stored in the device specified with (D) with the format specified with (S3).

(2) The Motion SFC program execution transits to the next block without waiting for the Native Mode command completion.
After process completion, the status storage device value set in the Ethernet communication line setting parameter is 20 (reception enabled). (Refer to APPENDIX 3.2.)

(3) Refer to the Cognex Corporation vision system manual and help sections, etc. for details of Native Mode command specified with (S2).
In (S2), write directly the Native Mode command as a 99 one-byte character or less character string enclosed with double quotation, or set the head of a device in which a 191 one-byte character or less character string is stored.

(4) The return value of Native Mode command is stored as below by specifying (S3) in the device specified with (D).
When the return value data is the following ([CR] indicates a return code, and [LF] indicates a line feed code.)

1[CR][LF]	Status code
19[CR][LF] <Float>92</Float>[CR][LF]	Data

(a) When 0 (ASCII Mode) is specified with (S3), the data is stored from the device specified with (D) by the following procedure.

1) Status code (16-bit integer format)

	(D)
1 →	+0

2) Number of character string parts of data 3) (16-bit integer format)

24 →	+1

3) The character string parts of data (ASCII code)
(The end code [NULL] is stored at the end of data.)

"1"	"9" →	+2
[CR]	[LF] →	+3
"<"	"F" →	+4
"l"	"o" →	+5
"a"	"t" →	+6
">"	"9" →	+7
"2"	"<" →	+8
"/"	"F" →	+9
"l"	"o" →	+10
"a"	"t" →	+11
">"	[CR] →	+12
[LF]	[NULL] →	+13

(b) When 1 (Binary Mode) is specified with (S3), the data is stored from the device specified with (D) by the following procedure.

1) Status code (16-bit integer format)

	(D)
1 →	+0

2) Number of character string parts of data 3) (16-bit integer format)

16 →	+1

3) Data converted into a 64-bit floating-point type value

19.0 →	+5	+4	+3	+2
92.0 →	+9	+8	+7	+6

(5) (S4) is set in increments of 10ms.
When the setting is omitted, the timeout time is 10 seconds (same as setting 1000).

[Errors]

(1) An operation error will occur if:
- The (S1) data is outside the range of 1 to 8.
- The Native Mode command specified with (S2) is wrong.
- The (S3) data is outside the range of 0 to 1.
- The (S4) data is outside the range of 1 to 32767.
- The vision system specified with (S1) has not been logged on.
- The character string for Native Mode specified with (S2) exceeds the range of the number of characters.
- The return value data is not a numerical value when 1 (Binary Mode) is specified with (S3).
- The vision system parameter setting (Refer to APPENDIX 3.2) is different from the setting on the vision system.
- The device storage of the Native Mode command return value is not completed when the specified timeout time has elapsed.
- The return value of Native Mode command exceeds the range of (D) to the end of the device. (An operation error occurs at the storing of data up until the end of the device).

[Program examples]

(1) Program which sends the Native Mode command "EV GetCellValue ("distance_1.max")" to the vision system (camera) 1, and stores the return value in #0 or later in Binary Mode

```
MVCOM K1,"EV GetCellValue ("distance_1.max")",#0,K1
```

F/FS	G
○	○

5.16 Data Control

5.16.1 16-bit integer type scaling: SCL

Format	SCL(S1), (S2), (S3), (D)

Number of basic steps	8

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	○	—	—
(S3))	—	○	—	—	—	—	—	—	—	—	—
(D)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which specifies the search/conversion method 0: Positive conversion by sequential search 1: Inverse conversion by sequential search 2: Positive conversion by binary search 3: Inverse conversion by binary search	—
(S2)	Input value for positive/inverse conversion	
(S3)	Start device No. which stores the scaling conversion data	
(D)	Device No. which stores the conversion result	

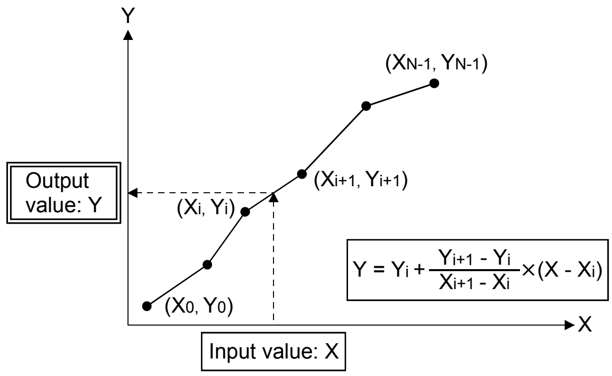
[Overview]

(1) 16-bit integer type scaling calculates the output value from the set input value based on the scaling conversion data where a maximum of 4000 points data ((X_0, Y_0) to (X_{N-1}, Y_{N-1}), N: the number of points) are defined.
Set the point data corresponding to the input value in ascending order. (Positive conversion: $X_0 < X_1 <.....< X_{N-1}$, Inverse conversion: $Y_0 < Y_1 <.....< Y_{N-1}$)

(2) The method for output value calculation is either positive conversion (Input value: point X, Output value: point Y) or inverse conversion (Input value: point Y, Output value: point X) and is specified with (S1).
Each of the calculation methods is as follows.
(a) When the input value is between two points of scaling conversion data, the output value is calculated from the nearest two points of the input value.

Positive conversion

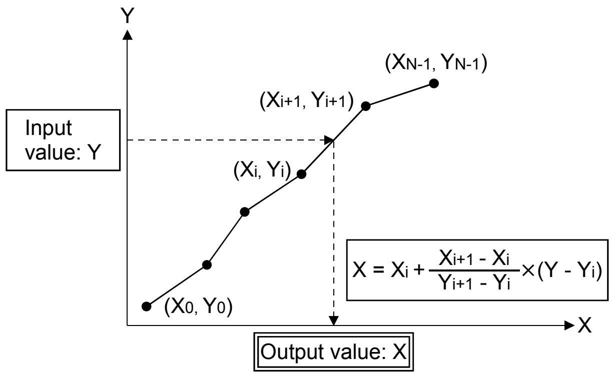
• $X_0 \leq X$ (Input value) $\leq X_{N-1}$



$$Y = Y_i + \frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \times (X - X_i)$$

Inverse conversion

• $Y_0 \leq Y$ (Input value) $\leq Y_{N-1}$



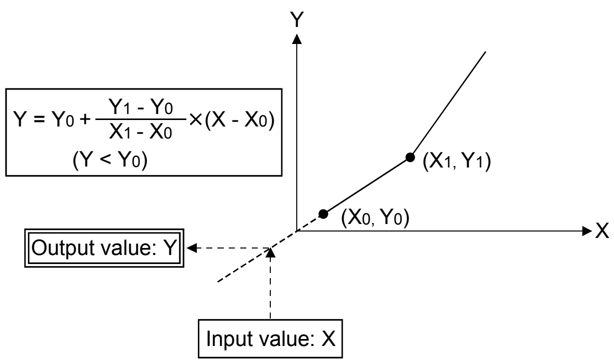
$$X = X_i + \frac{X_{i+1} - X_i}{Y_{i+1} - Y_i} \times (Y - Y_i)$$

N: Number of points

(b) When the input value is other than (a), the output value is calculated from the first or last two points of scaling conversion data.

Positive conversion

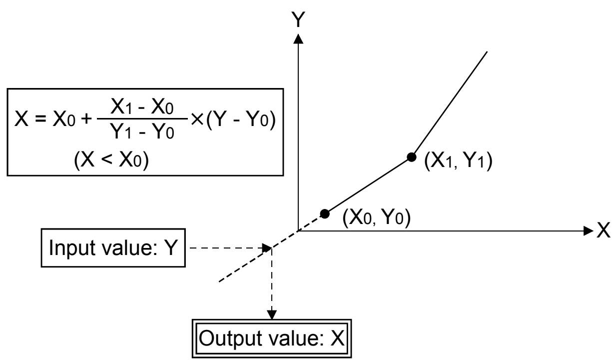
• X (Input value) < X_0



$$Y = Y_0 + \frac{Y_1 - Y_0}{X_1 - X_0} \times (X - X_0)$$
$(Y < Y_0)$

Inverse conversion

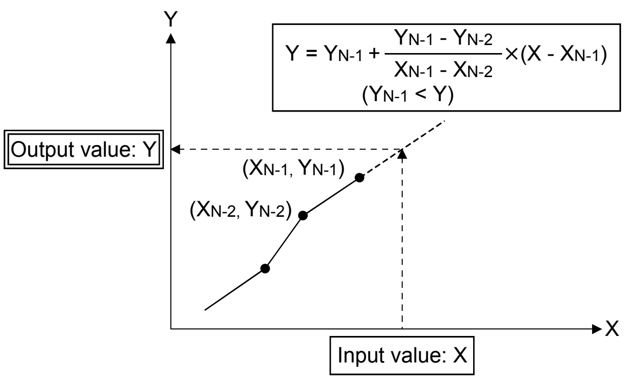
• Y (Input value) < Y_0



$$X = X_0 + \frac{X_1 - X_0}{Y_1 - Y_0} \times (Y - Y_0)$$
$(X < X_0)$

Positive conversion

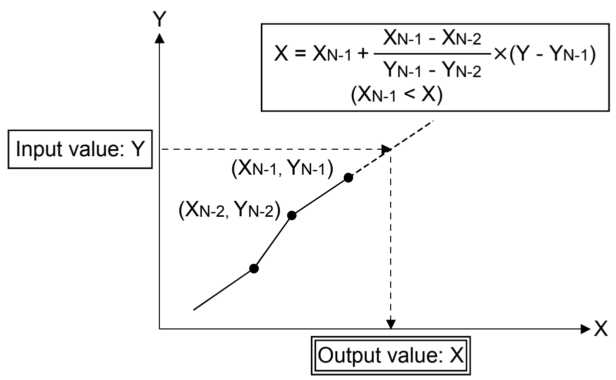
• X_{N-1} < X (Input value)



$$Y = Y_{N-1} + \frac{Y_{N-1} - Y_{N-2}}{X_{N-1} - X_{N-2}} \times (X - X_{N-1})$$
$(Y_{N-1} < Y)$

Inverse conversion

• Y_{N-1} < Y (Input value)



$$X = X_{N-1} + \frac{X_{N-1} - X_{N-2}}{Y_{N-1} - Y_{N-2}} \times (Y - Y_{N-1})$$
$(X_{N-1} < X)$

N: Number of points

POINT

When the input value is outside the range of scaling conversion data or calculation result of output value is outside the range of -32768 to 32767, an operation error will occur.

[Functions]

(1) Conversion of the input value specified with (S2) is executed according to the search/conversion method specified with (S1), using the scaling conversion data of device (S3) or later. The conversion result is stored in the device specified with (D).

(2) In the scaling, point data used for output value calculation must be searched from the input value, and the search method is specified with (S1).
The search method is either sequential search or binary search, and the features are shown below. Specify the search method according to the intended use.

	Search times when the number of points is 4000	Processing time	Precaution
Sequential search	1 to 4000 times	0.03 to 10.7 [ms] Since the data is searched in order from the head in sequential search, the maximum processing time increases in proportion to the number of points.	During search processing, whether the point data corresponding to the input value is in ascending order can be checked. If the input value is not in ascending order, an operation error will occur.
Binary search	12 times	0.05 [ms] Binary search requires relatively short search time since the processing time does not increase in proportion to the number of points.	During search processing, point data required for the binary search are only referred to. If the data is not in ascending order, the calculation result of output value could be unexpected one since all of the point data corresponding to the input value cannot be confirmed.

(3) The device No. specified with (S3) should be an even number. Set the point data in the specified device as follows.

Off set	Name		Description	Range
+0	The number of points (N)		Set the number of points for the scaling conversion data.	2 to 4000
+1	Unusable		Set 0.	0
+2	Point 0	X_0	Set the point data of (X_0, Y_0) to (X_{N-1}, Y_{N-1}) so that the device No. is in consecutive order.	-32768 to 32767
+3		Y_0		
+4	Point 1	X_1		
+5		Y_1		
+6	Point 2	X_2		
+7		Y_2		
:	:			
+ (2N)	Point (N-1)	X_{N-1}		
+ (2N+1)		Y_{N-1}		

POINT

Set the point data corresponding to the input value in ascending order. (Positive conversion: $X_0 < X_1 <.....< X_{N-1}$, Inverse conversion: $Y_0 < Y_1 <.....< Y_{N-1}$)

(4) When the conversion result to be stored in the device specified with (D) is not an integer value, its fractional portion is rounded down.

[Errors]

(1) An operation error will occur, and the conversion of input value will not be executed if:
- (S1) is set to other than 0 to 3.
- (S3) is not an even-numbered device.
- The number of points at the point table specified with (S3) is outside the range of 2 to 4000.
- Point table specified with (S3) is outside the device range.
- In sequential search ((S1) is 0 or 1.), the point corresponding to the input value (Positive conversion: X_0 to X_{N-1}, Inverse conversion: Y_0 to Y_{N-1}) is not in ascending order.
- The conversion result is outside the range of -32768 to 32767.

[Program examples]

(1) Program which sets 4 points of scaling conversion data to D3000 to D3009 and substitutes the output value, which is positively converted based on the input value "7500", to D3106.

[Motion SFC program]

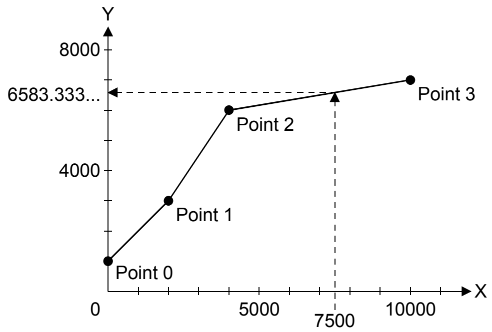
```
16-bit scaling

[F10]
//Scaling conversion data set
D3000=K4          //Number of point=4
D3001=K0          //Unusable area
//Set the point 0 to 3
D3002=K0          //Point 0
D3003=K1000       //(X0, Y0)=(0, 1000)
D3004=K2000       //Point 1
D3005=K3000       //(X1, Y1)=(2000, 3000)
D3006=K4000       //Point 2
D3007=K6000       //(X2, Y2)=(4000, 6000)
D3008=K10000      //Point 3
D3009=K7000       //(X3, Y3)=(10000, 7000)

[G10]
PX0               //Standby until PX0 turns ON

[F20]
//Positive conversion execution (Output value is
  stored in D3106)
SCL K0,K7500,D3000,D3106

END
```



- The output value is calculated from the setting value of Point 2 and 3 since the input value "7500" is between point 2 and 3.

$$\text{Output value} = 6000 + \frac{7000 - 6000}{10000 - 4000} \times (7500 - 4000)$$

$$= 6583.333...$$

A fractional portion is rounded down.

D3106 | K6583

F/FS	G
○	○

5.16.2 32-bit integer type scaling: DSCL

Format	DSCL(S1), (S2), (S3), (D)

Number of basic steps	8

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	—	○	—	—	—	○	—	○	—	—
(S3)	—	○	—	—	—	—	—	—	—	—	—
(D)	—	—	○	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S1)	Data which specifies the search/conversion method 0 Positive conversion by sequential search 1: Inverse conversion by sequential search 2: Positive conversion by binary search 3: Inverse conversion by binary search	—
(S2)	Input value for positive/inverse conversion	
(S3)	Start device No. which stores the scaling conversion data	
(D)	Device No. which stores the conversion result	

[Overview]

(1) 32-bit integer type scaling calculates the output value from the set input value based on the scaling conversion data where a maximum of 2000 points data ((X_0, Y_0) to (X_{N-1}, Y_{N-1}), N: the number of points) are defined.
The point data corresponding to the input value should be set in ascending order.
(Positive conversion: $X_0 < X_1 <.....< X_{N-1}$, Inverse conversion: $Y_0 < Y_1 <.....< Y_{N-1}$)

(2) The calculation method for output value is the same as 16-bit integer type scaling. (Refer to Section 5.16.1.)

POINT

When the input value is outside the scaling conversion data or calculation result of output value is outside the range of -2147483648 to 2147483647, an operation error will occur.

[Functions]

(1) Conversion of the input value specified with (S2) is executed according to the search/conversion method specified with (S1), using the scaling conversion data of device (S3) or later. The conversion result is stored in the device specified with (D).

(2) The setting method of (S1) is the same as 16-bit integer type scaling. (Refer to Section 5.16.1.)

(3) The device No. specified with (S3) should be an even number. Set the point data in the specified device as follows.

Off set	Name		Description	Range
+0	Number of points (N)		Set the number of points for the scaling conversion data.	2 to 2000
+1	Unusable		Set 0.	0
+2	Point 0	X_0	Set the point data of (X_0, Y_0) to (X_{N-1}, Y_{N-1}) so that the device No. is in consecutive order.	-2147483648 to 2147483647
+3				
+4		Y_0		
+5				
+6	Point 1	X_1		
+7				
+8		Y_1		
+9				
+10	Point 2	X_2		
+11				
+12		Y_2		
+13				
:	:			
+ (4N-2)	Point (N-1)	X_{N-1}		
+ (4N-1)				
+ (4N)		Y_{N-1}		
+ (4N+1)				

POINT

Set the point data corresponding to the input value in ascending order. (Positive conversion: $X_0 < X_1 < ..... < X_{N-1}$, Inverse conversion: $Y_0 < Y_1 < ..... < Y_{N-1}$)

(4) When the conversion result to be stored in the device specified with (D) is not an integer value, its fractional portion is rounded down.

[Errors]

(1) An operation error will occur, and the conversion of input value will not be executed if:
- (S1) is set to other than 0 to 3.
- (S2), (S3), and (D) are not even-numbered devices.
- The number of points at the point table specified with (S3) is outside the range of 2 to 2000.
- Point table specified with (S3) is outside the device range.
- In sequential search ((S1) is 0 or 1.), the point corresponding to the input value (Positive conversion: X_0 to X_{N-1}, Inverse conversion: Y_0 to Y_{N-1}) is not in ascending order.
- The conversion result is outside the range of -2147483648 to 2147483647.

[Program examples]

(1) Program which sets 4 points of scaling conversion data to D3000 to D3017 and substitutes the output value, which is positively converted based on the input value "-65000", to D3106L.

[Motion SFC program]

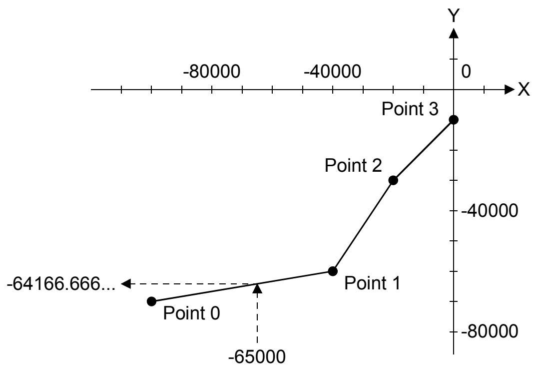
32-bit scaling

```
[F10]
//Scaling conversion data set
D3000=K4            //Number of point=4
D3001=K0            //Unusable area
//Set the point 0 to 3
D3002L=K-100000L    //Point 0
D3004L=K-70000L     //(X0, Y0)=(-100000, -70000)
D3006L=K-40000L     //Point 1
D3008L=K-60000L     //(X1, Y1)=(-40000, -60000)
D3010L=K-20000L     //Point 2
D3012L=K-30000L     //(X2, Y2)=(-20000, -30000)
D3014L=K0L          //Point 3
D3016L=K-10000L     //(X3, Y3)=(0, -10000)

[G10]
PX0                 //Standby until PX0 turns ON

[F20]
//Positive conversion execution (Output value is
  stored in D3106L)
DSCL K0,K-65000,D3000,D3106L
```

END



- The output value is calculated from the setting value of Point 0 and 1 since the input value "-65000" is between Point 0 and 1.

$$\text{Output value} = -70000 + \frac{(-60000 - (-70000))}{(-40000 - (-100000))} \times (-65000 - (-100000))$$

$$= -64166.666...$$

A fractional portion is rounded down.

D3106	D3107
K-64166	

F/FS	G
○	○

5.17 Program Control Ver.!

5.17.1 Conditional branch control: IF - ELSE - IEND

Format	IF(S) - ELSE - IEND

Number of basic steps	IF : 4 ELSE : 3 IEND : 1

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S)	Conditional data which controls the flow of program	—

[Functions]

(1) If the data specified with (S) is true, the block between IF and ELSE is executed.

(2) If the data specified with (S) is false, the block between ELSE and IEND is executed.

(3) ELSE can be omitted. In that case, the block between IF and IEND is executed only when the data specified with (S) is true.

(4) Maximum multiplicities of conditional branch control are eight including selective branch control. (SELECT - CASE - SEND)

```
IF... ─────────────────────────────┐
   IF... ─────────────────────┐    │
      SELECT... ─┐            │    │
         CASE... │            │    │
            ¦    │(3 level)   │(2 level)  │(1 level)
         CEND    │            │    │
      SEND ──────┘            │    │
   IEND ──────────────────────┘    │
IEND ──────────────────────────────┘
```

Ver.! : Refer to Section 1.3 for the software version that supports this function.

[Errors]

(1) In the following case, an operation error will occur, and the corresponding Motion SFC program No. execution will be stopped. For the subroutine called program, the call source program also stops to execute.
   • (S) is indirectly specified device, and the device No. is outside the range.

[Program examples]

(1) Program which adds K10 to #100 when #0 is K100 or adds K20 to #100 when #0 is other than K100.

```
IF #0 == K100
    #100 = #100 + K10
ELSE
    #100 = #100 + K20
IEND
```

(2) Program which executes the speed change of axis 2 with CHGV instruction when M0 or M1 is ON.

```
IF M0 + M1
    CHGV(K2, K10)
IEND
```

F/FS	G
○	○

5.17.2 Selective branch control: SELECT - CASE - SEND

Format	SELECT CASE(S1) - CEND CASE(S2) - CEND : CASE(Sn) - CEND CELSE - CEND SEND

Number of basic steps	SELECT : 1 CASE : 4 CEND : 3 CELSE : 1 SEND : 1

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S) to (Sn)	○	—	—	—	—	—	—	—	—	○	○

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(S) to (Sn)	Conditional data which controls the flow of program	—

[Functions]

(1) The block described between CASE and CEND is executed selectively according to the true/false of the data specified with (S1) to (Sn).

(2) The true/false evaluation is carried out in order from the top, and the block described between CASE which is first evaluated to be true and CEND is executed. After that, no true/false evaluation is carried out until SEND, and the next block of SEND is executed.

(3) When the data specified with (S1) to (Sn) are all false, the block described from CELSE to CEND is executed.

(4) CELSE can be omitted. In that case, if the data specified with (S1) to (Sn) are all false, the block between SELECT and SEND is not executed, and the next block of SEND is executed.

(5) The numbers of CASE(Sn) - CEND that be described between SELECT and SEND are as follows.
- When CELSE is not used : 64
- When CELSE is used : 63

(6) Maximum multiplicities of selective branch control are eight including conditional branch control. (IF - ELSE - IEND)

```
SELECT...
   CASE...
      IF...
         SELECT...
            CASE...
               :                    (3 level) (2 level) (1 level)
            CEND
         SEND
      IEND
   CEND
SEND
```

[Errors]

(1) In the following case, an operation error will occur, and the corresponding Motion SFC program No. execution will be stopped. For the subroutine called program, the call source program also stops to execute.
- (S) is indirectly specified device, and the device No. is outside the range.

[Program examples]

(1) Program which adds K10 to #100 when #0 is K100, adds K20 to #100 when #0 is K200 or more, or adds K100 to #100 in other cases.

```
SELECT
    CASE #0 == K100
        #100 = #100 + K10
    CEND
    CASE #0 >= K200
        #100 = #100 + K20
    CEND
    CELSE
        #100 = #100 + K100
    CEND
SEND
```

F/FS	G
○	○

5.17.3 Repeat control with specified count: FOR - NEXT

Format	FOR(D) = (S1)TO(S2)STEP(S3) - NEXT	Number of basic steps	FOR : 9 NEXT : 8

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(D)	—	○	○	○	—	—	—	—	—	—	—
(S1)	—	○	○	○	—	○	○	○	—	—	—
(S2)	—	○	○	○	—	○	○	○	—	—	—
(S3)	—	—	—	—	—	○	○	○	—	—	—

○ : Usable

[Setting data]

Setting data	Description	Data type of result
(D)	Device used for loop control counter	—
(S1)	Initial value of loop control counter	
(S2)	Final value of loop control counter	
(S3)	Incremental value of loop control counter	

[Functions]

(1) (S1) is substituted to the device specified with (D) as initial value, and the block between FOR and NEXT is repeatedly executed.

(2) The incremental value specified with (S3) is added to the device specified with (D) at every execution of NEXT. If the device value specified with (D) is larger than the final value specified with (S2), the repeat control of the block between FOR and NEXT is ended, and the next block of NEXT is executed.

(3) When the incremental value specified with (S3) is negative number, if the device value specified with (D) is smaller than the final value specified with (S2), the repeat control of the block between FOR and NEXT is ended.

(4) STEP can be omitted. If STEP is omitted, the repeat control is executed as "STEP 1".

(5) Maximum multiplicities of repeat control are eight.

```
FOR...
    FOR...
       :                (2 level) (1 level)
    NEXT
NEXT
```

(6) When data types of (D), (S1), (S2) and (S3) are different, type conversion processing is executed but an unintended operation may occur. Set the same data type.

[Errors]

(1) In the following case, an operation error will occur, and the corresponding Motion SFC program No. execution will be stopped. For the subroutine called program, the call source program also stops to execute.
- (S1) data is outside the range of (D) data type.
- (D), (S1), and (S2) are indirectly specified devices, and the device No. is outside the range.
- FOR to NEXT instruction is executed over the limited count for repeat control set in parameter in an operation control program or a transition program.

[Program examples]

(1) Program which repeats to substitute #0 data to Motion register (#) that is indirectly specified with the device No. "#0+100" when #0 is between 1 and 10 (Incremental value is 1.).
(When the program is ended, 1 to 10 is substituted to #101 to #110.)

```
FOR #0 = K1 TO K10
    #(#0 + K100) = #0
NEXT
```

When the incremental value is positive number, the device value specified with (D) is larger than the final value specified with (S2) after FOR to NEXT repeat is completed. In the above example, #0 set in (D) is 11.

(2) Program which repeats to subtract #0 from #100 when #0 is between 100 to 10 (Incremental value is -10.).

```
FOR #0 = K100 TO K10 STEP K-10
    #100 = #100 - #0
NEXT
```

When the incremental value is negative number, the device value specified with (D) is smaller than the final value specified with (S2) after FOR to NEXT repeat is completed. In the above example, #0 set in (D) is 0.

POINT

Since the incremental value continues to be added to the loop control counter specified with (D) until it reaches the final value, set the data type which the value can handle.
When the data range exceeds the loop control counter range, an unintended repeat operation may occur as the value is considered to be wrong.
In the following program, the data type of loop control counter #0 is 16-bit integer type, and the data range is from -32768 to 32767.

```
FOR #0 = K0 TO K30000 STEP K10000
    #1 = #1 + K1
NEXT
```

When this program is executed, #0 changes as follows and exceeds the 16-bit integer type data range in the middle. Therefore, the program is not ended with four executions of the loop.

- First execution of the loop : #0 is 0.
- Second execution of the loop : #0 is 10000.
- Third execution of the loop : #0 is 20000.
- Forth execution of the loop : #0 is 30000.
- Fifth execution of the loop : #0 is -25536.
  (Note): #0 is 40000, but overflow will occur as it is outside the data range.
- Sixth execution of the loop : #0 is -15536.

:

F/FS	G
○	○

5.17.4 Forced termination of repeat control: BREAK

Format	BREAK	Number of basic steps	3

[Usable data]

Setting data	Usable Data										
	Bit device	Word device				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S)	—	—	—	—	—	—	—	—	—	—	—

○ : Usable

[Setting data]

There are no setting data.

[Functions]

(1) Repeat control with specified count (FOR - NEXT instruction) is forced to terminate, and the program from the next block of NEXT is executed.

(2) BREAK is only described within the repeat control processing block between FOR and NEXT.

[Errors]

(1) There are no operation errors.

[Program examples]

(1) Program which forces to terminate the repeat control processing by FOR to NEXT when M0 or M1 turns ON.

```
FOR #0 = K1 TO K10
   #100 = #100 + K10
   IF M0 + M1
      BREAK
   IEND
NEXT
```

F/FS	G
○	○

5.18 Synchronous Control Dedicated Function (SV22 advanced synchronous control only) QDS Ver.!

5.18.1 Cam data read: CAMRD

Format	CAMRD(S1), (S2), (n), (D)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—
(n)	—	○	—	—	—	○	—	—	○	—	—
(D)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Cam No. (1 to 256)	—
(S2)	Cam data first position Stroke ratio data format : 1 to cam resolution Coordinate data format : 0 to (Coordinate number-1)	
(n)	Number of cam data points Stroke ratio data format: 1 to 4096 Coordinate data format : 1 to 2048	
(D)	Start device No. which stores the reading cam data	

[Functions]

(1) Of the cam No. data specified with (S1) in the cam open area, the data of the (n) number of points, starting from the position specified with (S2), is read. The read cam data is stored in the device specified with (D) or later.

(2) Set the cam data first position specified with (S2) within the following range.
- Stroke ratio data format : 1 to cam resolution (Note-1)
- Coordinate data format : 0 to (Coordinate number-1)

(Note-1): Since the stroke ratio of the zeroth point cam data is fixed at 0%, the cam data cannot be read.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(3) Specify the number of read points for (n). Specify the number of read points so that the device No. storing the end point data is within the range. The following shows the operation when the number of read points, starting from the first position, is outside the cam data range.
- Stroke ratio data format : If the value calculated from "(S2) + (n) - 1" is larger than the cam resolution, the cam data, ranging from the cam data first position to the cam resolution, is read.
- Coordinate data format : If the value calculated from "(S2) + (n)" is larger than the number of coordinates, the cam data, ranging from the cam data first position to the last coordinate, is read.

(4) The device No. specified with (D) should be an even number. The read cam data is stored in the specified device according to the cam data format as follows.
(a) Stroke ratio data format

Off set	Item	Range
+0	Can data format (Stroke ratio data format)	1
+1	Cam data starting point	0 to (Coordinate resolution-1)
+2	Cam resolution	256/512/1024/2048/4096/ 8192/16384/32768
+3		
+4	Stroke ratio at first point cam data value	-2147483648 to 2147483647[$\times 10^{-7}$%] (-214.7483648 to 214.7483647[%])
+5		
+6	Stroke ratio at second point cam data value	
+7		
:	:	
+ (2N+2)	Stroke ratio at Nth point cam data value	
+ (2N+3)		

(b) Coordinate data format

Off set	Item		Range
+0	Can data format (Coordinate data format)		2
+1	Unusable		0
+2	Coordinate number		2 to 16384
+3			
+4	At first point cam data value	Input value X_1	0 to 2147483647
+5			[Cam axis cycle unit]
+6		Output value Y_1	-2147483648 to 2147483647
+7			[Output axis position unit]
+8	At second point cam data value	Input value X_2	0 to 2147483647
+9			[Cam axis cycle unit]
+10		Output value Y_2	-2147483648 to 2147483647
+11			[Output axis position unit]
:	:		:
+ (4N)	At Nth point cam data value	Input value X_N	0 to 2147483647
+ (4N+1)			[Cam axis cycle unit]
+ (4N+2)		Output value Y_N	-2147483648 to 2147483647
+ (4N+3)			[Output axis position unit]

POINT

For coordinate data format, when reading is not completed in one attempt, divide the reading over several attempts.

[Errors]

(1) An operation error will occur, and the cam data read will not be executed if:
- Cam No. specified with (S1) is outside the range of 1 to 256.
- The cam No. data specified with (S1) does not exist in the cam open area.
- For the cam data in the stroke ratio data format, the cam data first position specified with (S2) is outside the range of 1 to the cam resolution.
- For the cam data in the coordinate data format, the cam data first position specified with (S2) is outside the range of 0 to (coordinate number - 1).
- For the cam data in the stroke ratio data format, the number of cam data points is outside the range of 1 to 4096.
- For the cam data in the coordinate data format, the number of cam data points is outside the range of 1 to 2048.
- The device numbers storing the number of cam data points specified with (n) are outside the range.
- (D) are not even-numbered devices.
- The cam data was read with "Read/write protection" password set.

[Program examples]

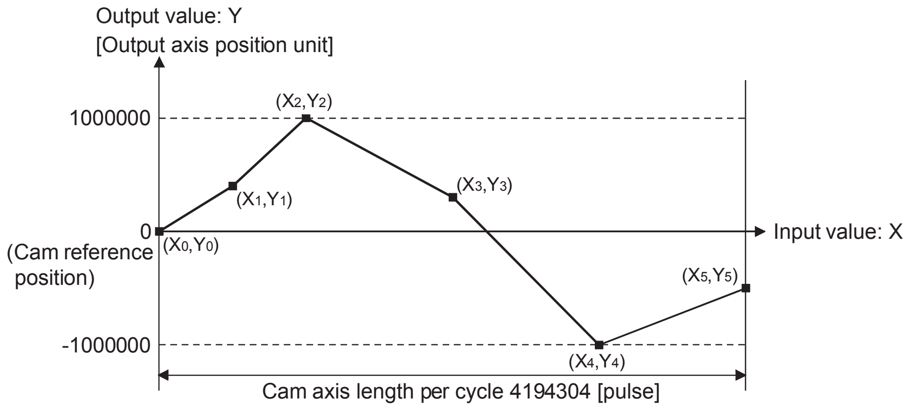
(1) Program which reads 2048-points of data, starting from the first point cam data of cam No. 2 (stroke ratio data format), and stores the read data to #0 to #4099.

```
CAMRD K2,K1,K2048, #0
```

(2) Program which reads 6-points of data, starting from the zeroth point cam data of cam No.1 (coordinate data format), and stores the read data to #100 to #127.

```
CAMRD K1,K0,K6, #100
```

(a) Cam data No.1



(b) Read cam data

```
#100=K2             //Coordinate data format
#102L=K6            //Number of coordinates 6
#104L=K0            //(1st point) input value
#106L=K0            //(1st point) output value
#108L=K524288       //(2nd point) input value
#110L=K400000       //(2nd point) output value
#112L=K1048576      //(3rd point) input value
#114L=K1000000      //(3rd point) output value
#116L=K2097152      //(4th point) input value
#118L=K300000       //(4th point) output value
#120L=K3145728      //(5th point) input value
#122L=K-1000000     //(5th point) output value
#124L=K4194304      //(6th point) input value
#126L=K-500000      //(6th point) output value
```

F/FS	G
○	○

5.18.2 Cam data write: CAMWR

Format	CAMWR(S1), (S2), (n), (S3)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—
(n)	—	○	—	—	—	○	—	—	○	—	—
(S3)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Cam No. (1 to 256)	—
(S2)	Cam data first position Stroke ratio data format : 1 to cam resolution Coordinate data format : 0 to (Coordinate number-1)	
(n)	Number of cam data points Stroke ratio data format: 1 to 4096 Coordinate data format : 1 to 2048	
(S3)	Start device No. which stores the writing cam data	

[Functions]

(1) Of the cam data stored in the device specified with (S3) or later, the data of the (n) number of points, starting from the cam data position specified with (S2), is written to the cam storage area and the cam open area.

(2) Set the cam data first position specified with (S2) within the following range.
- Stroke ratio data format : 1 to cam resolution (Note-1)
- Coordinate data format : 0 to (Coordinate number-1)

(Note-1): Since the stroke ratio of the zeroth point cam data is fixed at 0%, the cam data cannot be write.

(3) For (n), specify the number of write points, starting from the cam data first position specified with (S2). Specify the number of write points so that the device No. storing the end point data is within the range. If the number of write points, starting from the first position, is outside the cam data range, an operation error occurs and the data is not written.

(4) The device No. specified with (S3) should be an even number. The write cam data is stored in the specified device according to the cam data format as follows.

(a) Stroke ratio data format

Off set	Item	Range
+0	Can data format (Stroke ratio data format)	1
+1	Cam data starting point	0 to (Coordinate resolution-1)
+2 +3	Cam resolution	256/512/1024/2048/4096/ 8192/16384/32768
+4 +5	Stroke ratio at first point cam data value	-2147483648 to 2147483647[$\times 10^{-7}$%] (-214.7483648 to 214.7483647[%])
+6 +7	Stroke ratio at second point cam data value	
:	:	
+ (2N+2) + (2N+3)	Stroke ratio at Nth point cam data value	

(b) Coordinate data format

Off set	Item		Range
+0	Can data format (Coordinate data format)		2
+1	Unusable		0
+2 +3	Coordinate number		2 to 16384
+4 +5	At first point cam data value	Input value X_1	0 to 2147483647 [Cam axis cycle unit]
+6 +7		Output value Y_1	-2147483648 to 2147483647 [Output axis position unit]
+8 +9	At second point cam data value	Input value X_2	0 to 2147483647 [Cam axis cycle unit]
+10 +11		Output value Y_2	-2147483648 to 2147483647 [Output axis position unit]
:	:		:
+ (4N) + (4N+1)	At Nth point cam data value	Input value X_N	0 to 2147483647 [Cam axis cycle unit]
+ (4N+2) + (4N+3)		Output value Y_N	-2147483648 to 2147483647 [Output axis position unit]

POINT

For coordinate data format, when writing is not completed in one attempt, divide the writing over several attempts.

(5) During the execution of the CAMWR instruction, another CAMWR instruction, CAMWR2 instruction, or CAMMK instruction cannot be processed.
During the execution of the CAMWR instruction, the cam data writing flag (SM505) turns on. Therefore, create an interlock. When the CAMWR instruction, CAMWR2 instruction, or CAMMK instruction is executed while the cam data writing flag (SM505) is on, an error occurs.

POINT

The CAMWR instruction can be executed during the synchronous control. Note that the contents of the cam data in operation are changed depending on the instruction execution timing.

[Errors]

(1) An operation error will occur, and the cam data write will not be executed if:
- Cam No. specified with (S1) is outside the range of 1 to 256.
- For the cam data in the stroke ratio data format, the cam data first position specified with (S2) is outside the range of 1 to the cam resolution.
- For the cam data in the coordinate data format, the cam data first position specified with (S2) is outside the range of 0 to (coordinate number - 1).
- For the cam data in the stroke ratio data format, the number of cam data points is outside the range of 1 to 4096.
- For the cam data in the coordinate data format, the number of cam data points is outside the range of 1 to 2048.
- The start position and the number of cam data points, which are outside the range of the cam resolution or the number of coordinates, are set.
- The device numbers storing the number of cam data points specified with (n) are outside the range.
- (S3) are not even-numbered devices.
- Cam data format specified with (S3) is set to other than 1 or 2.
- For the cam data in the stroke ratio data format, the cam resolution is set a value other than "256/512/1024/2048/4096/8192/16384/32768".
- For the cam data in the coordinate data format, the coordinate number is set a value other than "2 to 16384".
- For the cam data in the stroke ratio data format, the cam data first position is outside the range of 0 to (cam resolution - 1).
- The writable area is insufficient when the cam data is being written.
- The input value of the coordinate data is a negative value.
- The input value of the coordinate data satisfies "$X_n > X_{n+1}$".
- The cam data was read with "Write protection" or "Read/write protection" password set.
- The CAMWR instruction is executed during the cam data writing (CAMWR instruction, CAMWR2 instruction, CAMMK instruction).

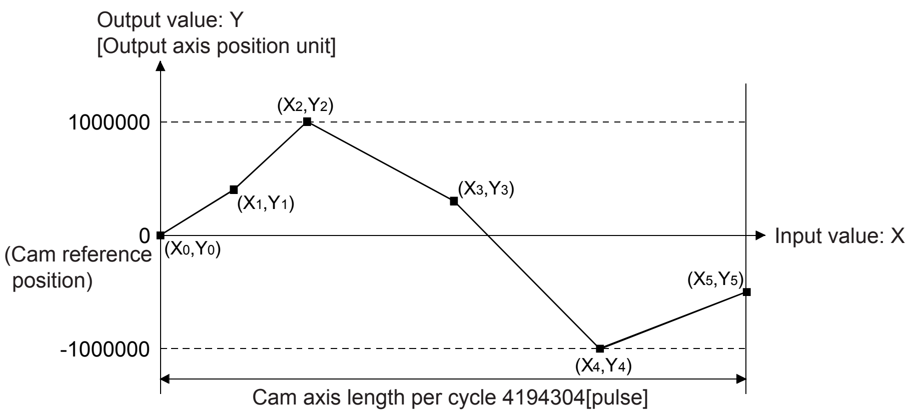
[Program examples]

(1) Program which writes the data stored in #0 to #4099 to the 2048-point area, starting from the first point cam data, of cam No. 256 (stroke ratio data format)

```
CAMWR K256,K1,K2048,#0
```

(2) Program (Cam axis length per cycle = 4194304) which writes the data stored in #0 to #27 to the 6-point area, starting from the zeroth point cam data, of cam No. 255 (coordinate data format)

```
#0=K2                   //Coordinate data format
#2L=K6                  //Number of coordinates 6
#4L=K0                  //(1st point) input value
#6L=K0                  //(1st point) output value
#8L=K524288             //(2nd point) input value
#10L=K400000            //(2nd point) output value
#12L=K1048576           //(3rd point) input value
#14L=K1000000           //(3rd point) output value
#16L=K2097152           //(4th point) input value
#18L=K300000            //(4th point) output value
#20L=K3145728           //(5th point) input value
#22L=K-1000000          //(5th point) output value
#24L=K4194304           //(6th point) input value
#26L=K-500000           //(6th point) output value
CAMWR K255,K0,K6,#0     //Cam data write
```



F/FS	G
○	○

5.18.3 Cam data write (Cam open area): CAMWR2

Format	CAMWR2(S1), (S2), (n), (S3)

Number of basic steps	7

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—
(n)	—	○	—	—	—	○	—	—	○	—	—
(S3)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Cam No. (1 to 256)	—
(S2)	Cam data first position Stroke ratio data format : 1 to cam resolution Coordinate data format : 0 to (Coordinate number-1)	
(n)	Number of cam data points Stroke ratio data format: 1 to 4096 Coordinate data format : 1 to 2048	
(S3)	Start device No. which stores the writing cam data	

[Functions]

(1) Of the cam data stored in the device specified with (S3) or later, the data of the (n) number of points, starting from the cam data position specified with (S2), is written to the cam open area.

(2) Set the cam data first position specified with (S2) within the following range.
- Stroke ratio data format : 1 to cam resolution (Note-1)
- Coordinate data format : 0 to (Coordinate number-1)

(Note-1): Since the stroke ratio of the zeroth point cam data is fixed at 0%, the cam data cannot be write.

(3) For (n), specify the number of write points, starting from the cam data first position specified with (S2). Specify the number of write points so that the device No. storing the end point data is within the range. If the number of write points, starting from the first position, is outside the cam data range, an operation error occurs and the data is not written.

(4) The device No. specified with (S3) should be an even number. The write cam data is stored in the specified device according to the cam data format as follows.

(a) Stroke ratio data format

Off set	Item	Range
+0	Can data format (Stroke ratio data format)	1
+1	Cam data starting point	0 to (Coordinate resolution-1)
+2	Cam resolution	256/512/1024/2048/4096/8192/16384/32768
+3		
+4	Stroke ratio at first point cam data value	-2147483648 to 2147483647[$\times 10^{-7}$%] (-214.7483648 to 214.7483647[%])
+5		
+6	Stroke ratio at second point cam data value	
+7		
:	:	
+ (2N+2)	Stroke ratio at Nth point cam data value	
+ (2N+3)		

(b) Coordinate data format

Off set	Item		Range
+0	Can data format (Stroke ratio data format)		2
+1	Cam data starting point		0
+2	Cam resolution		2 to 16384
+3			
+4	at first point cam data value	Input value X_1	0 to 2147483647 [Cam axis cycle unit]
+5			
+6		Output value Y_1	-2147483648 to 2147483647 [Output axis position unit]
+7			
+8	at second point cam data value	Input value X_2	0 to 2147483647 [Cam axis cycle unit]
+9			
+10		Output value Y_2	-2147483648 to 2147483647 [Output axis position unit]
+11			
:	:		:
+ (4N)	At Nth point cam data value	Input value X_N	0 to 2147483647 [Cam axis cycle unit]
+ (4N+1)			
+ (4N+2)		Output value Y_N	-2147483648 to 2147483647 [Output axis position unit]
+ (4N+3)			

(5) During the execution of the CAMWR2 instruction, another CAMWR instruction, CAMWR2 instruction, or CAMMK instruction cannot be processed.
During the execution of the CAMWR2 instruction, the cam data writing flag (SM505) turns on. Therefore, create an interlock. When the CAMWR instruction, CAMWR2 instruction, or CAMMK instruction is executed while the cam data writing flag (SM505) is on, an error occurs.

POINT

The CAMWR2 instruction can be executed during the synchronous control. Note that the contents of the cam data in operation are changed depending on the instruction execution timing.

[Errors]

(1) An operation error will occur, and the cam data write will not be executed if:
- Cam No. specified with (S1) is outside the range of 1 to 256.
- For the cam data in the stroke ratio data format, the cam data first position specified with (S2) is outside the range of 1 to the cam resolution.
- For the cam data in the coordinate data format, the cam data first position specified with (S2) is outside the range of 0 to (coordinate number - 1).
- For the cam data in the stroke ratio data format, the number of cam data points is outside the range of 1 to 4096.
- For the cam data in the coordinate data format, the number of cam data points is outside the range of 1 to 2048.
- The start position and the number of cam data points, which are outside the range of the cam resolution or the number of coordinates, are set.
- The device numbers storing the number of cam data points specified with (n) are outside the range.
- (S3) are not even-numbered devices.
- Cam data format specified with (S3) is set to other than 1 or 2.
- For the cam data in the stroke ratio data format, the cam resolution is set a value other than "256/512/1024/2048/4096/8192/16384/32768".
- For the cam data in the coordinate data format, the coordinate number is set a value other than "2 to 16384".
- For the cam data in the stroke ratio data format, the cam data first position is outside the range of 0 to (cam resolution - 1).
- The writable area is insufficient when the cam data is being written.
- The input value of the coordinate data is a negative value.
- The input value of the coordinate data satisfies "$X_n > X_{n+1}$".
- The cam data was read with "Write protection" or "Read/write protection" password set.
- The CAMWR2 instruction is executed during the cam data writing (CAMWR instruction, CAMWR2 instruction, CAMMK instruction).

[Program examples]

(1) Program which writes the data stored in #2048 to #6147 to the 2048-point area, starting from the 2049-point cam data, of cam No. 10 (stroke ratio data format)

```
CAMWR2 K10,K2049,K2048, #2048
```

F/FS	G
○	○

5.18.4 Cam auto-generation: CAMMK

Format	CAMMK(S1), (S2), (S3)

Number of basic steps	6

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(S3)	—	○	—	—	—	—	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Cam No. (1 to 256)	—
(S2)	Cam auto-generation type Cam for rotary cutter : 1 Easy stroke ratio cam: 2	
(S3)	Start device No. which stores the auto-generation data	

[Functions]

(1) The auto-generation cam No. data specified with (S1) is created in the cam open area based on the cam auto-generation type specified with (S2), and the auto-generation data to be stored in the device specified with (S3). The cam auto-generation data is stored in the cam storage area. The cam auto-generation will be automatically executed at next power supply ON of the Multiple CPU system or at next OFF to ON of the PLC ready flag (M2000).

(2) Specify the following cam auto-generation type with (S2).
• Cam for rotary cutter : 1
• Easy stroke ratio cam : 2

(3) For (S3), set the auto-generation data for the cam auto-generation type specified with (S2). The specified device No. should be an even number. Assign the auto-generation data to the specified device or later. The device No. storing the end point data must be within the range.

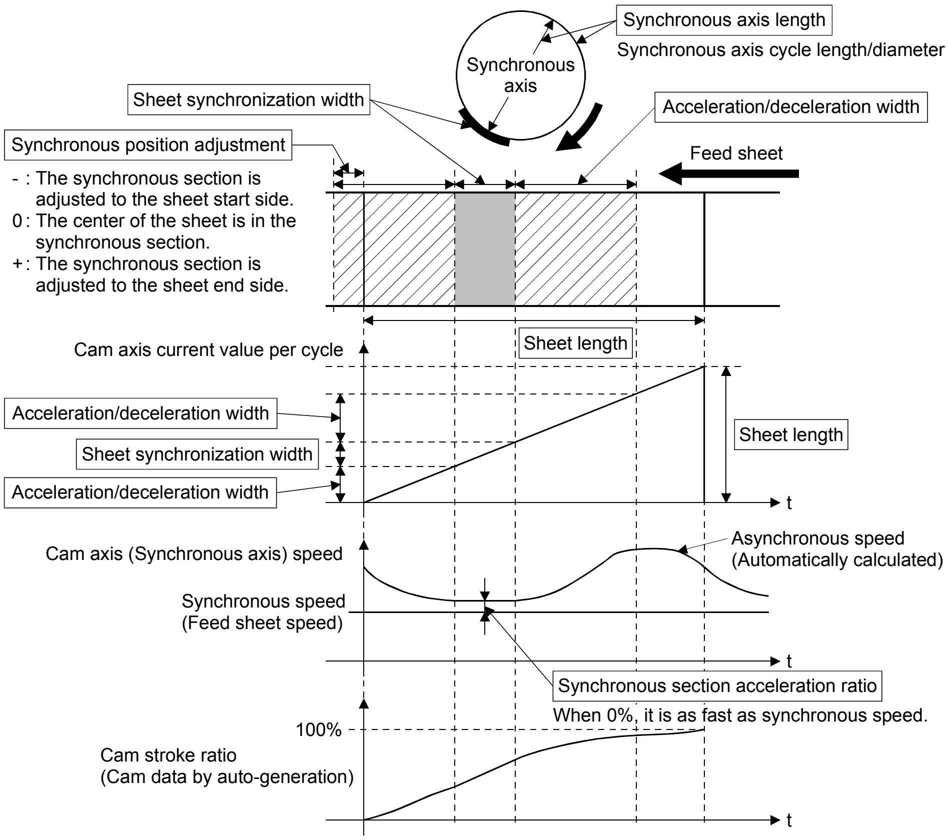
(4) During the execution of the CAMMK instruction, another CAMWR instruction, CAMWR2 instruction or CAMMK instruction cannot be processed.
During the execution of the CAMMK instruction, the cam data writing flag (SM505) turns on. Therefore, create an interlock. When the CAMWR instruction, CAMWR2 instruction or CAMMK instruction is executed while the cam data writing flag (SM505) is ON, an error occurs.

[Errors]

(1) An operation error will occur, and the cam auto-generation will not be executed if:
- Cam No. specified with (S1) is outside the range of 1 to 256.
- Auto-generation type specified with (S2) is set to a value that does not correspond to an auto-generation type.
- The device numbers storing the auto-generation data specified with (S3) are outside the range.
- (S3) are not even-numbered devices.
- The writable area is insufficient when the cam data is being written.
- A value outside the range is set for the auto-generation data.
- For the cam for rotary cutter, a value has been set as "sheet synchronization width ≥ sheet length" in the auto-generation parameter.
- For the cam for rotary cutter, the asynchronous speed will be reduced when the auto-generation data is set as "synchronous axis length (synchronous axis diameter × π") < sheet length".
- For the cam for rotary cutter, the asynchronous speed is 655.35 times of larger than synchronous speed by auto-generation data.
- The cam auto-generation was performed with the "Write protection" or "Read/write protection" password set in the cam data.
- The CAMMK instruction is executed during the cam data writing (CAMWR instruction, CAMWR2 instruction, CAMMK instruction).
- For the easy stroke ratio cam, the end point set for each section are not in ascending order.
- For the easy stroke ratio cam, the end point of the final section is less than the cam axis length per cycle.

[Cam for rotary cutter]

(1) Set the auto-generation data of the rotary cam cutter. (sheet length, synchronization width, etc.)



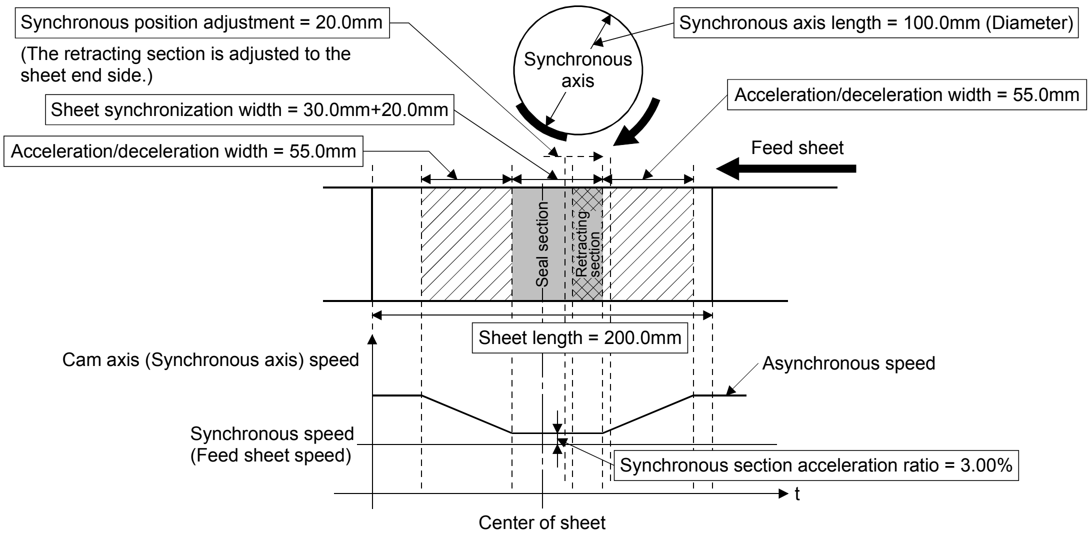
(2) Device assignment of the cam auto-generation data for the rotary cutter cam
When the synchronous position adjustment is set to 0, the cam pattern of which the sheet center is in the synchronous section is created.

Off set	Name	Description	Range
+0 +1	Resolution	Set the cam resolution for generating the cam.	256/512/1024/2048/4096/ 8192/16384/32768
+2	Auto-generation option	• Select the trapezoidal acceleration/deceleration system or the S-curve acceleration/deceleration system with bit 0. • Select the diameter or the cycle length for the synchronous axis length with bit 1. • Set 0 for bits 2 to 15.	• Bit 0........ Acceleration/deceleration system 0: Trapezoidal acceleration/ deceleration 1: S-curve acceleration/ deceleration • Bit 1........ Synchronous axis length setting 0: Diameter 1: Cycle length
+3	Synchronous section acceleration ratio	Set when the synchronous speed in the synchronous section needs to be adjusted. The speed is "Synchronous speed × (100% + Acceleration ratio)" in the synchronous section.	-5000 to 5000[0.01%]
+4 +5	Sheet length	Set the sheet length.	1 to 2147483647 [(Optional) Same units]
+6 +7	Sheet synchronization width	• Set the sheet synchronization width (seal width). • When the synchronous speed section for retracting is required in front of and behind the sheet synchronization width, add the retracting width.	1 to 2147483647 [(Optional) Same units]
+8 +9	Synchronous axis length	• Set the rotary cutter axis length. • When the synchronous axis length of the auto-generation option is set to the diameter, "Cycle length = setting value × π". • When the synchronous axis length of the auto-generation option is set to the cycle length, "Cycle length = setting value".	For diameter setting 1 to 680000000 For cycle length setting 1 to 2147483647 [(Optional) Same units]
+10 +11	Synchronous position adjustment	• Set the position adjustment of the synchronous section. - : The synchronous section is adjusted to the sheet start side. 0 : The center of the sheet is in the synchronous section. + : The synchronous section is adjusted to the sheet end side. • Set the value within one-half of the sheet length.	-1073741823 to 1073741823 [(Optional) Same units]
+12 +13	Acceleration/ deceleration width	• Set the sheet width (one side) of the acceleration/deceleration area. • When a negative value is set, the acceleration/deceleration width is determined to be the maximum.	0 to 2147483647 [(Optional) Same units] (Note): For a value other than the above, the acceleration/deceleration width is determined to be the maximum.
+14	Number of cutter	Set the number of cutter.	1 to 256
+15	Asynchronous speed result	When the cam auto-generation is successfully performed, the asynchronous speed is stored as the ratio to the synchronous speed.	0 to 65535[0.01 times]

(3) Program examples

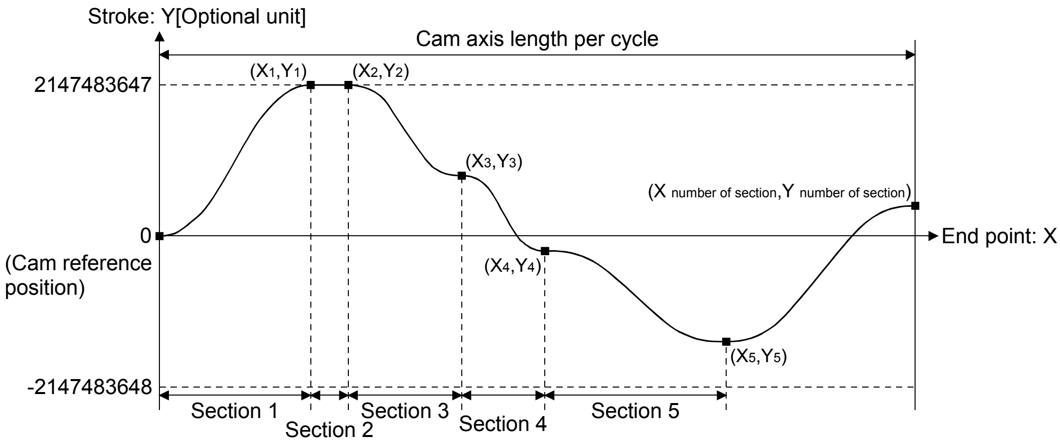
(a) Program which creates cam data (resolution: 512) for the rotary cutter operation pattern in Cam No.5.

```
D5000L=K512           // Resolution = 512
D5002=K0              // Acceleration/deceleration system = Trapezoidal,
                         Synchronous axis length setting = Diameter
D5003=K300            // Synchronous section acceleration ratio = 3.00%
D5004L=K2000          // Sheet length = 200.0mm
D5006L=K500           // Sheet synchronization width = 30.0mm (Seal section) +
                      20.0mm (Retracting operation)
D5008L=K1000          // Synchronous axis length = 100.0mm(Diameter)
D5010L=K200           // Synchronous position adjustment = 20.0mm
D5012L=K550           // Acceleration/deceleration width = 55.0mm
D5014=K1              // Number of cutter = 1
CAMMK K5,K1,D5000     // Cam auto-generation (Asynchronous speed result is stored
                         in D5015.)
```



[Easy stroke ratio cam]

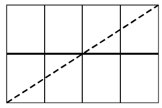
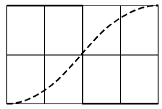
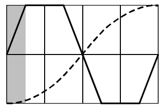
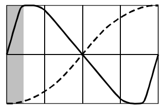
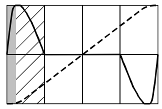
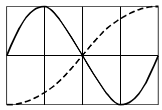
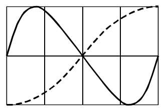
(1) Cam data can be automatically generated without using the cam data setting of MT Developer2 by setting the stroke amount and sections.
With the current value per cycle "0" as starting point, automatically generates cam data from the stroke and cam curve type of each section until the specified end point (cam axis current value per cycle).



(2) Device assignment of the cam auto-generation data for the easy stroke ratio cam

Off set	Name		Description	Range
+0 +1	Resolution		Set the cam resolution for generating the cam.	256/512/1024/2048/4096/ 8192/16384/32768
+2 +3	Cam axis length per cycle		Set the cycle length of one cam operation cycle.	1 to 2147483647 [Cam axis length per cycle units]
+4 +5	Cam data starting point		Set the starting point as the point corresponding to "cycle length=0" of cam data.	0 to (Resolution - 1)
+6	Number of sections		Set the number of sections of cam data. Set data for the number of sections specified.	1 to 32
+7	Unusable		Set 0.	0
+8	Section 1	Cam curve type (Note-2)	Set the cam curve.	0: Constant speed 5: Cycloid 1: Constant acceleration 6: 5th curve 2: Distorted trapezoid 3: Distorted sine 4: Distorted constant speed
+9		Unusable	Set 0.	0
+10 +11		End point (X_1)	Set the point for cam axis length per cycle (cam axis current value per cycle). It is necessary to set a value larger than the end point immediately before (Xn<Xn+1). Also, for the final end point, set as the cam axis length per cycle.	1 to Cam axis length per cycle [Cam axis length per cycle units] (Note-1)
+12 +13		Stroke (Y_1)	Set the stroke position from the cam reference position of when at the end point specified by cam axis current value per cycle. When set at 1000000000, it becomes the position set in [Pr.441] Cam stroke amount (D15064+150n, D15065+150n).	-2147483648 to 2147483647 [Optional units]
+14	Section 2	Cam curve type (Note-2)	The data specified by "number of sections" becomes valid. It is not necessary to set the data after the specified number of sections.	0: Constant speed 5: Cycloid 1: Constant acceleration 6: 5th curve 2: Distorted trapezoid 3: Distorted sine 4: Distorted constant speed
+15		Unusable		0
+16 +17		End point (X_2)		1 to Cam axis length per cycle [Cam axis length per cycle units] (Note-1)
+18 +19		Stroke (Y_2)		-2147483648 to 2147483647 [Optional units]
:	:			:
+194	Section 32	Cam curve type (Note-2)		0: Constant speed 5: Cycloid 1: Constant acceleration 6: 5th curve 2: Distorted trapezoid 3: Distorted sine 4: Distorted constant speed
+195		Unusable		0
+196 +197		End point (X_{32})		1 to Cam axis length per cycle [Cam axis length per cycle units] (Note-1)
+198 +199		Stroke (Y_{32})		-2147483648 to 2147483647 [Optional units]

(Note-1): If setting is outside range, the cam axis length per cycle will be set as the final end point of the section settings.
(Note-2): The types of cam curve shapes are shown below.
Create the cam curves using the values below.

Cam curve type		Acceleration curve shape —— : Stroke ratio ▨ : Range L1 ----- : acceleratrion ▧ : Range L2	Curve applicable range (P1 to P2)	Acceleration/deceleration range compensation	
Setting value	Cam curve name			Range L1	Range L2
0	Constant speed	 P1 P2	0.00 to 1.00	—	—
1	Constant acceleration	 P1 P2	0.00 to 1.00	—	—
2	Distorted trapezoid	 P1 P2	0.00 to 1.00	0.1250	—
3	Distorted sine	 P1 P2	0.00 to 1.00	0.1250	—
4	Distorted constant speed	 P1 P2	0.00 to 1.00	0.0625	0.2500
5	Cycloid	 P1 P2	0.00 to 1.00	—	—
6	5th curve	 P1 P2	0.00 to 1.00	—	—

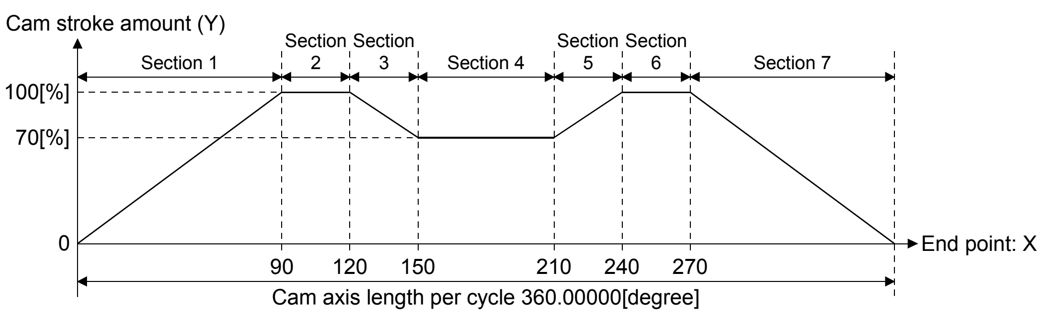
POINT

(1) Set data for the number of sections specified. It is not necessary to set the data after the number of sections specified.
(2) Set the end point data in ascending order.
(3) Various cam patterns are created by the setting of the stroke and cam data of each section. If the amount of change in stroke is large, it may cause a servo error in the servo amplifier including overspeed, data error etc. When creating cam, confirm the cam operation in amplifier-less operation.
(4) Cannot set detailed settings of cam curve including curve applicable range, etc. like cam data setting of MT Developer2.
(5) Cam data will end at the section where the end point is exceeds the cam axis length per cycle set by the auto-generation data.

(3) Program that creates easy stroke ratio cam data

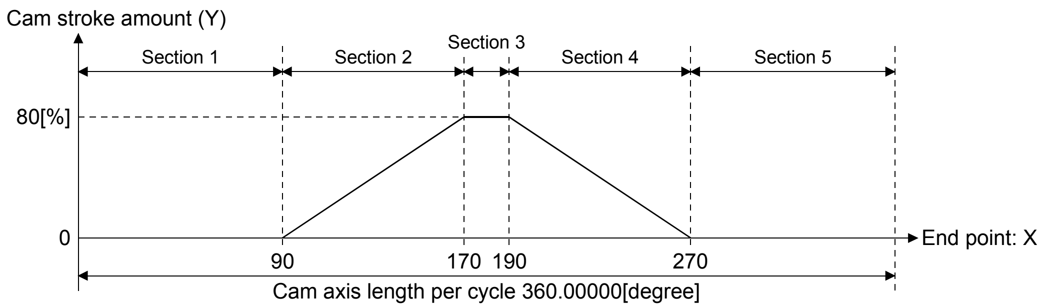
(a) Program which creates cam data (resolution: 512) in cam No. 5.

```
D5000L=K512              //Resolution=512
D5002L=K36000000         //Cam axis length per cycle=360.0[degree]
D5004L=K0                //Cam data starting point=0th point
D5006=K7                 //Number of sections=7 sections
D5007=K0                 //Unusable
D5008=K0                 //(Section 1) Cam curve type=Constant speed
D5009=K0                 //Unusable
D5010L=K9000000          //(Section 1) End point (X1)=90.0[degree]
D5012L=K1000000000       //(Section 1) Stroke (Y1)=100.0[%]
D5014=K0                 //(Section 2) Cam curve type =Constant speed
D5015=K0                 //Unusable
D5016L=K12000000         //(Section 2) End point (X2)=120.0[degree]
D5018L=K1000000000       //(Section 2) Stroke (Y2)=100.0[%]
D5020=K0                 //(Section 3) Cam curve type =Constant speed
D5021=K0                 //Unusable
D5022L=K15000000         //(Section 3) End point (X3)=150.0[degree]
D5024L=K700000000        //(Section 3) Stroke (Y3)=70.0[%]
D5026=K0                 //(Section 4) Cam curve type =Constant speed
D5027=K0                 //Unusable
D5028L=K21000000         //(Section 4) End point (X4)=210.0[degree]
D5030L=K700000000        //(Section 4) Stroke (Y4)=70.0[%]
D5032=K0                 //(Section 5) Cam curve type =Constant speed
D5033=K0                 //Unusable
D5034L=K24000000         //(Section 5) End point (X5)=240.0[degree]
D5036L=K1000000000       //(Section 5) Stroke (Y5)=100.0[%]
D5038=K0                 //(Section 6) Cam curve type =Constant speed
D5039=K0                 //Unusable
D5040L=K27000000         //(Section 6) End point (X6)=270.0[degree]
D5042L=K1000000000       //(Section 6) Stroke (Y6)=100.0[%]
D5044=K0                 //(Section 7) Cam curve type =Constant speed
D5045=K0                 //Unusable
D5046L=K36000000         //(Section 7) End point (X7)=360.0[degree]
D5048L=K0                //(Section 7) Stroke (Y7)=0[%]
CAMMK K5,K2,D5000        //Cam auto-generation
```



(b) Program which creates cam data (resolution: 512) in cam No. 6

```
D6000L=K512           //Resolution=512
D6002L=K36000000      //Cam axis length per cycle=360.0[degree]
D6004L=K0             // Cam data starting point=0th point
D6006=K5              //Number of sections=5 sections
D6007=K0              //Unusable
D6008=K0              //(Section 1) Cam curve type =Constant speed
D6009=K0              //Unusable
D6010L=K9000000       //(Section 1) End point (X1)=90.0[degree]
D6012L=K0             //(Section 1) Stroke (Y1)=0[%]
D6014=K0              //(Section 2) Cam curve type =Constant speed
D6015=K0              //Unusable
D6016L=K17000000      //(Section 2) End point (X2)=170.0[degree]
D6018L=K800000000     //(Section 2) Stroke (Y2)=80[%]
D6020=K0              //(Section 3) Cam curve type =Constant speed
D6021=K0              //Unusable
D6022L=K19000000      //(Section 3) End point (X3)=190.0[degree]
D6024L=K800000000     //(Section 3) Stroke (Y3)=80[%]
D6026=K0              //(Section 4) Cam curve type=Constant speed
D6027=K0              //Unusable
D6028L=K27000000      //(Section 4) End point (X4)=270.0[degree]
D6030L=K0             //(Section 4) Stroke (Y4)=0[%]
D6032=K0              //(Section 5) Cam curve type=Constant speed
D6033=K0              //Unusable
D6034L=K36000000      //(Section 5) End point (X5)=360.0[degree]
D6036L=K0             //(Section 5) Stroke (Y5)=0[%]
CAMMK K6,K2,D6000     //Cam auto-generation
```



F/FS	G
○	○

5.18.5 Cam position calculation: CAMPSCL

Format	CAMPSCL(S1), (S2), (D)

Number of basic steps	6

[Usable data]

Setting data	Usable Data										
	Bit device	Word device (Note-1)				Constant			Calculation expression	Bit conditional expression	Comparison conditional expression
		16-bit integer type	32-bit integer type (L)	64-bit floating point type (F)	Coasting timer	16-bit integer type (K/H)	32-bit integer type (K/H, L)	64-bit floating point type (K)			
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	—	—	—	—	—	—
(D)	—	—	○	—	—	—	—	—	—	—	—

○ : Usable

(Note-1): The special register (SD) cannot be used.

[Setting data]

Setting data	Description	Data type of result
(S1)	Cam position calculation: Cam No. (1 to 256)	—
(S2)	Start device No. which stores the cam position calculation control data	
(D)	Device No. which stores the cam position calculation result	

[Functions]

(1) For the cam No. data specified with (S1), the cam axis current feed value or the cam axis current value per cycle is calculated from the cam position calculation control data specified with (S2), and the value is output to the device specified with (D).

(2) Specify the cam No. to perform the cam position calculation in (S1). When cam No. 0 is specified, the cam position is calculated as the linear cam.

(3) The device No. specified with (S2) should be an even number. Set the cam position calculation control data in the specified device as follows.

(a) Device assignment of the cam position calculation control data

Off set	Name	Description	Range
+0	Cam position calculation type	Specify the cam axis current feed value calculation/cam axis current value per cycle calculation	0: Cam axis current feed value calculation 1: Cam axis current value per cycle calculation
+1	Unusable	Set 0.	0
+2 +3	Cam stroke amount	Set the cam stroke amount for the cam position calculation.	-2147483648 to 2147483647 [Output axis position units]
+4 +5	Cam axis length per cycle	Set the cam axis length per cycle for the cam position calculation.	1 to 2147483647 [Cam axis cycle unit]
+6 +7	Cam reference position	Set the cam reference position for the cam position calculation.	-2147483648 to 2147483647 [Output axis position units]
+8 +9	Cam axis current value per cycle	• Set the cam axis current value per cycle for the cam position calculation when calculating the cam axis current feed value. • Set the cam axis current value per cycle as the starting point to search when calculating the cam axis current value per cycle and the cam position.	0 to (Cam axis length per cycle) [Cam axis cycle unit]
+10 +11	Cam axis current feed value	Set the cam axis current feed value for the cam position calculation when calculating the cam axis current value per cycle. (This is not used when the cam position calculation type is set to the cam axis current feed value calculation.)	-2147483648 to 2147483647 [Output axis position units]

(4) Specify the device No. with (D) to an even number.
The specified device stores the cam position calculation result as shown below when the calculation is completed.

• Cam axis current feed value calculation:
The cam axis current feed value that is calculated within the following range is stored.
-2147483648 to 2147483647 [Output axis position units]

• Cam axis current value per cycle calculation:
The cam axis current value per cycle that is calculated within the following range is stored.
0 to (Cam axis length per cycle-1) [Cam axis cycle unit]

(5) The cam position calculation does not update the cam reference position automatically.

[Errors]

(1) An operation error will occur, and the cam position calculation will not be executed if:
- Cam No. specified with (S1) is outside the range of 0 to 256.
- The cam No. data specified with (S1) does not exist in the cam open area.
- The device numbers storing the cam position calculation control data specified with (S2) are outside the range.
- (S2), (D) are not even-numbered devices.
- Cam position calculation type specified with cam position calculation control data is set to other than 0 or 1.
- Cam axis length per cycle is outside the range of 1 to 2147483647.
- Cam axis current value per cycle is outside the range of 0 to (cam axis length per cycle).
- The device numbers storing the cam position calculation result specified with (D) are outside the range.
- The cam axis current value per cycle cannot be calculated by the cam axis current value per cycle calculation.

[Program examples]

(1) Program which calculates the cam axis current feed value in the linear cam pattern (cam No. 0)

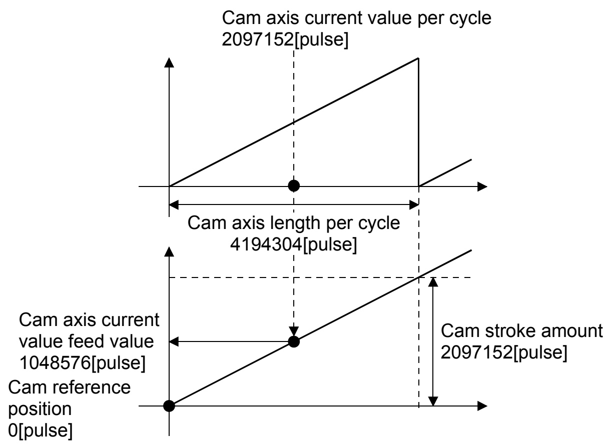
```
Cam (Linear)

[F10]
//Cam position calculation data set
D10000=K0 //Cam position calculation type=
              Cam axis current feed value calculation
D10001=K0           //Unusable
D10002L=K2097152L//Cam stroke amount=
                     2097152[pulse]
D10004L=K4194304L//Cam axis length per cycle=
                     4194304[pulse]
D10006L=K0L         //Cam reference position=0[pulse]
D10008L=K2097152L
  //Cam axis current value per cycle=2097152[pulse]
D10010L=K0L         //Cam axis current feed value=
                     0 (The setting is not required.)

[G10]
PX0            //Standby until PX0 turns ON

[F20]
//Cam position calculation for cam No.0
  (The result is stored in W1F00L.)
CAMPSCL K0,D10000,W1F00L

END
```



(2) Program which calculates the cam axis current feed value in the two-way cam pattern operation.

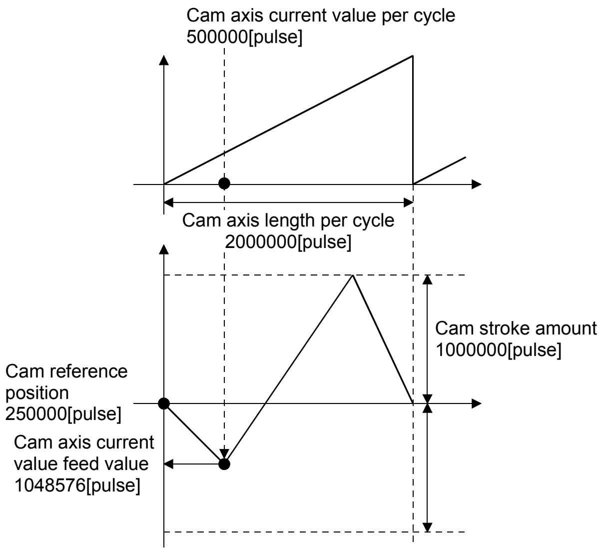
```
Cam (Two-way)

[F10]
//Set No.5 (Two-way cam) to the Cam No.
#1000=K5
//Cam position calculation data set
D2200=K0 //Cam position calculation type=
          Cam axis current feed valuecalculation
D2201=K0            //Unusable
D2202L=K1000000L//Cam stroke amount=1000000[pulse]
D2204L=K2000000L//Cam axis length per cycle=
                  2000000[pulse]
D2206L=K250000L  //Cam reference position=
                  250000[pulse]
D2208L=K500000L //Cam axis current value per cycle=
                 500000[pulse]
D2210L=K0L        //Cam axis current feed value=
                 0 (The setting is not required.)

[G10]
PX0              //Standby until PX0 turns ON

[F20]
//Cam position calculation for cam No.0
  (The result is stored in W1F00L.)
CAMPSCL #1000,D2200,W1F00L

END
```



(3) Program which calculates the cam axis current value per cycle in the feed operation cam pattern

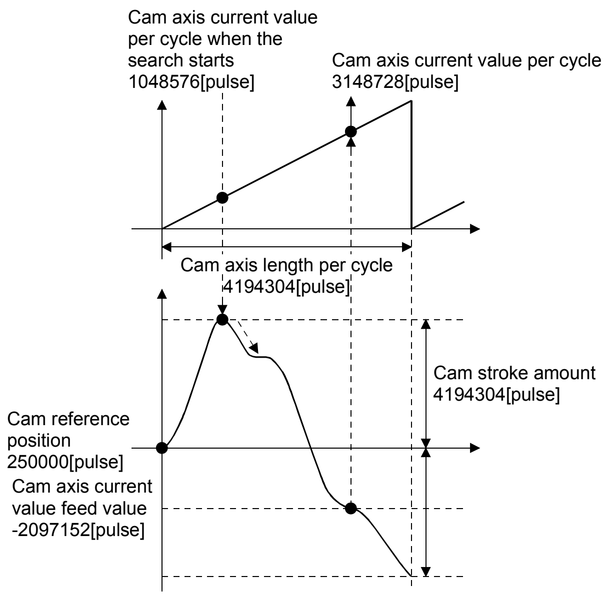
```
Cam (Feed)

[F10]
//Set No.253 (Feed cam) to the Cam No.
W1000=K256
//Cam position calculation data set
W0000=K1 //Cam position calculation type=
          Cam axis current feed value calculation
W0001=K0             //Unusable
W0002L=K4194304L   //Cam stroke amount=
                    4194304[pulse]
W0004L=K4194304L   //Cam axis length per cycle=
                    4194304[pulse]
W0006L=K0L          //Cam reference position=0[pulse]
W0008L=K1048576L  //Cam axis current value per cycle
                   =1048576[pulse]
W000AL=K-2097152L//Cam axis current feed value=
                   -2097152[pulse]

[G10]
PX0              //Standby until PX0 turns ON

[F20]
//Cam position calculation for cam No.0
  (The result is stored in W1F00L.)
CAMPSCL W1000,W0000,#0L

END
```



6. TRANSITION PROGRAMS

6.1 Transition Programs

(1) Transition programs

(a) Substitution operation expressions, motion-dedicated functions, bit device control commands and transition conditions can be set in transition programs.

(b) Multiple blocks can be set in one transition program.

(c) There are no restrictions on the number of blocks that may be set in a single transition program.
Note that one program is within 64k bytes.

(d) The maximum number of characters in one block is 128.

(e) Transition condition must be set in the last block of a transition program.
Transition program is repeated until the transition condition enables, and when the transition condition has enabled, it shifts to the next step.
Transition condition can be set only in the last block.

(f) As a special transition program, a program which only no operation (NOP) is set in one block can be created.
This program is used when it is not set as interlock to process to next step with completion of servo program.
Refer to Section "4.9 Branches, Couplings" for details.

A transition program example is shown below.

1 block

```
#0=D0+(D1+D2)*#5//Substitution expression (four arithmetic operations)
W0:F=SIN(#10F)//Substitution expression (standard function)
CHGV(K2,K10)//Motion-dedicated function
SET M100=M0+X0//Bit device control (SET=)
RST M10=D100>K10//Bit device control (RST=)
DIN D0,X0//Bit device control (DIN)
D0>K100//Standby until transition condition enables
```

1 program

Comment

Transition condition

What can be set as a transition condition in the last block are bit conditional expressions, comparison conditional expressions and device set (SET=)/device reset (RST=) which return logical data values (true/false). In the case of device set (SET=)/device reset (RST=), whether the bit or comparison conditional expression specified at (S) is true or false is a transition condition, and when the transition condition enables, device set/reset is executed and execution shifts to the next step.

Transition condition description examples are given below.

Classification	Description example
Bit conditional expression	M0
	!M0+X10*M100
Comparison conditional expression	(D0>K100)+(D100L!=K20L)
Device set (SET=)	SET Y0=M100
Device reset (RST=)	RST M10=D0==K100

POINT

(1) A transition program differs from an operation control program in that a transition condition is set in the last block.
Other settings are the same as those of the operation control program.

(2) When setting device set (SET=)/device reset (RST=) in the last block as a transition condition, the bit or comparison conditional expression specified with (S) is not omissible.

(3) Only the bit or comparison conditional expression cannot be set in other than the last block. Device set (SET=)/device reset (RST=) can be set in other than the last block.

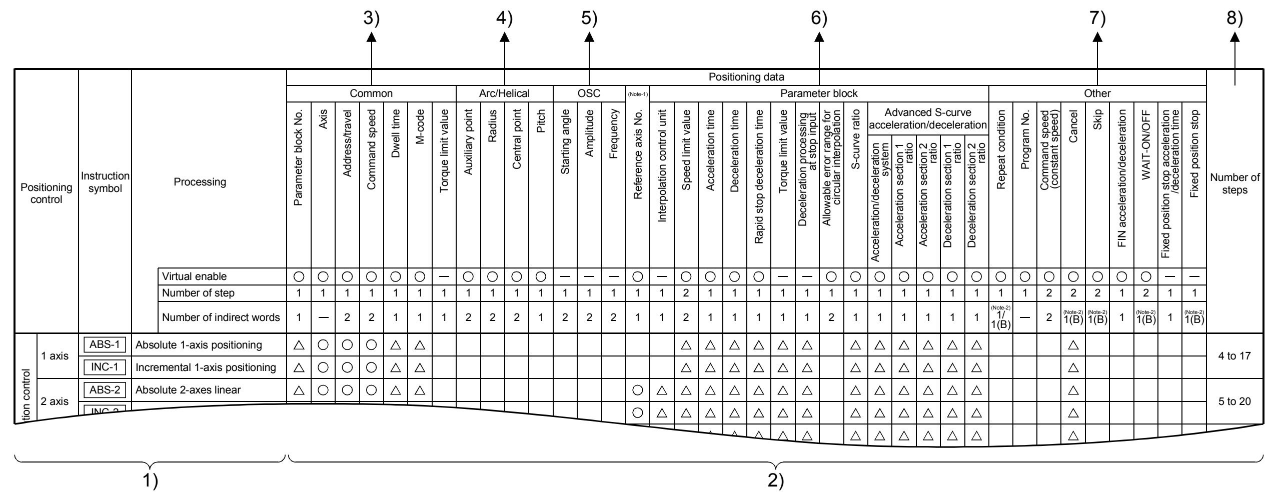
7. MOTION CONTROL PROGRAMS

7.1 Servo Instruction List

Table 7.1 lists servo instructions used in servo programs.
Refer to Section 7.2 to 7.4 for details of the current value change control (CHGA, CHGA-E, CHGA-C).
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion Controller (SV13/SV22) Programming Manual (REAL MODE)" for other servo instructions.

(1) Guide to servo instruction list

Table 7.1 Guide to Servo Instruction List



Number	Description	
1)	Instruction symbol	Gives the servo instructions usable in servo programs.
	Processing	Gives the processing outlines of the servo instructions.
2)	(a) Indicates positioning data which can be set in servo instructions. 1) ○: Item which must be set (Data which cannot execute the servo instruction unless it sets.) 2) △: Item which is set when required (Data which will be controlled by the default value unless it sets.)	
	(b) Allows direct or indirect designation (except axis No.) 1) Direct designation : Set with numerical value. 2) Indirect designation : Set with word device. • Servo program execution is controlled using the preset word device contents. • Each setting item may either be 1 or 2 word data. • For 2 word data, set the start device No.	
	(c) Number of steps As there are more setting items, there are more number of instruction steps. (The number of steps is displayed when a servo program is created.) (The instruction + ○ item comprise the minimum steps, and one △ item increases the number of steps by 1.)	
3)	Items common to the servo instructions	
4)	Items set in circular interpolation starting servo programs	
5)	Items set for high-speed oscillation	
6)	Set when changing the parameter block (default value when not set) data set in the servo program to control. (The parameter block data are not changed.)	
7)	Setting items other than the common, arc and parameter block items (Items to be set vary with the servo instruction.)	
8)	Indicates the number of steps of each servo instruction.	

(2) Servo instruction list

Table 7.2 indicates the servo instructions available for servo programs and the positioning data set in servo instructions.

Table 7.2 Servo Instruction List

Positioning control		Instruction symbol	Processing	Positioning data: Common							Arc/Helical				OSC		
				Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
			Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
			Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Linear interpolation control	1 axis	ABS-1	Absolute 1-axis positioning	△	○	○	○	△	△								
		INC-1	Incremental 1-axis positioning	△	○	○	○	△	△								
	2 axes	ABS-2	Absolute 2-axes linear interpolation	△	○	○	○	△	△								
		INC-2	Incremental 2-sxes linear interpolation	△	○	○	○	△	△								
	3 axes	ABS-3	Absolute 3-axes linear interpolation	△	○	○	○	△	△								
		INC-3	Incremental 3-axes linear interpolation	△	○	○	○	△	△								
	4 axes	ABS-4	Absolute 4-axes linear interpolation	△	○	○	○	△	△								
		INC-4	Incremental 4-axes linear interpolation	△	○	○	○	△	△								
Circular interpolation control	Auxiliary point-specified	ABS ⌒	Absolute auxiliary point-specified circular interpolation	△	○	○	○	△	△		○						
		INC ⌒	Incremental auxiliary point-specified circular interpolation	△	○	○	○	△	△		○						
	Radius-specified	ABS ↷	Absolute radius-specified circular interpolation less than CW 180°	△	○	○	○	△	△			○					
		ABS ↻	Absolute radius-specified circular interpolation CW 180° or more	△	○	○	○	△	△			○					
		ABS ↶	Absolute radius-specified circular interpolation less than CCW 180°	△	○	○	○	△	△			○					
		ABS ↺	Absolute radius-specified circular interpolation CCW 180° or more	△	○	○	○	△	△			○					
		INC ↷	Incremental radius-specified circular interpolation less than CW 180°	△	○	○	○	△	△			○					
		INC ↻	Incremental radius-specified circular interpolation CW 180° or more	△	○	○	○	△	△			○					
		INC ↶	Incremental radius-specified circular interpolation less than CCW 180°	△	○	○	○	△	△			○					
		INC ↺	Incremental radius-specified circular interpolation CCW 180° or more	△	○	○	○	△	△			○					

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
		△	△	△	△	△	△		△	△	△	△	△	△				△						4 to 17
		△	△	△	△	△	△		△	△	△	△	△	△				△						
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						5 to 20
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						7 to 21
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						8 to 22
○	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						7 to 22
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						6 to 21
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control		Instruction symbol	Processing	Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
			Positioning data:	Common							Arc/Helical				OSC		
			Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
			Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Circular interpolation control	Central point-specified	ABS	Absolute central point-specified circular interpolation CW	△	○	○	○	△	△				○				
		ABS	Absolute central point-specified circular interpolation CCW	△	○	○	○	△	△				○				
		INC	Incremental central point-specified circular interpolation CW	△	○	○	○	△	△				○				
		INC	Incremental central point-specified circular interpolation CCW	△	○	○	○	△	△				○				
Helical interpolation control	Auxiliary point-specified	ABH	Absolute auxiliary point- specified helical interpolation	△	○	○	○	△	△		○			○			
		INH	Incremental auxiliary point- specified helical interpolation	△	○	○	○	△	△		○			○			
	Radius-specified	ABH	Absolute radius-specified helical interpolation less than CW 180°	△	○	○	○	△	△			○		○			
		ABH	Absolute radius-specified helical interpolation CW 180° or more	△	○	○	○	△	△			○		○			
		ABH	Absolute radius-specified helical interpolation less than CCW 180°	△	○	○	○	△	△			○		○			
		ABH	Absolute radius-specified helical interpolation CCW 180° or more	△	○	○	○	△	△			○		○			
		INH	Incremental radius-specified helical interpolation less than CW 180°	△	○	○	○	△	△			○		○			
		INH	Incremental radius-specified helical interpolation CW 180° or more	△	○	○	○	△	△			○		○			
		INH	Incremental radius-specified helical interpolation less than CCW 180°	△	○	○	○	△	△			○		○			
		INH	Incremental radius-specified helical interpolation CCW 180° or more	△	○	○	○	△	△			○		○			
	Central point-specified	ABH	Absolute central point-specified helical interpolation CW	△	○	○	○	△	△				○	○			
		ABH	Absolute central point-specified helical interpolation CCW	△	○	○	○	△	△				○	○			
		INH	Incremental central point-specified helical interpolation CW	△	○	○	○	△	△				○	○			
		INH	Incremental central point-specified helical interpolation CCW	△	○	○	○	△	△				○	○			

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						7 to 22
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						10 to 27
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						9 to 26
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						10 to 27
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control		Instruction symbol	Processing	Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
				Positioning data: Common							Arc/Helical				OSC		
			Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
			Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
			Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Fixed-pitch feed	1 axis	FEED-1	1-axis fixed-pitch feed start	△	○	○	○	△	△								
Fixed-pitch feed	2 axes	FEED-2	2-axes linear interpolation fixed-pitch feed start	△	○	○	○	△	△								
Fixed-pitch feed	3 axes	FEED-3	3-axes linear interpolation fixed-pitch feed start	△	○	○	○	△	△								
Speed control (Ⅰ)	Forward rotation	VF	Speed control (Ⅰ) forward rotation start	△	○		○		△								
Speed control (Ⅰ)	Reverse rotation	VR	Speed control (Ⅰ) reverse rotation start	△	○		○		△								
Speed control (Ⅱ)	Forward rotation	VVF	Speed control (Ⅱ) forward rotation start	△	○		○		△	△							
Speed control (Ⅱ)	Reverse rotation	VVR	Speed control (Ⅱ) reverse rotation start	△	○		○		△	△							
Speed-position control	Forward rotation	VPF	Speed-position control forward rotation start	△	○	○	○	△	△	△							
Speed-position control	Reverse rotation	VPR	Speed-position control reverse rotation start	△	○	○	○	△	△	△							
Speed-position control	Restart	VPSTART	Speed-position control restart		○												
Speed-switching control		VSTART	Speed-switching control start	△													
Speed-switching control		VEND	Speed-switching control end														
Speed-switching control		ABS-1	Speed-switching control end point address		○	○	○	△	△	△							
Speed-switching control		ABS-2	Speed-switching control end point address		○	○	○	△	△	△							
Speed-switching control		ABS-3	Speed-switching control end point address		○	○	○	△	△	△							
Speed-switching control		INC-1	Travel value up to speed-switching control end point		○	○	○	△	△	△							
Speed-switching control		INC-2	Travel value up to speed-switching control end point		○	○	○	△	△	△							
Speed-switching control		INC-3	Travel value up to speed-switching control end point		○	○	○	△	△	△							
Speed-switching control		VABS	Speed-switching point absolute specification			○	○		△	△							
Speed-switching control		VINC	Speed-switching point incremental specification			○	○		△	△							

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
		△	△	△	△	△	△		△	△	△	△	△	△				△						4 to 17
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						5 to 19
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						7 to 21
		△	△	△	△	△	△		△	△	△	△	△	△				△						3 to 15
		△	△	△	△	△	△		△	△	△	△	△	△				△						
		△	△	△	△	△	△		△	△	△	△	△	△				△						3 to 16
		△	△	△	△	△	△		△	△	△	△	△	△				△						
		△	△	△	△	△	△		△	△	△	△	△	△				△						4 to 18
		△	△	△	△	△	△		△	△	△	△	△	△				△						
																		△						2 to 4
	△	△	△	△	△	△	△		△	△	△	△	△	△				△						1 to 13
																								1
																		△						4 to 9
																		△						5 to 10
																		△						7 to 12
																		△						4 to 9
																		△						5 to 10
																		△						7 to 12
																								4 to 6

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data													
			Common							Arc/Helical				OSC		
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
		Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
		Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Speed control with fixed position stop: Forward rotation	PVF	Speed control with fixed position stop absolute specification	△	○	○	○	△	△								
Speed control with fixed position stop: Reverse rotation	PVR		△	○	○	○	△	△								
Position follow-up control	PFSTART	Position follow-up control start	△	○	○	○		△								
Constant-speed control	CPSTART1	1-axis constant-speed control start	△	○		○										
	CPSTART2	2-axes constant-speed control start	△	○		○										
	CPSTART3	3-axes constant-speed control start	△	○		○										
	CPSTART4	4-axes constant-speed control start	△	○		○										
	ABS-1	Constant-speed control passing point absolute specification		○	○			△	△							
	ABS-2			○	○			△	△							
	ABS-3			○	○			△	△							
	ABS-4			○	○			△	△							
	ABS⌒			○	○			△	△	○						
	ABS↶			○	○			△	△		○					
	ABS↷			○	○			△	△		○					
	ABS↶			○	○			△	△		○					
	ABS↷			○	○			△	△		○					
	ABS↶			○	○			△	△			○				
	ABS↷			○	○			△	△			○				
	ABH⌒	Constant-speed control passing point helical absolute specification		○	○			△	△	○			○			
	ABH↶			○	○			△	△		○		○			
	ABH↷			○	○			△	△		○		○			
	ABH↶			○	○			△	△		○		○			
	ABH↷			○	○			△	△		○		○			
	ABH↶			○	○			△	△			○	○			
	ABH↷			○	○			△	△			○	○			

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
		△		△	△	△	△		△	△	△	△	△	△				△				○	○	6 to 19
		△		△	△	△	△		△	△	△	△	△	△				△				○	○	
		△	△	△	△	△	△		△	△	△	△	△	△				△						4 to 16
		△	△	△	△	△	△		△	△	△	△	△	△				△		△				3 to 15
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△		△				3 to 17
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△		△				4 to17
	△	△	△	△	△	△	△	△	△	△	△	△	△	△				△		△				
																	△		△		△			2 to 10
																	△		△		△			3 to 11
																	△		△		△			4 to 12
																	△		△		△			5 to 13
																	△		△		△			5 to 14
																	△		△		△			4 to 13
																	△		△		△			
																	△		△		△			
																	△		△		△			
																	△		△		△			5 to 14
																	△		△		△			
																	△		△		△			9 to 14
																	△		△		△			8 to 13
																	△		△		△			
																	△		△		△			
																	△		△		△			
																	△		△		△			9 to 14
																	△		△		△			

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data													
			Common							Arc/Helical				OSC		
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
		Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
		Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Constant-speed control	INC-1	Constant-speed control passing point incremental specification		○	○			△	△							
	INC-2			○	○			△	△							
	INC-3			○	○			△	△							
	INC-4			○	○			△	△							
	INC			○	○			△	△	○						
	INC			○	○			△	△		○					
	INC			○	○			△	△		○					
	INC			○	○			△	△		○					
	INC			○	○			△	△		○					
	INC			○	○			△	△			○				
	INC			○	○			△	△			○				
	INH	Constant-speed control passing point helical incremental specification		○	○			△	△	○			○			
	INH			○	○			△	△		○		○			
	INH			○	○			△	△		○		○			
	INH			○	○			△	△		○		○			
	INH			○	○			△	△		○		○			
	INH			○	○			△	△			○	○			
	INH			○	○			△	△			○	○			
	CPEND	Constant-speed control end					△									

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
																	△		△		△			2 to 10
																	△		△		△			3 to 11
																	△		△		△			4 to 12
																	△		△		△			5 to 13
																	△		△		△			5 to 14
																	△		△		△			4 to 13
																	△		△		△			
																	△		△		△			
																	△		△		△			
																	△		△		△			5 to 14
																	△		△		△			
																	△		△		△			9 to 14
																	△		△		△			8 to 13
																	△		△		△			
																	△		△		△			
																	△		△		△			
																	△		△		△			9 to 14
																	△		△		△			
																								1 to 2

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

Table 7.2 Servo Instruction List (continued)

Positioning control	Instruction symbol	Processing	Positioning data													
			Common							Arc/Helical				OSC		
			Parameter block No.	Axis	Address/travel value	Command speed	Dwell time	M -code	Torque limit value	Auxiliary point	Radius	Central point	Pitch	Starting angle	Amplitude	Frequency
		Virtual enable	○	○	○	○	○	○	—	○	○	○	○	—	—	—
		Number of steps	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		Number of indirect words	1	—	2	2	1	1	1	2	2	2	1	2	2	2
Repetition of same control (used in speed switching control, constant-speed control)	FOR-TIMES	Repeat range start setting														
	FOR-ON															
	FOR-OFF															
	NEXT	Repeat range end setting														
Simultaneous start	START	Simultaneous start														
Home position return	ZERO	Home position return start		○												
High speed oscillation	OSC	High-speed oscillation	△	○				△						○	○	○
Current value change	CHGA	Servomotor/Virtual Servomotor Shaft Current Value Change		○	○											
	CHGA-E	Encoder current value change		○	○											
	CHGA-C	CAM shaft current value change		○	○											

Positioning data																								Number of steps
(Note-1)	Parameter block														Others									
										Advanced S-curve acceleration/deceleration														
Reference axis No.	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Acceleration/deceleration system	Acceleration section 1 ratio	Acceleration section 2 ratio	Deceleration section 1 ratio	Deceleration section 2 ratio	Repeat condition	Program No.	Command speed (constant speed)	Cancel	Skip	FIN acceleration/deceleration	WAIT-ON/OFF	Fixed position stop acceleration /deceleration time	Fixed position stop	
○	—	○	○	○	○	—	—	○	○	○	○	○	○	○	○	○	○	○	○	○	○	—	—	
1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	1	2	1	1	
1	1	2	1	1	1	1	1	2	1	1	1	1	1	1	(Note-2) 1/ 1(B)	—	2	(Note-2) 1(B)	(Note-2) 1(B)	1	(Note-2) 1(B)	1	(Note-2) 1(B)	
															○									
															○									2
															○									
																								3
																○								2 to 3
																								2
						△												△						5 to 10
																								3

○ : Must be set. △ : Set if required.
(Note-1) : Only reference axis speed specification.
(Note-2) : (B) indicates a bit device.

7.2 Servo Motor/Virtual Servo Motor Shaft/Command Generation Axis Current Value Change

The current value of the specified axis/virtual servo motor/command generation axis is changed

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer2																						Speed change
			Common							Arc			Parameter block										Others		
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Advanced S-curve acceleration/deceleration	Cancel	FIN acceleration/deceleration	
CHGA	Absolute	1		○	○																				Disable

○: Item which must be set
△: Item which is set when required

[Controls]

Control using CHGA instruction

(1) Executing the CHGA instruction changes the current value in the following procedure.
   (a) The start accept flag (M2001 to M2032) corresponding to the specified axis is turned on.
   (Note): For the command generation axis, [St.345] Command generation axis start accept flag (M9810+20n) corresponding to the specified axis is turned on.
   (b) The current value of the specified axis is changed to the specified address.
   (c) Start accept flag is turned off at completion of the current value change.

(2) The current value of the specified axis is changed in the SV13/SV22 real mode. The address which made the current value change by CHGA instruction is valid on the power supply turning on.

(3) The current value of the specified virtual servo motor shaft is changed in the SV22 virtual mode.

(4) The current value of the specified command generation axis is changed in the SV22 advanced synchronous control. QDS Ver.!

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(5) The used axis No. can be set within the following range.

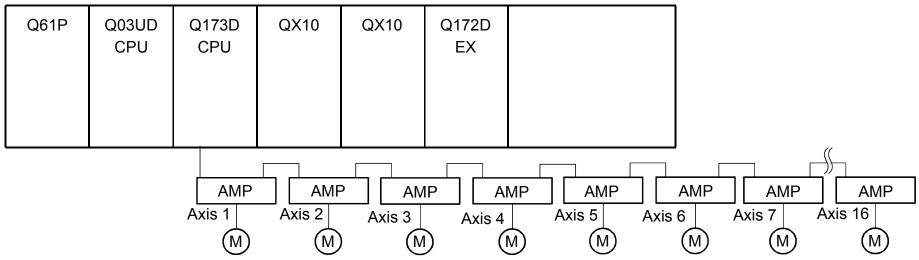
Q173DSCPU	Q173DCPU(-S1)	Q172DSCPU	Q172DCPU(-S1)
Axis 1 to 32		Axis 1 to 16	Axis 1 to 8

[Program example]

A program which made the current value change control in the real mode is described as the following conditions.

(1) System configuration

The current value change control of axis 2 is executed.

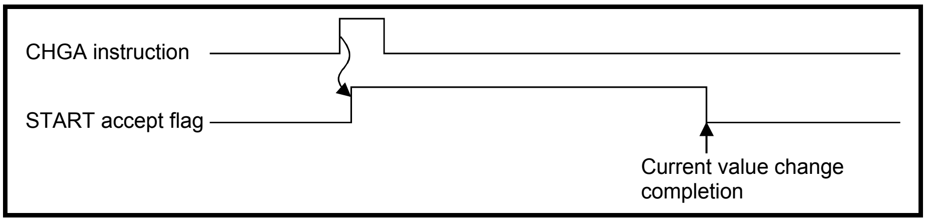


(2) The current value change control conditions

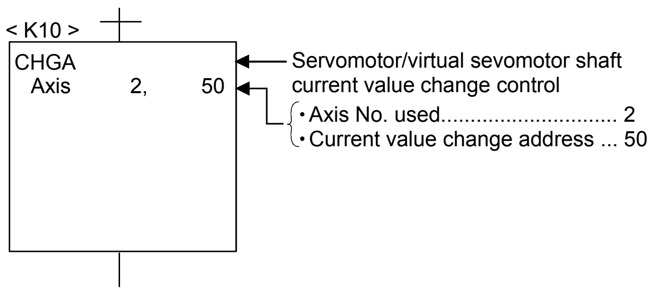
(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Control axis No.	2
Current value change address	50

(3) Operation timing



(4) Servo program



POINT

Current value changing instructions

- When PLC ready flag (M2000) or PCPU READY complete flag (SM500) is OFF, a minor error (Note) [100] occurs and a current value change is not made.
- This change is made only during a stop. If a current value change is made while the specified axis is starting, a minor error (Note) [101] (start accept signal of the corresponding axis is ON) occurs and the current value change is not made.
- If the servo of the corresponding axis is not READY, a major error (Note) [1004] occurs and the current value change is not made.
- If the corresponding axis is in a servo error, a major error (Note) [1005] occurs and the current value change is not made.

For SV22

- The current value change of specified axis is executed in the real mode, and the current value change of specified servomotor axis is executed in the virtual mode.
- Set the current value change program of the virtual servo motor shaft within the virtual mode program No. range set in "K mode allocation" of MT Developer2.
- Set the current value change program of the command generation axis within the command generation axis program No. range set in "Command generation axis program allocation setting" of MT Developer2. QDS
- Set the current value change program of the servomotor (output) shaft within the real mode program No. range.
- If a virtual servomotor shaft current value change is executed in the real mode, a servo program setting error (Note) [903] occurs and the current value change is not made.
- If a servomotor (output) shaft current value change is executed in the virtual mode, a servo program setting error (Note) [904] occurs and the current value change is not made.
- If a current value change is made during mode changing, a servo program setting error (Note) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made.

(Note): Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/ "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

7.3 Synchronous Encoder Shaft Current Value Change Control (SV22 virtual mode only)

The current value of the specified synchronous encoder shaft is changed.

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer2																						Speed change
			Common							Arc			Parameter block										Others		
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Advanced S-curve acceleration/deceleration	Cancel	FIN acceleration/deceleration	
CHGA-E	Absolute	1		○	○																				Disable

○: Item which must be set
△: Item which is set when required

[Controls]

Control using CHGA-E instruction

(1) Executing the CHGA-E instruction changes the current value of the synchronous encoder shaft in the following procedure.
  (a) The synchronous encoder shaft current value changing flag (M2101 to M2112) corresponding to the specified synchronous encoder shaft is turned on.
  (b) The current value of the specified synchronous encoder shaft is changed to the specified address.
  (c) The synchronous encoder shaft current value changing flag is turned off at completion of the current value change.

(2) The used axis No. can be set within the following range.

Q173DSCPU	Q173DCPU(-S1)	Q172DSCPU	Q172DCPU(-S1)
Axis 1 to 12			Axis 1 to 8

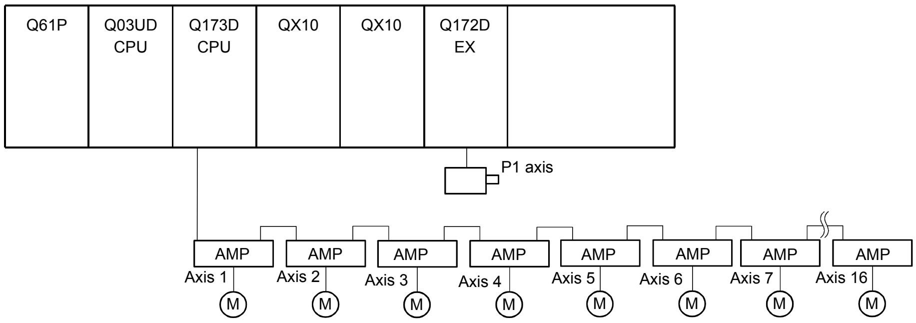
(3) The address which made the current value change by CHGA-E instruction is valid after also the power supply turned off.

[Program example]

A program which made the current value change control of the synchronous encoder shaft is described as the following conditions.

(1) System configuration

The current value change control of the synchronous encoder shaft P1 is executed.

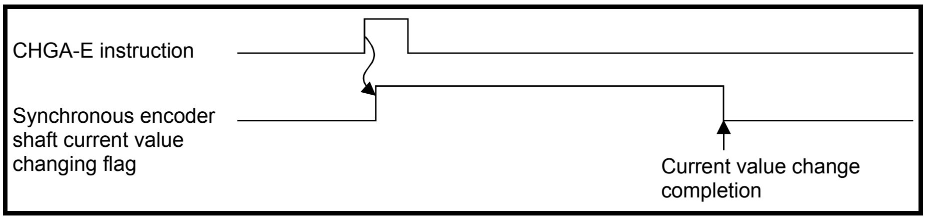


(2) The current value change control conditions

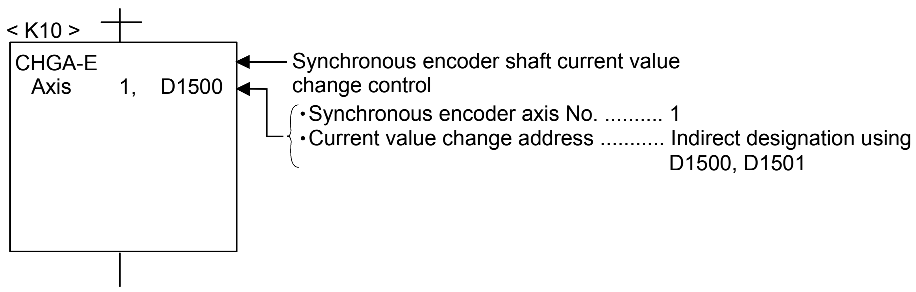
(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Synchronous encoder No.	1
Current value change address	Indirect designation using D1500, D1501

(3) Operation timing



(4) Servo program



POINT

Synchronous encoder current value changing instructions

- The current value change of the synchronous encoder is executed if operation is being performed in the virtual mode (during pulse input from the synchronous encoder).
  If the current value is changed, the feed current value of the synchronous encoder continues from the new value. (Note-1)
- The current value change of the synchronous encoder does not affect the current value of the output module.
- Set the current value change program of the synchronous encoder shaft program within the virtual mode program No. range set in "K mode allocation" of MT Developer2.
- When PLC ready flag (M2000) or PCPU READY complete flag (SM500) is OFF, a minor error (Note-2) [100] occurs and a current value change is not made.
- If a synchronous encoder current value change is executed in the real mode, a servo program setting error (Note-2) [903] or [905] occurs and the current value change is not made. ([903] when the current value change servo program is set to within the virtual mode program No. range, or 905 when it is set to within the real mode program No. range.) (Note-1)
- If a current value change is made during mode changing, a servo program setting error (Note-2) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made. (Note-1)

(Note-1): The current value change can be executed in real mode for the version (Refer to Section 1.3) that supports "incremental synchronous encoder current value in real mode".

(Note-2): Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/"Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

7.4 Cam Shaft Within-One-Revolution Current Value Change Control (SV22 virtual mode only)

The current value of the specified cam shaft within-one-revolution is changed.

Servo instruction	Positioning method	Number of Control axes	Items set using MT Developer2																						Speed change
			Common							Arc			Parameter block										Others		
			Parameter block No.	Axis	Address/travel	Command speed	Dwell time	M-code	Torque limit value	Auxiliary point	Radius	Central point	Interpolation control unit	Speed limit value	Acceleration time	Deceleration time	Rapid stop deceleration time	Torque limit value	Deceleration processing at stop input	Allowable error range for circular interpolation	S-curve ratio	Advanced S-curve acceleration/deceleration	Cancel	FIN acceleration/deceleration	
CHGA-C	Absolute	1		○	○																				Disable

○: Item which must be set

△: Item which is set when required

[Controls]

Control using CHGA-C instruction

(1) Executing the CHGA-C instruction changes the within-one-revolution current value of the specified cam shaft to the address.

(2) The cam shaft may be starting.

(3) The used axis No. can be set within the following range.

Q173DSCPU	Q173DCPU(-S1)	Q172DSCPU	Q172DCPU(-S1)
Axis 1 to 32		Axis 1 to 16	Axis 1 to 8

(4) The address which made the current value change by the CHGA-C instruction is valid after also the power supply turned off.

[Program example]

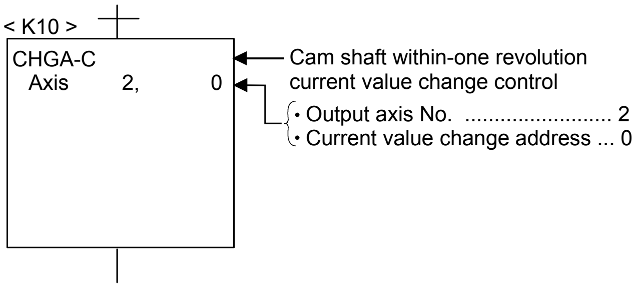
A program which made the current value change control of the cam shaft within-one-revolution current value change is described as the following conditions.

(1) Current value change control conditions

(a) The current value change control conditions are shown below.

Item	Setting
Servo program No.	10
Output axis No.	2
Current value change address	0

(2) Servo program



POINT

Cam shaft within-one revolution current value changing instructions

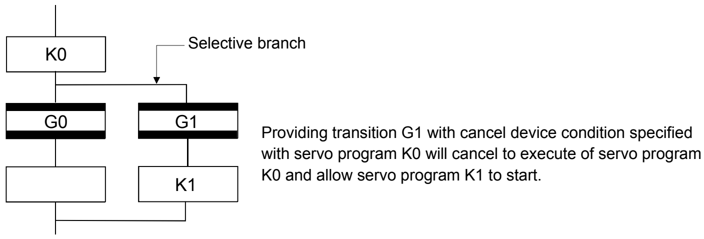
- If a new within-one revolution current value is outside the range 0 to (one-revolution pulse count - 1), a minor error (Note) [6120] occurs and current value change is not.
- Set the current value change program the cam shaft within-one-revolution within the virtual mode program No. range set in "K mode allocation" of MT Developer2.
- When PLC ready flag (M2000) or PCPU READY complete flag (SM500) is OFF, a minor error (Note) [100] occurs and a current value change is not made.
- If the cam shaft within-one-revolution current value change is executed in the real mode, a servo program setting error (Note) [903] or [905] occurs and the current value change is not made. ([903] when the current value change servo program is set to within the virtual mode program No. range, or 905 when it is set to within the real mode program No. range.)
- If a current value change is made during mode changing, a servo program setting error (Note) [907] (real mode → virtual mode changing) or [908] (virtual mode → real mode changing) occurs and the current value change is not made.

(Note): Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)"/"Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)" for minor error, major error and servo program setting error.

7.5 Programming Instructions

7.5.1 Cancel • start

When a cancel start has been set in the setting items of the servo program which was started at the motion control step of the Motion SFC program, the cancel of the running servo program is valid but the servo program specified to start after a cancel is ignored, without being started.
The following example shows the Motion SFC program which exercises control equivalent to a cancel start.



7.5.2 Indirect designation using motion devices

The coasting timer (FT) cannot used to make indirect specification in the servo program and mechanical system program.

8. MOTION DEVICES

The motion registers (#0 to #12287) and coasting timer (FT) are available as Motion CPU-dedicated devices.
They can be used in operation control (F/FS) programs or transition (G) programs.

8.1 Motion Registers (#0 to #12287)

Motion device	Item	Specifications
Motion register (#)	Number of points	12288 points (#0 to #12287)
	Data size	16-bit/point
	Latch	Only a user device is latched. (All points are cleared by latch clear operation.)
	Usable tasks	Normal, event and NMI
	Access	Read and write enabled in whole range

(1) Motion register list

- Common to all operating system software

Q173DSCPU/Q172DSCPU		Q173DCPU(-S1)/Q172DCPU(-S1)		Signal direction
Device No.	Application	Device No.	Application	
#0 to	User device (8000 points)	#0 to	User device (8000 points)	Cleared by latch clear.
#8000 to	Monitor device (640 points)	#8000 to	Monitor device (640 points)	Cleared at power on or reset only.
#8640 to	Motion error history device (96 points)	#8640 to	Motion error history device (96 points)	Cleared by the Motion error history request flag on. (keep at power on or reset).
#8736 to	Product information list device (16 points)	#8736 to	Product information list device Ver.! (16 points)	Set at power on or reset.
#8752 to #12287	System area (3536 points)	#8752 to #12287	System area (3536 points)	Cleared at power on or reset only.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

8

(a) Monitor devices (#8000 to #8639)
Information for each axis is stored in the monitor devices.
The details of the storage data are shown below.

Axis No.	Device No.
1	#8000 to #8019
2	#8020 to #8039
3	#8040 to #8059
4	#8060 to #8079
5	#8080 to #8099
6	#8100 to #8119
7	#8120 to #8139
8	#8140 to #8159
9	#8160 to #8179
10	#8180 to #8199
11	#8200 to #8219
12	#8220 to #8239
13	#8240 to #8259
14	#8260 to #8279
15	#8280 to #8299
16	#8300 to #8319
17	#8320 to #8339
18	#8340 to #8359
19	#8360 to #8379
20	#8380 to #8399
21	#8400 to #8419
22	#8420 to #8439
23	#8440 to #8459
24	#8460 to #8479
25	#8480 to #8499
26	#8500 to #8519
27	#8520 to #8539
28	#8540 to #8559
29	#8560 to #8579
30	#8580 to #8599
31	#8600 to #8619
32	#8620 to #8639

Signal name

	Signal name	Refresh cycle	Signal direction
0	Servo amplifier type	When the servo amplifier power-on	Monitor device
1	Motor current	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
2 3	Motor speed	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
4 5	Command speed	Operation cycle	Monitor device
6 7	Home position return re-travel value	At home position return re-travel	Monitor device
8	Servo amplifier display servo error code	Main cycle	Monitor device
9	Parameter error No. QDS	Main cycle	Monitor device
10	Servo status 1 QDS	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
11	Servo status 2 QDS	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
12	Servo status 3 QDS	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
13 14 15 16 17	Unusable	—	—
18	Servo status 7 QDS Ver.!	Operation cycle 1.7[ms] or less: Operation cycle Operation cycle 3.5[ms] or more: 3.5[ms]	Monitor device
19	Unusable	—	—

(Note): Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for the monitor devices.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(b) Motion error history devices (#8640 to #8735)

The Motion error history devices are shown below.

<table>
<tr><th rowspan="2">Device No.</th><th rowspan="2" colspan="2">Signal name</th><th colspan="2">Signal direction</th><th rowspan="2">Refresh cycle</th><th rowspan="2">Fetch cycle</th></tr>
<tr><th>Status</th><th>Command</th></tr>
<tr><td>#8640
to
#8651</td><td>Seventh error information in past
(Oldest error information)</td><td rowspan="8">Motion error history
(8 errors)
(96 points)</td><td rowspan="8">○</td><td rowspan="8">—</td><td rowspan="8">At error occurrence</td><td rowspan="8">—</td></tr>
<tr><td>#8652
to
#8663</td><td>Sixth error information in past</td></tr>
<tr><td>#8664
to
#8675</td><td>Fifth error information in past</td></tr>
<tr><td>#8676
to
#8687</td><td>Fourth error information in past</td></tr>
<tr><td>#8688
to
#8699</td><td>Third error information in past</td></tr>
<tr><td>#8700
to
#8711</td><td>Second error information in past</td></tr>
<tr><td>#8712
to
#8723</td><td>First error information in past</td></tr>
<tr><td>#8724
to
#8735</td><td>Latest error information</td></tr>
</table>

(Note-1): Refer to Section "12.2 Motion Error Related Device" for the Motion error history.

(c) Product information list device (#8736 to #8751) Ver.!

The product information list devices are shown below.

<table>
<tr><th>Device No.</th><th>Signal name</th><th>Refresh cycle</th><th>Fetch cycle</th><th>Signal direction</th></tr>
<tr><td>#8736
to
#8743</td><td>Operating system software version</td><td rowspan="2">At power on</td><td rowspan="2"></td><td rowspan="2">Monitor device</td></tr>
<tr><td>#8744
to
#8751</td><td>Motion CPU module serial number</td></tr>
</table>

(Note): Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)" for the product information list device.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

8.2 Coasting Timer (FT)

Motion device	Item	Specification
Coasting timer (FT) (Note-1), (Note-2)	Number of points	1 point (FT)
	Data size	32-bit/point (-2147483648 to 2147483647)
	Latch	No latch. Cleared to zero at power-on or reset, a count rise is continued from now on.
	Usable tasks	Normal, event, NMI
	Access	Read only enabled
	Timer specifications	888μs timer (Current value (FT) is incremented by 1 per 888μs.)

(Note-1): Use SD720 and SD721 for the 444μs coasting timer.
(Note-2): Use SD722 and SD723 for the 222μs coasting timer. QDS

9. OPERATION FOR MOTION SFC AND PARAMETER

9.1 Task Definitions

When to execute the Motion SFC program processing can be set only once in the program parameter per program.
Roughly classified, there are the following three different tasks.

Task type	Contents
Normal task	Executes in Motion CPU main cycle (free time).
Event task	1. Executes in fixed cycle (0.22ms QDS, 0.44ms, 0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms). 2. Executes when the input set to the event task factor among external interrupts (16 points of QI60) turns on. 3. Executes by an interrupt from the PLC CPU.
NMI task	Executes when the input set to the NMI task factor among external interrupts (16 points of QI60) turns on.

POINT

Set "0.2ms" as operation cycle in the system basic setting of MT Dveloper2 to execute the event task in fixed cycle 0.22ms. QDS

9.2 Number of Consecutive Transitions and Task Operation

9.2.1 Number of consecutive transitions

With "execution of active step → judgment of next transition condition → transition processing performed when condition enables (transition of active step)" defined as a single basic operation of the Motion SFC program execution control in the execution cycle of the corresponding task, this operation is performed for the number of active steps to terminate processing once. And the same operation is processed continuously in the next cycle.
In this case, the transition destination step is executed in the next cycle when the transition condition enables.

Consecutive transition control indicates that transition destination steps are executed one after another in the same one execution cycle when their transition conditions have enabled (single basic operation is performed consecutively).
In this case, set the number of consecutive transitions.
Control exercised is common to the Motion SFC programs executed by normal tasks.

POINT

Set the number of consecutive transitions to the Motion SFC programs executed by event and NMI tasks for every program.

9

9.2.2 Task operation

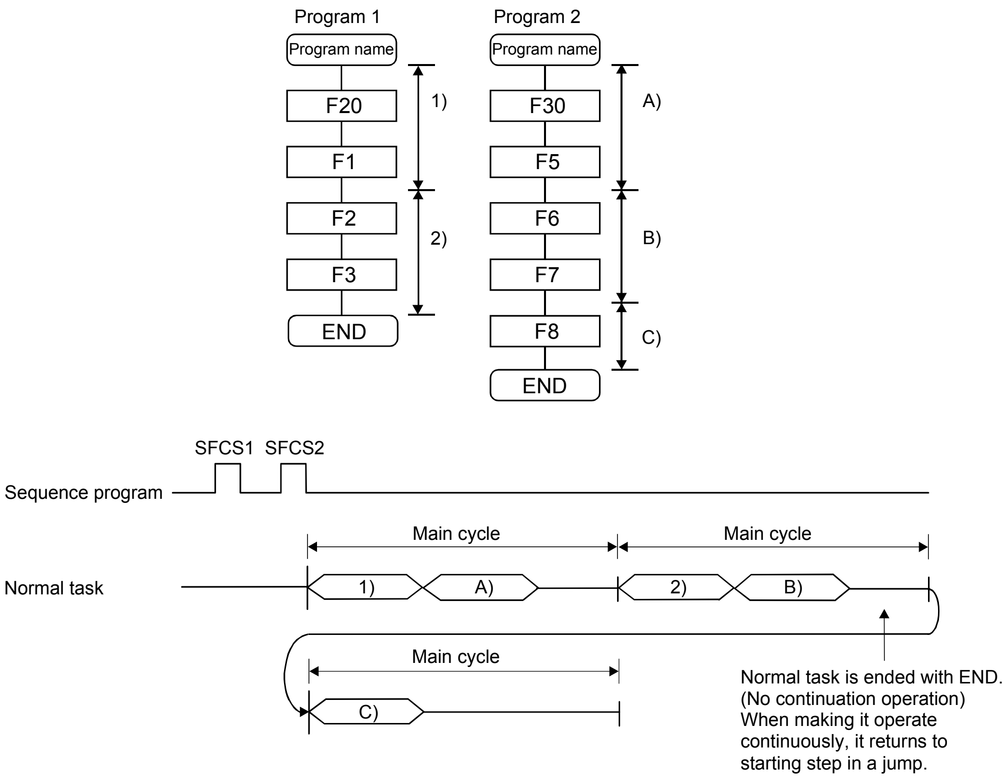
(1) Normal task operation

[Operations]

The Motion SFC program is executed in the main cycle (free time) of the Motion CPU.

The processing outline is shown below.

• Number of consecutive transitions is set to "2".



[Points]

(a) The Motion SFC program which includes motion control steps should be set to a normal task.

(b) During execution of an event or NMI task, the execution of the normal task is suspended.
Note that since the normal task allows the event task disable instruction (DI) to be described in an operation control step, the event task can be disabled in the area enclosed by the event task disable instruction (DI) and event task enable instruction (EI).

(2) Event task operation

[Operations]

An event task executes the Motion SFC program at occurrence of an event. There are the following events.

(a) Fixed cycle

The Motion SFC program is executed periodically in any of 0.22ms QDS, 0.44ms, 0.88ms, 1.77ms, 3.55ms, 7.11ms and 14.2ms cycles.

(b) External interrupt (16 points of I0 to I15)

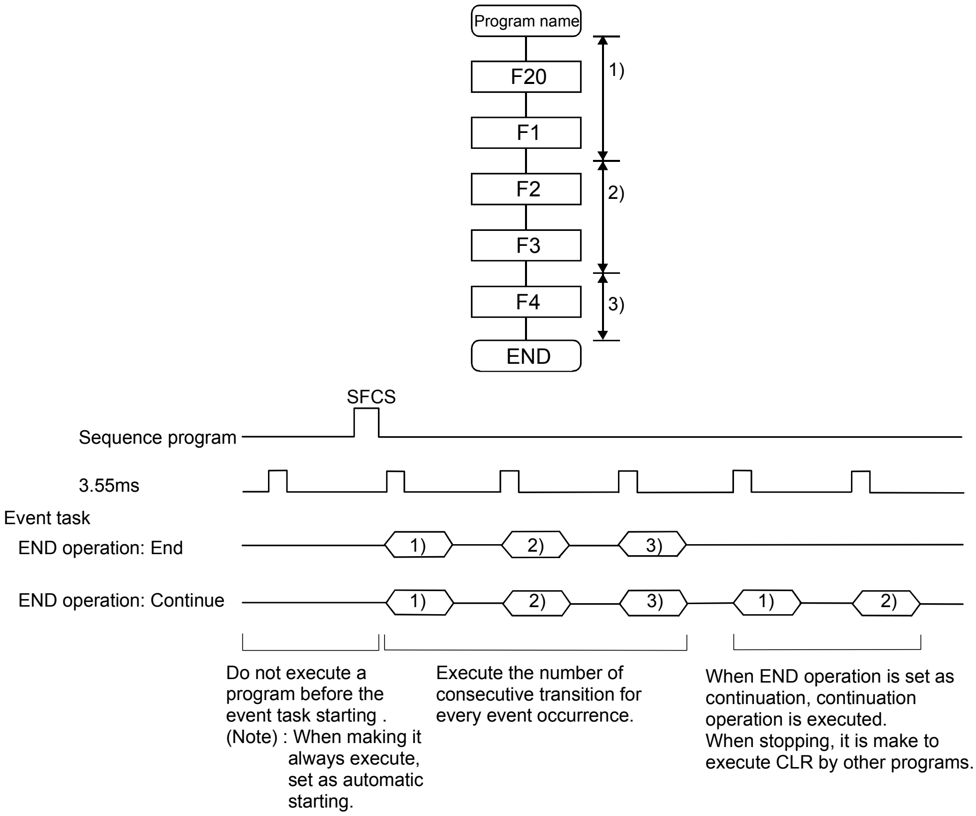
Among 16 points of the QI60 (16-point interrupt module) loaded in the motion slot, the Motion SFC program is run when the input set for an event task turns on.

(c) PLC interrupt

The Motion SFC program is executed when the D(P).GINT instruction is executed in the sequence program.

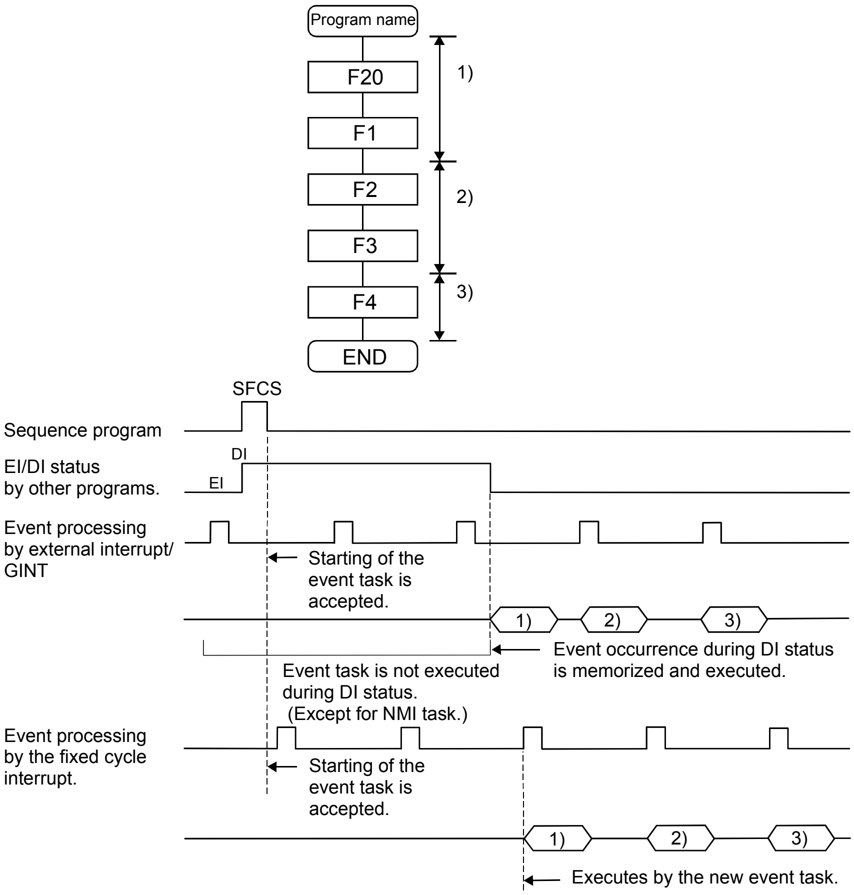
<Example 1> Operation for fixed cycle task (3.55 [ms])

- Number of consecutive transitions is set to "2".



<Example 2> Operation for PLC interrupt by D(P).GINT

• Number of consecutive transitions is set to "2".



[Points]

(a) Multiple events can be set to one Motion SFC program. However, multiple fixed cycles cannot be set.

(b) Multiple Motion SFC programs can be executed by one event.

(c) Motion control steps cannot be executed during the event task.

(d) The event task cannot be executed when it is disabled by the normal task. The event that occurred during event task disable is executed the moment the event task is enabled.

[Errors]

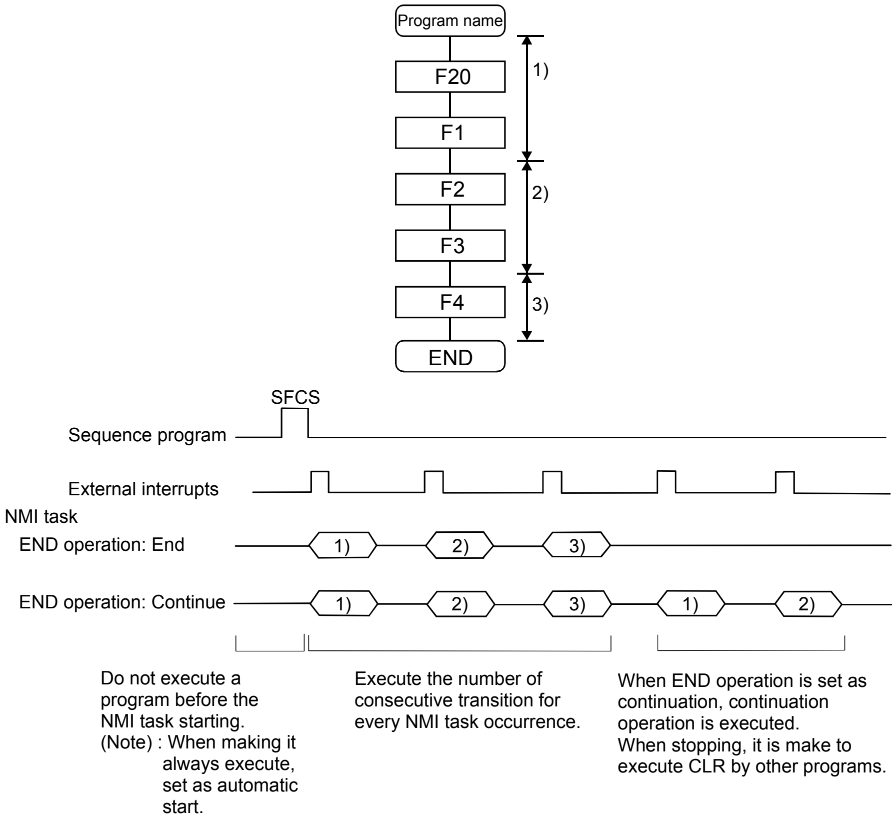
When the motion control step is executed by the Motion SFC program set to the event task, the Motion SFC error (error code: 16113) occurs and stops the Motion SFC program running.

(3) NMI task operation

[Operations]

The Motion SFC program is executed when the input set to the NMI task factor among external interrupts (16 points of QI60) turns on.

- Number of consecutive transitions is set to "2".



[Points]

(a) NMI task has the highest priority among the normal, event and NMI tasks.

(b) If the event task is disabled (DI) by the normal task, the interruption of the NMI task is executed, without being masked.

(c) When parallel branching during NMI task execution, the additional route from the branch starts execution from the next interrupt occurrence.

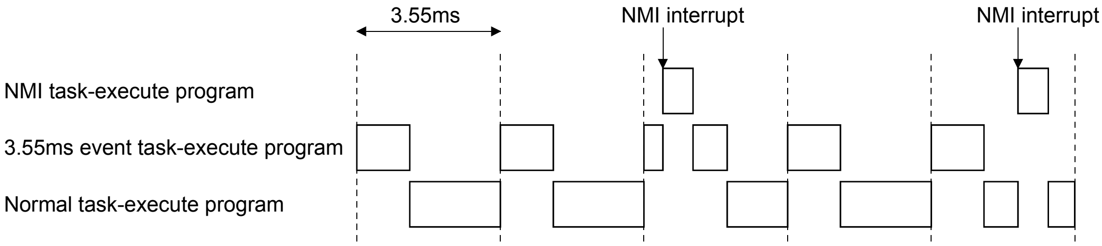
[Errors]

The motion control step is executed during NMI task.

If the motion control step is executed during NMI task, the Motion SFC error (error code: 16113) occurs and stops the Motion SFC program.

9.3 Execution Status of the Multiple Task

Execution status of each Motion SFC program when the Motion SFC program is executed multiple tasks is shown below.



When there are programs which are executed by the NMI task, 3.55ms fixed-cycle even task with a program to run by the NMI task, and the normal task like a chart,

(1) The 3.55ms fixed-cycle event task is executed at intervals of 3.55ms;

(2) The NMI task is executed with the highest priority when an NMI interrupt is input; and

(3) The normal task is executed at free time.
as shown above.

[Points]
One Motion SFC program can be executed partially by another task by setting the area to be executed by another task as a subroutine and setting a subroutine running task as another task.
<Example>

No. 0 Main Motion SFC program	Normal task
No. 1 Subroutine	Event task (3.55ms cycle)

⚠CAUTION

- If too many NMI tasks and event tasks are executed, an operation cycle over occurs, or the majority normal tasks are not executed and a WDT error may occur.

9.4 How to Start the Motion SFC Program

The Motion SFC program is executed during PLC ready flag (M2000) is on.
The Motion SFC program may be started by any of the following three methods.

(1) Automatic start
(2) Start from the Motion SFC program
(3) Start by Motion dedicated PLC instruction (D(P).SFCS) from PLC

Set the starting method in the program parameter for every Motion SFC program.
Refer to Section "9.12 Program Parameters" for parameter setting.

9.4.1 Automatic start

[Operations]
An automatic start is made by turning PLC ready flag (M2000) on.

9.4.2 Start from the Motion SFC program

[Operations]
A start is made by executing a subroutine call/start step in the SFC program.
Refer to "Chapter 4 MOTION SFC PROGRAMS" for details of the subroutine call/start step.

9.4.3 Start by Motion dedicated PLC instruction from PLC (PLC instruction: D(P).SFCS)

The SFC program is started by executing the D(P).SFCS in the sequence program.
Refer to "Chapter 3 MOTION DEDICATED PLC INSTRUCTION" for details.

9.5 How to End the Motion SFC Program

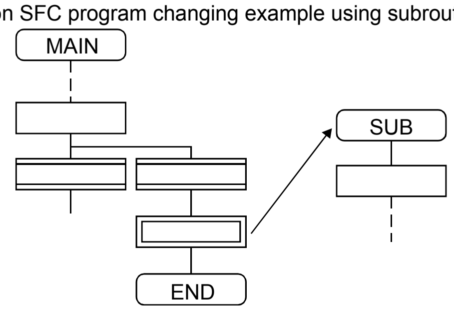
[Operations]
(1) The Motion SFC program is ended by executing END set in itself.
(2) The Motion SFC program is stopped by turning off the PLC ready flag (M2000).
(3) The program can be ended by the clear step.
Refer to Section "4.5.4 Clear step" for details of the clear step.

[Point]
(1) Multiple ENDs can be set in one Motion SFC program.

9.6 How to Change from One Motion SFC Program to Another

Use a subroutine start to stop the Motion SFC program running and switch it to another Motion SFC program.

Motion SFC program changing example using subroutine start



9.7 Operation Performed at Multiple CPU system Power-Off or Reset

When the Multiple CPU system is powered off or reset operation is performed, Motion SFC programs run are shown below.

(1) When the Multiple CPU system is powered off or reset operation is performed, Motion SFC programs stop to execute.

(2) At Multiple CPU system power-on or key-reset, the contents of the motion registers #0 to #7999 are held. Initialize them in the Motion SFC programs as required.

(3) After Multiple CPU system power-on or reset processing, Motion SFC programs run is shown below.

- The SFC programs set to start automatically are run from the beginning by turning PLC ready flag (M2000) on in the sequence program.
- The other Motion SFC programs are also executed from the first at starting.

9.8 Operation Performed when CPU is Switched from RUN/STOP

When a RUN/STOP switch is operated, PLC ready flag (M2000) turns on/off in accordance with "Operation at STOP to RUN" of system basic setting.
Refer to Section "3.1.3 Individual parameters" of the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for the details of "Operation at STOP to RUN".
And, refer to the next section for PLC ready flag (M2000) off/on.

9.9 Operation Performed when PLC Ready Flag (M2000) Turns OFF/ON

This section explains about the turns off/on of PLC ready flag (M2000).
The on/off condition of PLC ready flag (M2000) differences in "Operation at STOP to RUN" of system basic setting.
Refer to Section "3.1.3 Individual parameters" of the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for details.

[M2000 OFF → ON]
If there is no fault when PLC ready flag (M2000) turns off to on, the PCPU READY complete flag (SM500) turns on.
When this PCPU READY complete flag (SM500) turns on, Motion SFC programs can be executed.
An automatic start Motion SFC program starts execution from the first.

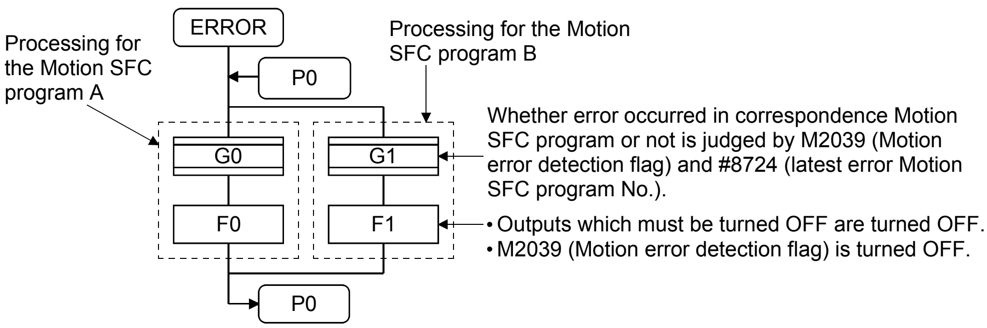
[M2000 ON → OFF]
When PLC ready flag (M2000) turns off, Motion SFC programs stops to execute and the PCPU READY complete flag (SM500) turns off.
Since actual outputs PY has whole point turn off.

POINT

When the PLC ready flag (M2000) turns off, Motion SFC programs stop but actual outputs PY in the Motion SFC programs do not turn off.

9.10 Operation at the Error Occurrence

Outputs are held if Motion SFC programs stop due to error occurrence.
To turn off outputs at error occurrence, executes the following Motion SFC program.



9.11 Task Parameters

No.	Item		Setting item	Initial value	Remark
1	Number of consecutive transitions	Normal task (Normal task common)	1 to 30	3	These parameters are imported at leading edge of PLC ready flag (M2000) and used for control thereafter. When setting/changing the values of these parameters, turns the PLC ready flag (M2000) off.
2	Interrupt setting		Set whether the event task or NMI task is used for external interrupt inputs (I0 to I15).	Event task	
3	Limited count for repeat control Ver.!	Normal task	1 to 100000	1000	
		Event task	1 to 10000	100	
		NMI task	1 to 10000	100	

(1) Number of consecutive transitions

[Description]

With "execution of active step → judgment of next transition condition → transition processing performed when condition enables (transition of active step)" defined as a single basic operation of the Motion SFC program execution control in the execution cycle of the corresponding task, this operation is performed for the number of active steps to terminate processing once. And the same operation is processed continuously in the next cycle.
In this case, the transition destination step is executed in the next cycle when the transition condition enables.

Consecutive transition control indicates that transition destination steps are executed one after another in the same one execution cycle when their transition conditions have enabled (single basic operation is performed consecutively).
In this case, the number of consecutive transitions can be set.
Controls in common to the Motion SFC programs executed by normal tasks.

POINT

Set the number of consecutive transitions to the Motion SFC programs executed by event and NMI tasks for every program.

[Errors]

These parameters are imported and checked at leading edge of PLC ready flag (M2000).
When the value that was set is outside the setting range, the Motion SFC parameter error (error code: 17000) will occur and the initial value is used to control.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(2) Interrupt setting

[Description]

Set whether 16 interrupt input points (I0 to I15) of the QI60 interrupt module loaded in the motion slot are used as NMI or event task inputs.
Setting can be made freely per point.
All points default to event tasks.

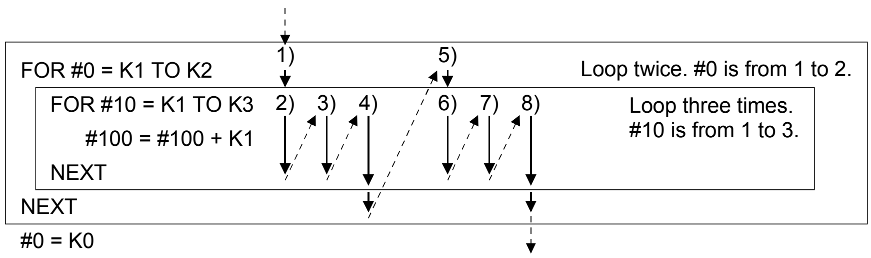
[Errors]

None.

(3) Limited count for repeat control Ver.!

[Description]

Operation control program requires longer processing time if the operation control program or transition program has more repeat control instructions (FOR - NEXT). The longer processing time may cause the increase of main cycle and the operation cycle over in event task/NMI task, which is prevented by setting "Limited count for repeat control". Set "Limited count for repeat control" in every normal task, event task and NMI task.
If the repeat control instruction (FOR - NEXT) is executed over "Limited count for repeat control " in an operation control program or a transition program, a Motion SFC error will occur (error code: 16208), and the corresponding Motion SFC program No. will stop to execute. For the subroutine called program, the call source program also stops to execute.
The repeat control instruction is executed once when the repeat control is judged to continue at FOR instruction execution (when the condition is true).
In the program shown below, each block is executed as the arrow indicates, and the repeat control instruction is executed eight times.



[Errors]

None.

These parameters are imported and checked at leading edge of PLC ready flag (M2000).
When the value that was set is outside the setting range, the initial value is used to control.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

9.12 Program Parameters

the following parameters for every Motion SFC program.

<table>
<tr><th>No.</th><th>Item</th><th>Setting range</th><th>Initial value</th><th>Remark</th></tr>
<tr><td>1</td><td>Start setting</td><td>Automatically started or not</td><td>Not setting</td><td rowspan="7">These parameters are imported at leading edge of PLC ready flag (M2000) and used for control there after.
When setting/changing the values of these parameters, turn PLC ready flag (M2000) off.</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3">Execute task</td><td>It is only one of normal, event and NMI tasks</td><td>Normal task</td></tr>
<tr><td>When you have set the event task, further set the event which will be enabled.
Always set any one of the following 1 to 3.
1. Fixed cycle
It is one of 0.22ms QDS, 0.44ms, 0.88ms, 1.77ms, 3.55ms, 7.11ms and 14.2ms or nothing.
2. External interrupt (make selection from those set to event task)
Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15.
3. PLC interrupt
Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15.
The above 1 to 3: Multiple setting is possible.
The same event can be shared among multiple Motion SFC programs.</td><td rowspan="2">None</td></tr>
<tr><td>When you have set the NMI task, further set the interrupt input which will be enabled.
1. External interrupt (make selection from those set to NMI task)
Multiple interrupt can be set from among I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15.</td></tr>
<tr><td>3</td><td>Number of consecutive transitions</td><td>1 to 10
Set the number of consecutive transitions toward the program set to the event or NMI task.</td><td>1</td></tr>
<tr><td>4</td><td>END operation</td><td>End/continue
Set the operation mode of the END step toward the program set to the event or NMI task.</td><td>End</td></tr>
<tr><td>5</td><td>Executing flag</td><td>None/Bit device
Set the bit device turned ON while executing Motion SFC program.
X0 to X1FFF (Note-1)
Y0 to Y1FFF
M0 to M8191
B0 to B1FFF
U□\G10000.0 to U□\G(10000+p-1).F (Note-2)
(Self CPU only)</td><td>None</td></tr>
</table>

(Note-1): The range of "PXn+4 to PXn+F" cannot be used (fixed at 0) for the input device (PXn+0 to PXn+F) allocated to the built-in interface in Motion CPU (DI). (n: First input No.) QDS

(Note-2): "p" indicates the user setting area points of the Multiple CPU high speed transmission area for the each CPU.

POINT

(1) The settings of "END operation" are invalid for the subroutine called program. "END operation" is controlled as "end".
(2) Refer to Chapter 2 of the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for the user setting area points of the Multiple CPU high speed transmission area.

(1) Start setting

[Description]

The following control is changed by "automatically started or not" setting.

- Program run by normal task

<table>
<tr><th>No.</th><th>Item</th><th>When "automatically started"</th><th>When "not automatically started"</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Start control</td><td>In the main cycle after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the normal task.</td><td>The program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.
• When started by the Motion SFC start request (D(P).SFCS)
In the main cycle after execution of the Motion SFC start request (D(P).SFCS), the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the normal task.
• When subroutine started
In the (next) main cycle after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the normal task.
• When subroutine called
The program is executed in the same cycle from the first step.</td></tr>
<tr><td colspan="2">After that, the program is executed continuously by the number of consecutive transitions of the normal task in the Motion CPU main cycle. (The settings of "executed task" and "number of consecutive transitions" of the subroutine called program are invalid. It is controlled as the normal task.)</td></tr>
<tr><td>2</td><td>END control
END</td><td colspan="2">Ends the self program.
Again, the program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.</td></tr>
</table>

- Program run by event task

<table>
<tr><th>No.</th><th>Item</th><th>When "automatically started"</th><th>When "not automatically started"</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Start control</td><td>At occurrence of a valid event after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.</td><td>The program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from within the Motion SFC program.
• When started by the Motion SFC start request (D(P).SFCS)
At occurrence of an event after execution of the Motion SFC start request (D(P).SFCS), the program is run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.
• When subroutine started
At occurrence of an event after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the corresponding program.
• When subroutine called
The program is executed immediately from the first step.</td></tr>
<tr><td colspan="2">The program is executed continuously by the number of consecutive transitions of the corresponding program each time an event occurs. (The subroutine called program is controlled in accordance with the "executed task" and "number of consecutive transitions" of the call source program.)</td></tr>
<tr><td>2</td><td>END control
END</td><td colspan="2">As specified for END operation.</td></tr>
</table>

- Program run by NMI task

<table>
<tr><th>No.</th><th>Item</th><th>When "automatically started"</th><th>When "not automatically started"</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Start control</td><td>At occurrence of a valid event after PLC ready flag (M2000) ON, the program is executed from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.</td><td>The program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from within the Motion SFC program.
• When started by the Motion SFC start request (D(P).SFCS)
At occurrence of an event after execution of the Motion SFC start request (D(P).SFCS), the program is run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program.
• When subroutine started
At occurrence of an event after execution of GSUB, the program is executed from the first step in accordance with the number of consecutive transitions of the corresponding program.
• When subroutine called
The program is executed immediately from the first step.</td></tr>
<tr><td colspan="2">The program is executed continuously by the number of consecutive transitions of the corresponding program each time an event occurs.</td></tr>
<tr><td>2</td><td>END control
END</td><td colspan="2">As specified for END operation.</td></tr>
</table>

[Errors]
None.

POINT
In the case of the program which is executed by the normal task, write the program so that it is not ended by $\boxed{\text{END}}$ but it returns to the starting step by a jump when starting of the automatically from an initial again.

(2) Execute task

[Description]
Set the timing (task) to execute a program.
Specify whether the program will be run by only one of the "normal task (main cycle), event task (fixed cycle, external interrupt, PLC interrupt) and NMI task (external interrupt)".

When the event task is set, multiple events among the "fixed cycle, external interrupt (for event task) and PLC interrupt".
However, multiple fixed cycles cannot be set toward one Motion SFC program.
<Example> Interrupt setting: Inputs for event task I6, I7, I8, I9, I10, I11, I12, I13, I14 and I15
Motion SFC program No. 10 – event : Fixed cycle (3.55ms)
Motion SFC program No. 20 – event :
Fixed cycle (1.77ms) + external interrupt (I6)
Motion SFC program No. 30 – event :
External interrupts (I7, I15) + PLC CPU interrupt

When the NMI task is set, multiple interrupt inputs among the external interrupts (for NMI task) can be set.

<Example> Interrupt setting: Inputs for NMI task I0, I1, I2, I3, I4, I5
Motion SFC program No. 10 – NMI : I0
Motion SFC program No. 20 – NMI : I1 + I2
Motion SFC program No. 30 – NMI : I5

[Errors]
This program parameter is imported at leading edge of PLC ready flag (M2000), and is checked at starting of the Motion SFC program (automatic start, start from PLC or subroutine start).
When the value is illegal, either of the Motion SFC parameter error (error code: 17010) will occur and it is controlled with initial value.

POINT
(1) Since the execute task can be set for every Motion SFC program No., multiple programs need not be written for single control (machine operation) to divide execution timing-based processes. It can be achieved easily by subroutine starting the areas to be run in fixed cycle and to be run by external interrupt partially in the Motion SFC program run by the normal task. (2) Set a large fixed cycle than the motion operation cycle after confirming the motion operation cycle.

(3) Number of consecutive transitions

[Description]

Set the number of consecutive transitions to program executed by the event or NMI task for every program.

Refer to Section "9.11 Task Parameters" for number of consecutive transitions.

[Errors]

This program parameter is imported at leading edge of PLC ready flag (M2000), and is checked at starting of the Motion SFC program (automatic start, start from PLC or subroutine start).

When the value is illegal, either of the Motion SFC parameter error (error code: 17001, 17002) will occur and it is controlled with initial value.

(4) END operation

[Description]

Set the operation at execution of the END step toward the program executed by the event or NMI task.

This varies the specifications for the following items.

- Program run by NMI task

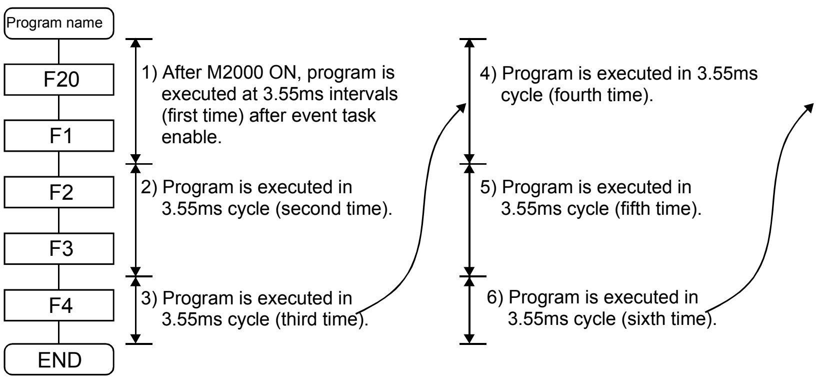
No.	Item	When "ended"	When "continued"
1	Control at END execution	Ends the self program.	Ends to execute the self program with this event/interrupt.
2	Restart after END execution	Again, the program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.	Restarted at occurrence of the next event/interrupt, and run from the initial (first) step in accordance with the number of consecutive transitions of the corresponding program. After that, at occurrence of an event/interrupt, the program is executed in accordance with the number of consecutive transitions of the corresponding program.
3	Restart after end by clear step CLR	Again, the program is started by the Motion SFC start request (D(P).SFCS) from the PLC or by a subroutine call/start (GSUB) made from the Motion SFC program.	

POINT

The END operation of subroutine called program is controlled as an "end".

- The following operation example assumes that the END operation is "continued."

  Program parameters
  - Automatically started
  - Execute task = event 3.55ms
  - Number of consecutive transitions = 2
  - End operation "continued"



(5) Executing flag

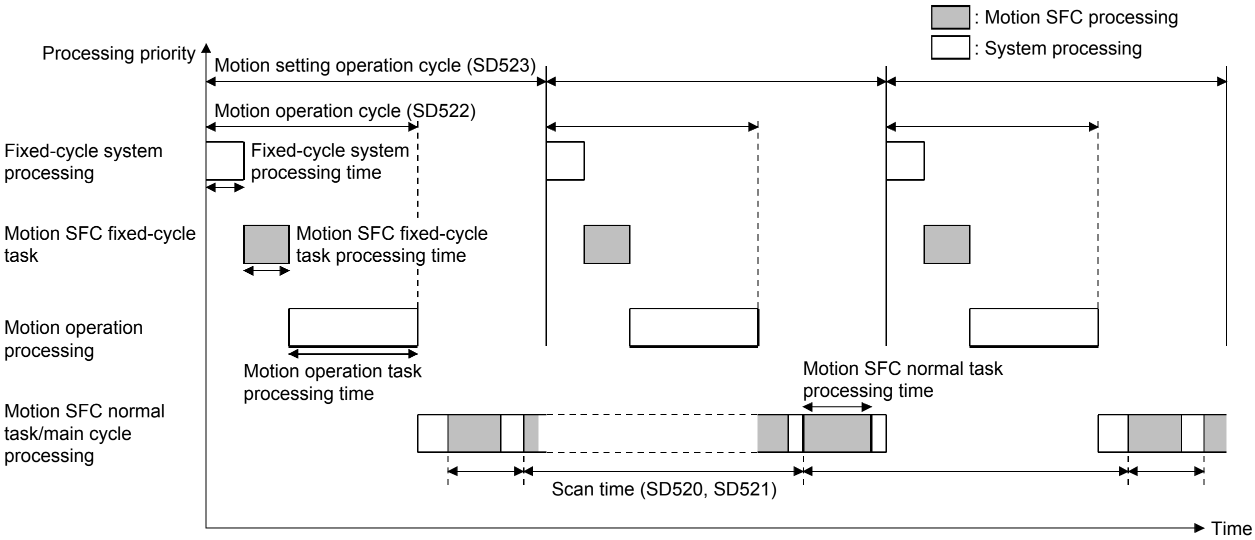
The set bit device turns ON by Motion SFC program start, and it turns OFF by program end.

9.13 Task and Interrupt Processing

In the Motion CPU, the required operations over a fixed cycle are divided into tasks. Depending on the Motion CPU internal processing timing, the interrupt processing can affect tasks, therefore programs need to be designed with care.

(1) Processing timing and processing time monitor device

The Motion CPU internal processing timing and corresponding processing time monitor devices are shown in the following diagram.



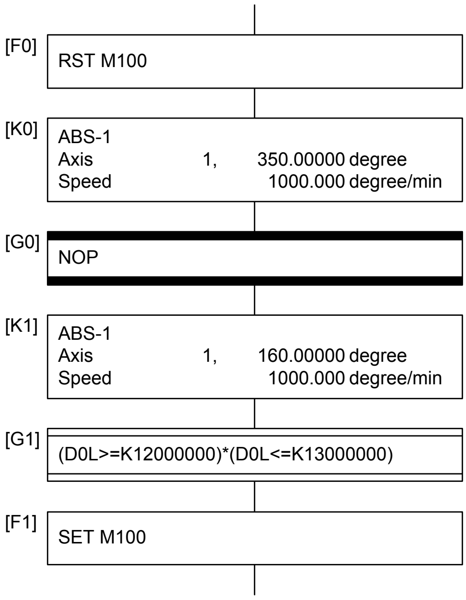
(2) Interrupt processing

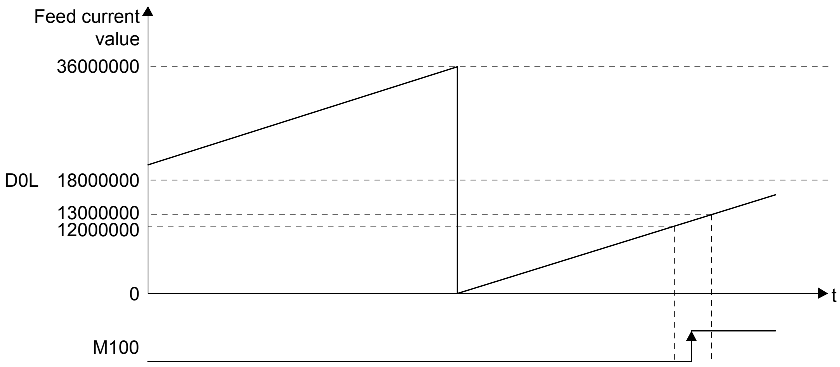
In the Motion CPU, fixed-cycle system processing, Motion SFC fixed-cycle task, and Motion operation processing are executed with priority over main cycle processing. Therefore, if Motion setting operation cycle (SD523) is exceeded in the middle of main cycle processing, the main cycle processing is interrupted by the execution of the next Motion operation cycle processing. Main cycle processing restarts when the interrupting Motion operation cycle processing is completed.

Because this interrupt processing is also executed in the middle of operations of the Motion SFC program, depending on the program contents, the Motion SFC program may not execute correctly.

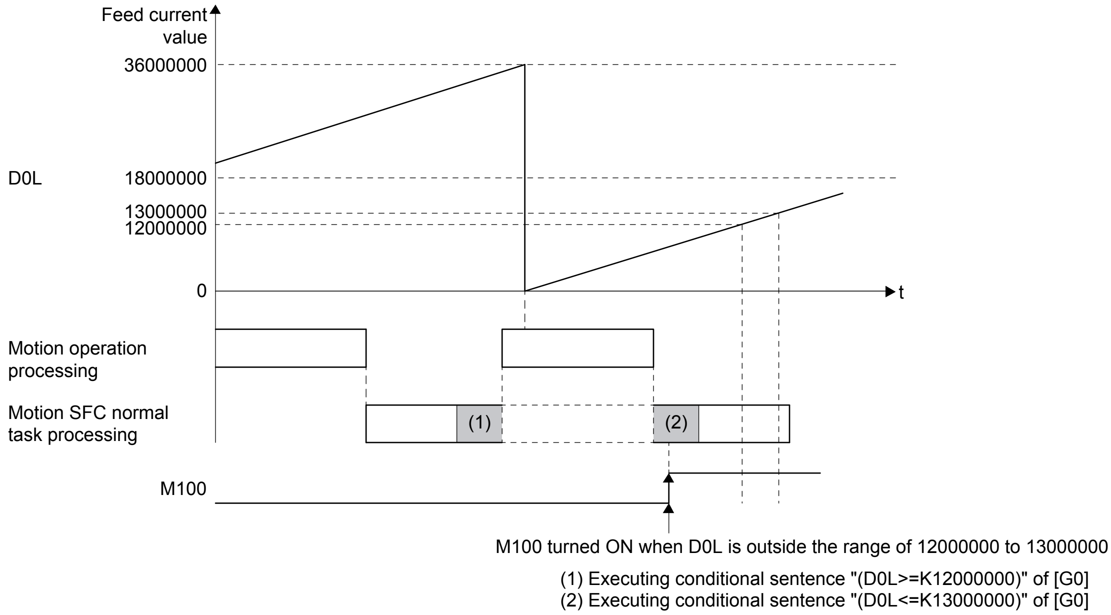
(3) Operation example

Axis 1 of the Motion SFC program below moves to "350.00000[degree] → 0[degree] → 160.00000[degree]". When the axis 1 feed current value (D0L) moves in the range of 120.00000 to 130.00000[degree], M100 is turned ON.

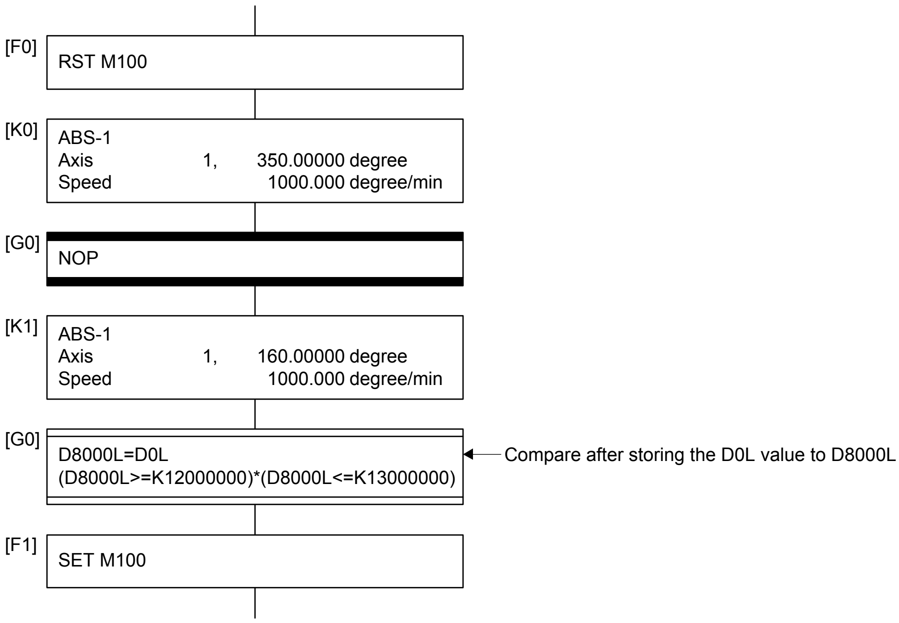




However, when the timing of axis 1 moving from 359.99999[degree] to 0[degree] coincides with the timing of the interrupt execution processing in the middle of [G1] processing in Motion SFC normal task processing, an unintended operation may occur.



When using the device whose value changes by the Motion operation cycle in a conditional expression, store the value to a device and make the program compare values as shown below.



10. ONLINE CHANGE IN THE MOTION SFC PROGRAM

10.1 Online Change in the Motion SFC Program

This function is used to write to the Motion SFC program to the SRAM built-in Motion CPU during the positioning control (7-segment LED : Steady "RUN" display).
Program correction and a check of operation can be executed repeatedly at the Multiple CPU system start.

Data in which online change is possible are shown below.

Applicable data		Online change	Remarks
System setting data	System setting	×	
	Servo setting data	×	
Motion SFC program	Motion SFC parameter	×	
	Motion SFC chart	○	Online change is possible for the only program during stop.
	Operation control step (F/FS)	○	
	Transition (G)	○	
	Servo program (K)	○	Online change of mode allocation is not possible. (SV22)
Mechanical system program (SV22)		×	
Synchronous control parameter (SV22) QDS Ver.!		×	
Cam data (SV22)		×	

○ : Possible × : Not possible

POINT

(1) Program writing is executed during the positioning control in the online change. Be safely careful enough for work.

(2) Programs writing to the SRAM built-in Motion CPU at the mode operated by ROM in the online change. If the online change is executed at the mode operated by ROM, it returns to the contents of program written in the FLASH ROM built-in Motion CPU by the next power ON or reset of the Multiple CPU system.

(3) If the online change is executed simultaneously to one Motion CPU from the multiple personal computers, a program writing may not be executed. Please do not perform.

(4) If the online changes are executed by other personal computer during the following operation with MT Developer2, injustice of a monitor value and operation failure may occur. Please do not perform.
- Monitor mode of the Motion SFC program
- Debug mode of the Motion SFC program
- Test mode

(5) If the online change of Motion SFC chart added newly is executed, since the online change of Motion SFC parameter cannot be executed, it operates as the normal task (default value).

(6) When using the SV22, the program range of [Mode allocation] of the servo program editor screen cannot be changed by online change. If the online change is executed by changing the virtual mode program, or command generation axis program QDS in the [Mode allocation], the contents of change are not reflected, and the online change is cancelled.

(7) If the cables between the personal computer and PLC CPU module fall out, or the power supply of the Multiple CPU system turns OFF or resets, the program is corrupted.
Write the program again by the data writing of MT Developer2.

(8) The online change only writes when the program being operated by the Motion CPU and the MT Developer2 project data (before change) match. Before writing perform a check, and when data does not match, cancel the online change.

10.1.1 Operating method for the online change

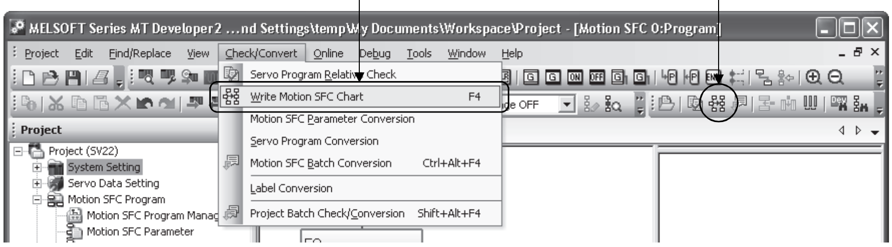
Select the "Online change OFF/ON" with the online change setting screen displayed on [Tools] menu – [Online Change Setting]" of MT Developer2.
The methods for online change of Motion SFC program are shown below.

Target data of online change	Operation for online change
Motion SFC chart	• Select [Check/Convert] to [Write Motion SFC Chart] of menu bar. • Click [Write Motion SFC Chart] of toolbar.
Operation control program (F/FS)	Click [Convert] of operation control program/transition program editor screen.
Transition program (G)	
Servo program (K)	Click [Convert] of servo program editor screen.

(1) Online change of the Motion SFC chart

Online change of the Motion SFC chart in edit is executed by selecting button or menu of toolbar.
Online change is possible to the Motion SFC program during stop.
If the online change is made to the Motion SFC program during execution, an alarm message indicates. (Execution/stop state of the Motion SFC program can be checked with the program batch monitor.)
If the start request is made to the Motion SFC program during online change, the Motion SFC start error (error code16007: online change) will occur and the Motion SFC program does not start.

[Write Motion SFC Chart] menu
[Write Motion SFC Chart] icon

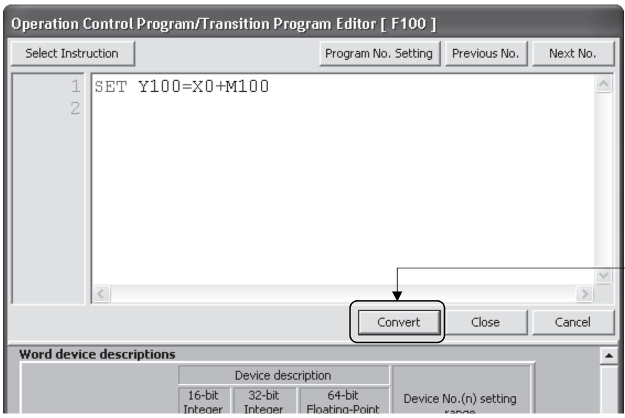


(2) Online change of the operation control/transition program

Online change of the operation control/transition program during edit is executed by selecting the [Convert] button.

Online change is possible to the operation control/transition program during execution.

A program that the online change was made is executed from the next scan.



[Convert] button

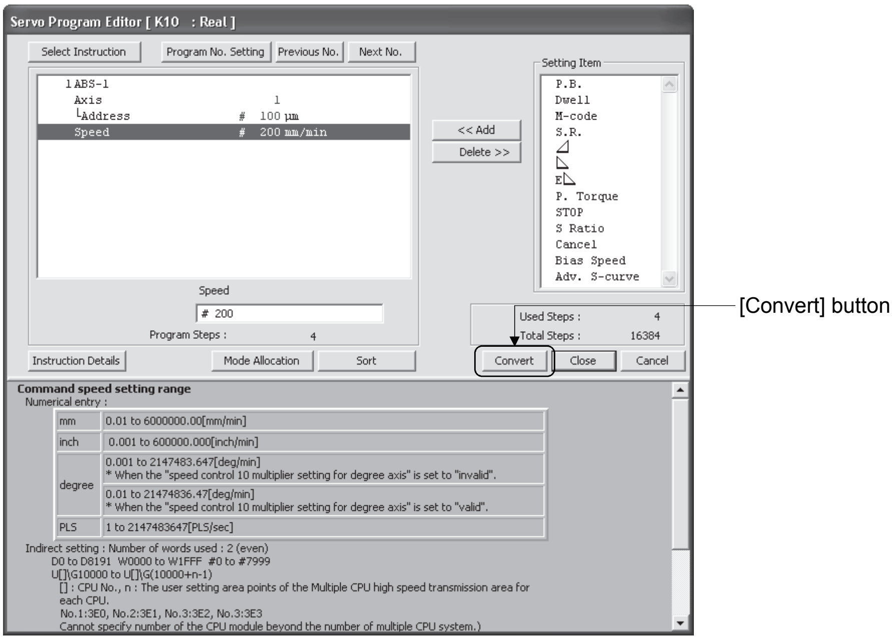
Operations for which made the online change to the operation control/transition program during execution in the following conditions are shown below. Be careful to execute the online change in the following conditions.

Program	Condition	Operation
FSn Gn or FSn Gn	• Online change of the FSn operation control program is executed during FSn execution in the state of waiting for the completion of condition for Gn.	• After completion of online change, the FSn repeats the operation control program that the online change was made until the completion of condition for Gn.
Gn or Gn	• Online change of the Gn program is executed in the state of waiting for the completion of condition for Gn. (The conditional sentences of program to write are except the TIME instruction.)	• After completion of online change, the Gn does not transit to the next step until the completion of condition for program that the online change was made.
Gn or Gn	• Online change of the Gn program including the TIME instruction is executed in the state of waiting for the completion of condition for Gn.	• After completion of online change, Gn is ended regardless of the waiting time of TIME instruction and the next step is executed.
Kn or Gn	• Online change of the Gn program during the servo program execution for Kn.	• After execution of servo program, the program of changed Gn is executed.

(3) Online change of the servo program

Online change of the servo program during edit is executed by selecting the [Convert] button.

Online change is possible to the servo program during execution.
A program that the online change was made is executed at the next servo program start.



Operations for which made the online change to the servo program in the following conditions during execution are shown below. Be careful to execute the online change in the following conditions.

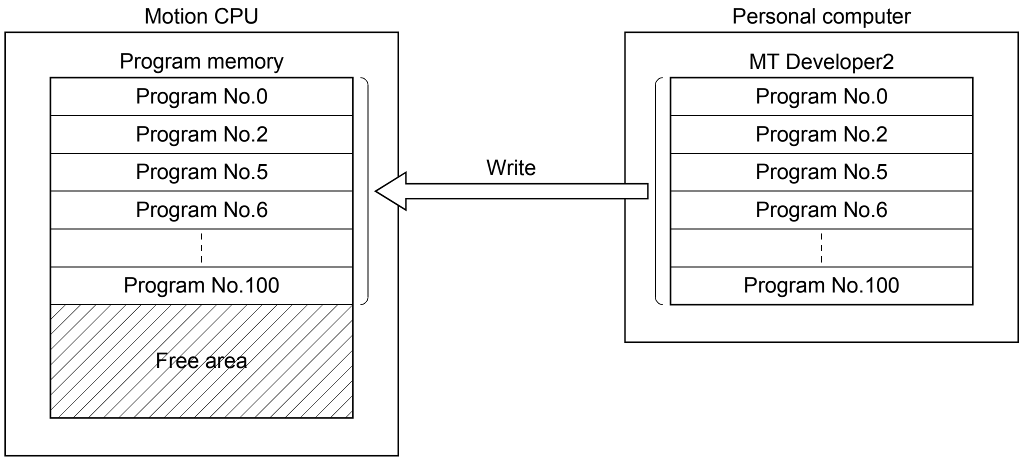
Program	Condition	Operation
ON bit device Kn or OFF bit device Kn	• Online change of the servo program Kn at the WAITON or after WAITOFF is executed in the state of waiting for the completion of condition for WAITON/WAITOFF.	• After completion of condition for WAITON/WAITOFF, the servo program before the online change is started. • The servo program that the online change was made is executed at the next servo program start.
Gn Kn or Gn Kn	• Online change of the servo program Kn after Gn is executed in the state of waiting for the completion of condition for Gn.	• After completion of condition for Gn, the servo program that online change was made is executed.

10.1.2 Writing of program

The outline operations to write the program from MT Developer2 to the program memory of Motion CPU are described.

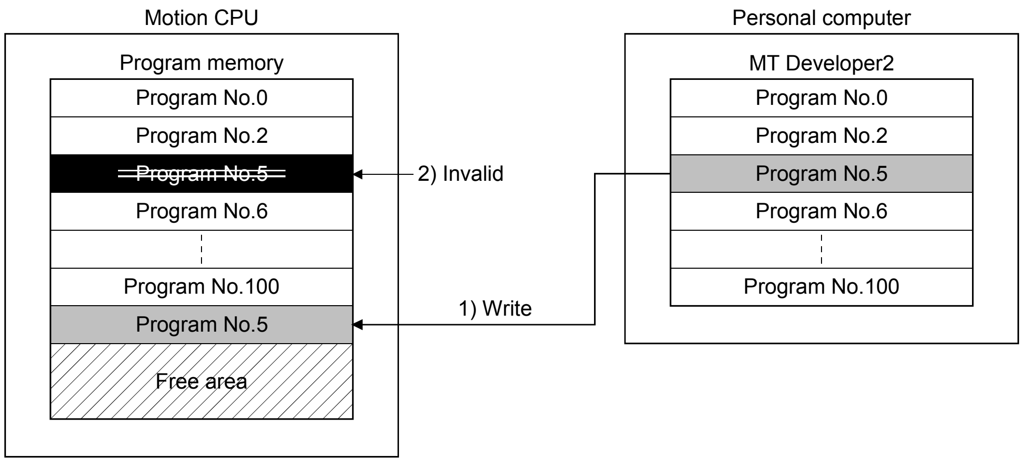
(1) Writing of program by the writing operation of MT Developer2

(a) The programs are stored in the program memory of Motion CPU stuffing to the front for every kind.

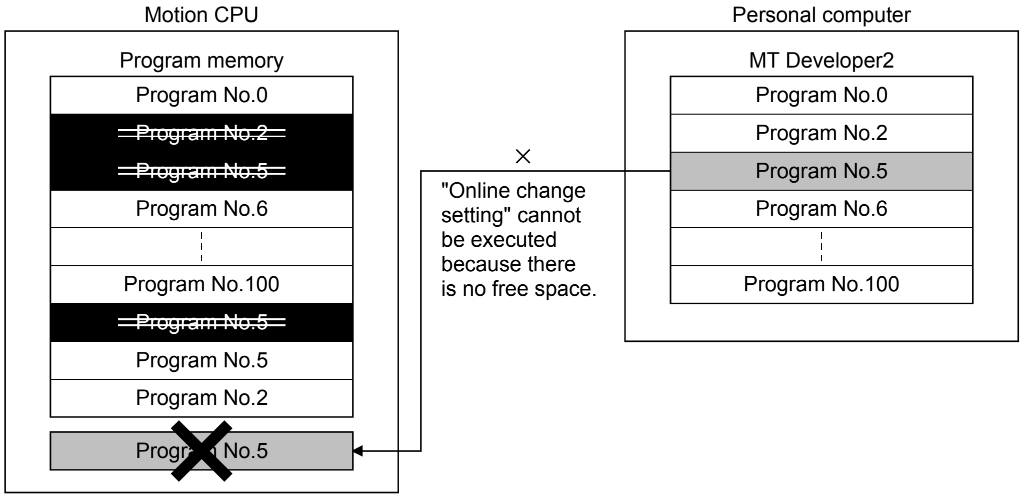


(2) Writing of program by the online change operation of MT Developer2

(a) After online change, a program to execute the online change is stored in the free area after the program stored previously. (Refer to 1))
After that, the program written in previously is made invalid and the new program is made valid. (Refer to 2))



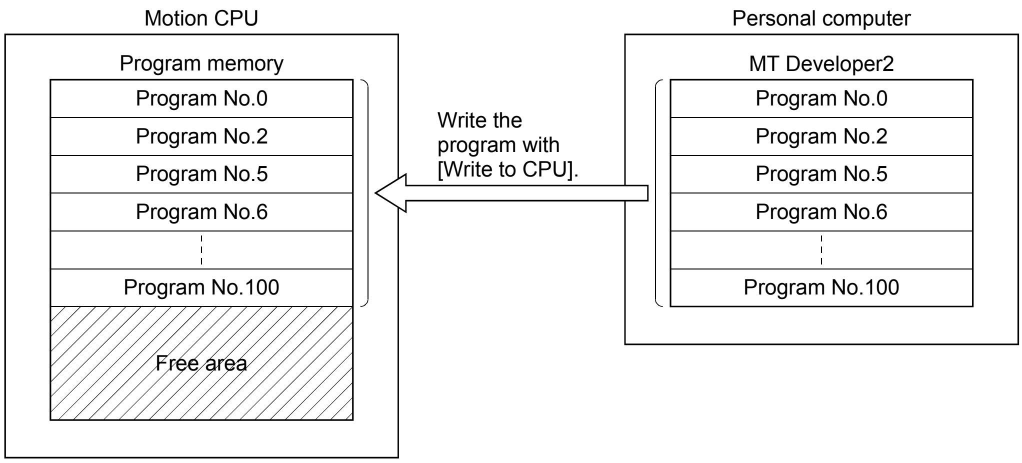
(b) If the online change is executed repeatedly, the free space in program memory is lost and the online change may not be executed. In this case, an error message is displayed by MT Developer2 at the online change, and "Online change OFF" is set.



(c) In the case of (b), arrange to stuff to the front the invalid programs. Operation procedures is shown below.

1) Execute batch conversion (Motion SFC chart, operation control/transition program) or sort (servo program) using MT Developer2.
2) Make to stop status the Motion CPU, and then execute writing operation of MT Developer2.

Refer to the help of MT Developer2 for details of the operation procedures.



MEMO

11. USER FILES

This section describer the user file list and directory structure.

11.1 Project

There are "single file format" of treating a project as one file and "workspace format" of managing multiple projects in a workspace for user file.

(1) Single file format

Every project is treated as one file (file extension: ∗.mtw).

(2) Workspace format

Every user file is managed in a workspace, and multiple projects can be stored in a single workspace. When a project is initially saved, a "Workspace name" folder and a "Project name" folder are created in the place designated in the [Save Folder Path] field and uses files are stored. (Refer to next page.)

POINT

- The number of characters that can be set for "Save folder path" + "Workspace name" + "Project name" is within 200 characters.

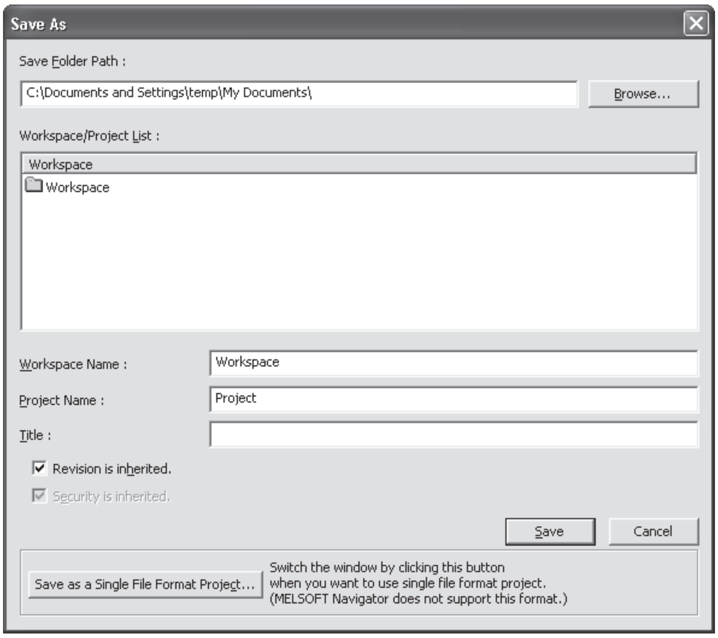
<Example>

C:\Documents and Settings\Administrator\My Documents\Workspace\Project

Save folder path

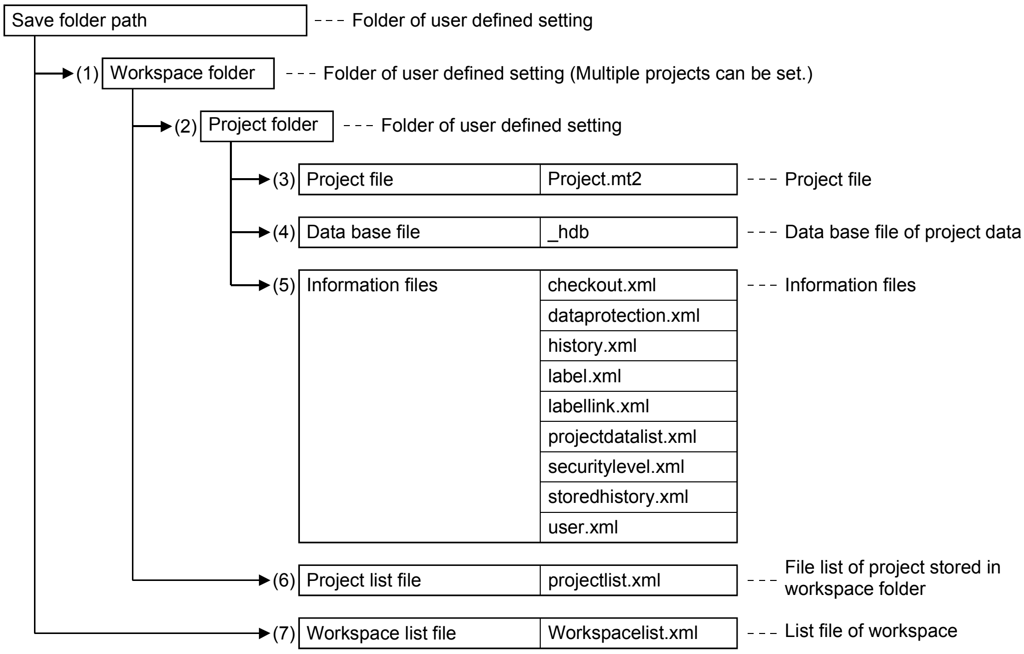
Workspace name

Project name



11.2 User File List

Folder configuration of user file to be saved in the workspace format is shown below.

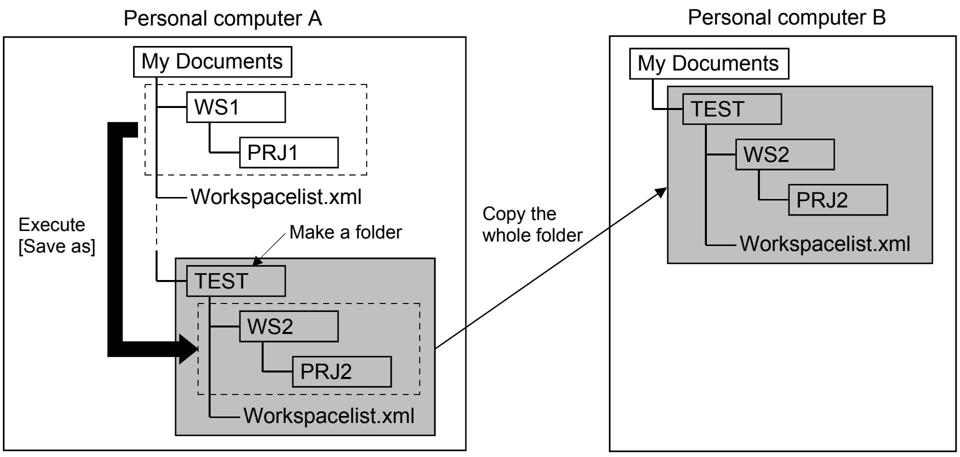


POINT

(1) Double-clicking on the "Project file (Project.mt2)" opens a project.
(2) A "Workspace" folder and a "Workspace list file (Workspacelist.xml)" that composes a workspace should be stored in the same folder.
(3) The procedure for using on another PC a project stored in a workspace folder is shown below.

<Example> Save the project "Workspace name: WS1, Project name: PRJ1" of PC A as "Workspace name: WS2, Project name: PRJ2" and use it on PC B.

1) Create any folder (ex. TEST) on PC A.
2) Read a project.
3) Save ([Save as]) the read project in the folder created by procedure 1).
4) Close MT Developer2.
5) Copy the whole saved folder to electronic media (USB memory etc.).
6) Copy the data in the electronic media PC B and read the project.



MEMO

12. ERROR CODE LISTS

When an error occurs while the Motion CPU is running, the error information is stored in the Motion error history devices (#8640 to #8735), special relay (SM) and special register (SD).

12.1 Confirming Error Code

When an error occurs, the error code and error contents can be read using MT Developer2 or GX Works2/GX Developer.

The occurrence data of the Motion error history uses a watch function with the internal Motion CPU.
Make the set of the clock data and the clock data read request (SM801) by user programs.
As for the self-diagnosis error code, confirmation can be done by the PC diagnosis of GX Works2/GX Developer.
Refer to the Operating Manual of GX Works2 or GX Developer and help of MT Developer2 for the operating method.

12.2 Motion Error Related Devices

(1) Motion error history devices (#8640 to #8735)

Eighth in the past (Seventh in the past to latest) error information are stored as a history. #8724 to #8735 are latest errors.
All errors, including the Motion SFC control errors and the minor, major, servo, servo program setting and mode changing errors are stored. At error occurrence, the "Motion error detection flag (M2039)" is also set.
Motion error history is backed-up.
Use the Motion error history clear request flag (M2035) or MT Developer2 to clear the Motion error history.
Also, the self-diagnostic errors of error code 10000 or less are stored in the Motion error history.

Refer to the following manuals for details of each error.

Error type	Reference manual
• Minor error • Major error • Servo error • Servo program setting error	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV13/SV22) Programming Manual (REAL MODE)
	Q173D(S)CPU/Q172D(S)CPU Motion controller (SV22) Programming Manual (VIRTUAL MODE)
	Q173DSCPU/Q172DSCPU Motion controller (SV22) Programming Manual (Advanced Synchronous Control)
• Self-diagnosis error	Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)

Table 12.1 Motion error history device

Error information								Signal name
Seventh in past	Sixth in past	Fifth in past	Fourth in past	Third in past	Second in past	First in past	Latest	
#8640	#8652	#8664	#8676	#8688	#8700	#8712	#8724	Error Motion SFC program No.
#8641	#8653	#8665	#8677	#8689	#8701	#8713	#8725	Error type
#8642	#8654	#8666	#8678	#8690	#8702	#8714	#8726	Error program No.
#8643	#8655	#8667	#8679	#8691	#8703	#8715	#8727	Error block No./Motion SFC list/ Line No./Axis No.
#8644	#8656	#8668	#8680	#8692	#8704	#8716	#8728	Error code
#8645	#8657	#8669	#8681	#8693	#8705	#8717	#8729	Error occurrence time (Year/month)
#8646	#8658	#8670	#8682	#8694	#8706	#8718	#8730	Error occurrence time (Day/hour)
#8647	#8659	#8671	#8683	#8695	#8707	#8719	#8731	Error occurrence time (Minute/second)
#8648	#8660	#8672	#8684	#8696	#8708	#8720	#8732	Error setting data information
#8649	#8661	#8673	#8685	#8697	#8709	#8721	#8733	Unusable
#8650	#8662	#8674	#8686	#8698	#8710	#8722	#8734	Error setting data
#8651	#8663	#8675	#8687	#8699	#8711	#8723	#8735	

The contents of Motion error history device error information are shown in Table 12.2.

Table 12.2 Motion error history device error information

Signal name	Description	
	Motion SFC control errors	Conventional errors
Error Motion SFC program No.	0 to 255: Motion SFC program No. in error -1 : Independent of Motion SFC program	-1
Error type	20 :F/FS 21 :G 22 :K or other (not any of F/FS, G and SFC chart) 23 :Motion SFC chart	2: Minor/major error (Command generation axis) (SV22 advanced synchronous control method) 3: Minor/major error 4: Minor/major error (virtual servomotor shaft) (SV22 virtual mode switching method) 5: Minor/major error (synchronous encoder shaft) (SV22) 6: Error detected in the servo amplifier (MR-J3-B) 7: Servo program setting error 8: Mode change error (SV22 virtual mode switching method) 9: Manual pulse generator axis setting error 10: Test mode request error 11: WDT error 13: Self-diagnostic error (Error code: 10000 or less) 14: System setting error, Motion slot fault detection 15: Error detected in the servo amplifier (MR-J4-B) 40: Error detected in the inverter 41: Error detected in the Nikki Denso servo driver 42: Error detected in the SSCNETⅢ/H head module 43: Error detected in the stepping driver 44: Operation error detected in the stepping driver 50: Safety observation error 51: Safety observation warning
Error program No.	0 to 4095: F/FS, G, K program No. 0 to 255 : GSUB program No. -1 : Independent of F/FS, G, K, GSUB	• Error type: "2", "3", "4" or "7" 0 to 4095: Servo program No. FFFFH : JOG operation FFFEH : Manual pulse generator FFFDH : Test mode (Home position return, servo diagnosis, servo startup) FFEFH : Synchronous control FFDFH : Speed control FFDEH : Torque control FFDDH : Continuous operation to torque control FF00H : Others • Error type: except "2", "3", "4" or "7" -1
Error block No./ Motion SFC list line No./axis No.	0 to 8191: F/FS or G program's block No. (line No.) when error type is "20" or "21" 0 to 8188: Motion SFC list line No. when error type is "23" -1 : Independent of block when error type is "22" or error type is "20" or "21"	1 to 32: Corresponding axis No. when error type is any of "2" to "6", "15", "40", "41", "43", "44" -1 : Others 1 to 8 : Corresponding SSCNETⅢ/H head module axis No. when error type is any of "42"

Ver.! : Refer to Section 1.3 for the software version that supports this function.

Table 12.2 Motion error history device error information (Continued)

<table>
<tr><th colspan="2" rowspan="2">Signal name</th><th colspan="2">Description</th></tr>
<tr><th>Motion SFC control errors</th><th>Conventional errors</th></tr>
<tr><td colspan="2">Error code</td><td>16000 and later</td><td>• Error type is followings;
"2" : Error code stored in D12602+20n or D12603+20n QDS
"3" : Error code stored in D6+20n or D7+20n
"4" : Error code stored in D802+10n or D803+10n
"5" : Error code stored in D1122+10n or D1123+10n (SV22 virtual mode switching method)
Error code stored in D13250+10n or D13251+10n (SV22 advanced synchronous control method) QDS
"6", "40" Ver.! "41" Ver.!, "43" QDS Ver.! : Error code stored in D8+20n
"7" : Error code stored in SD517
"8" : Error code stored in SD504
"9" or "10" : -1
"11" : Error code stored in SD512
"13" or "14" : Error code stored in SD0
"15" QDS : Error code stored in #8008+20n
"42" QDS Ver.! : Error code stored in alarm/warning number of SSCNETⅢ/H head module monitor device
"44" QDS Ver.! : Error code stored in driver operation alarm/detail number (hexadecimal display (First 2 digits: Driver operation alarm, last 2 digits: Detail number))
"50" or "51" : Error code stored in SD32</td></tr>
<tr><td rowspan="3">Error occur-rence time</td><td>Year/ month</td><td colspan="2" rowspan="3">The clock data at error occurrence (SD210, SD211, SD212) are set.
(BCD code, year in its lower 2 digits)</td></tr>
<tr><td>Day/ hour</td></tr>
<tr><td>Minute/ second</td></tr>
</table>

Ver.! : Refer to Section 1.3 for the software version that supports this function.

Table 12.2 Motion error history device error information (Continued)

Signal name	Description	
	Motion SFC control errors	Conventional errors
Error setting data information	b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b15 → Error setting data 0 : No data 1 : Data	b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 b10 → Speed control 10 × multiplier setting for degree axis 0 : Invalid 1 : Valid The status at error occurrence is set when the unit of error setting data is set to "11: Control unit (Speed data)" and the control unit is set to "10: degree". b11, b12 → Control unit/Display format Ver.! • The control unit is set when the unit of error setting data is set to "01: Axis unit, Output module unit", "10: Control unit (Address data, Radius-specified allowable error range for circular interpolation)" and "11: Control unit (Speed data)". 00 : mm 01 : inch 10 : degree 11 : pulse (Note): Virtual servo motor axis : "11: Fixed at pulse" • The display format is set when the unit of error setting data is set to "00: None". Ver.! 00 : Signed decimal display 01 : Unsigned decimal display 10 : Hexadecimal (least bit 4 digits display) 11 : Hexadecimal (8 digits display) b13, b14 → Unit of error setting data 00 : None 01 : Axis unit, Output module unit (At output module error occurrence) 10 : Control unit (Address data, Radius-specified allowable error range for circular interpolation) 11 : Control unit (Speed data) b15 → Error setting data 0 : No data 1 : Data
Unusable	—	
Error setting data	• Details code of error is stored. • An error without a details code is fixed at 0.	• Setting data in error cause • Error type is followings; "15" QDS : Parameter error No. stored in #8009+20n (Hexadecimal) "42" QDS Ver.! : Fixed at 0 "50" or "51" : SD33 (Details of safety observation error) is stored.

(Note-1): If the synchronous control dedicated function for the Motion SFC program fails to be executed, an error code is output to the Motion error history device and its details code is also output to the error setting data.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(2) Motion error detection flag (M2039)

The Motion error detection flag (M2039) turns on when any of the errors detected by the Motion CPU occurs.
At error occurrence, data are set to the error devices in the following procedure.

(a) Set the error code to each axis or error devices.

(b) Turns on the error detection signal of each axis or error.

(c) Set the error information to Motion error history devices (#8640 to #8735).

(d) Turns on the Motion error detection flag (M2039) .

In the user program, reset the "Motion error detection flag (M2039)" after reading the error history at the "Motion error detection flag (M2039)".
After that, "Motion error detection flag (M2039)" turns on again at occurrence of a new error.

POINT

(1) Eliminate the error cause after confirming error content, and then turn OFF the Motion error detection flag (M2039) by user side.
The self-diagnostic error information except the stop error is cleared by turning M2039 ON to OFF.

(2) Set the clock data and clock data read request (SM801) in the user program.

(3) Error setting at servo warning occurrence QDS

Set whether or not to output errors to the Motion error history or self-diagnostic error at servo warning occurrence.
Set this setting in the system basic setting of system settings.
Refer to the "Q173D(S)CPU/Q172D(S)CPU Motion controller Programming Manual (COMMON)" for details.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

12.3 Motion SFC Error Code List

(1) Motion SFC program start errors (16000 to 16099)

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
16000	—	PLC ready OFF (SFCS)	At a start by D(P).SFCS instruction, PLC ready flag (M2000) or PCPU READY complete flag (SM500) is OFF.	The specified Motion SFC program does not start.	Provide ON of the PLC ready flag (M2000) and PCPU READY complete flag (SM500) as start interlocks.
16001	—	Motion SFC program No. error (SFCS)	At a start by D(P).SFCS or GSUB, the SFC program No. is outside the range of 0 to 255.	The specified Motion SFC program does not start. When it started by GSUB, the start source Motion SFC also stop to execute.	Check the Motion SFC program No., and correct a sequence program.
16002	—	None Motion SFC program (SFCS)	At the Motion SFC program start made by D(P).SFCS or GSUB, the specified Motion SFC program does not exist.		Check the Motion SFC program No., and correct a sequence program, or create the non-created Motion SFC program.
16003	—	Double start error	At the Motion SFC program start made by D(P).SFCS or GSUB, the same Motion SFC program starts.		Double start should be managed on the user side. Provide the user's starting signal as a start interlock in the sequence program.
16004	—	PLC ready OFF (GINT)	D(P).GINT instruction was executed with PLC ready flag (M2000) or PCPU READY complete flag (SM500) is OFF.	The active step of Motion SFC program executed by "PLC interrupt" is not processed. "Interrupt instruction No." is set in the error Motion SFC program No.	Provide ON of PLC ready flag (M2000) and PCPU READY complete flag (SM500) as D(P).GINT execution interlocks.
16005	—	None Motion SFC program	At a Motion SFC program start by automatic start setting or GSUB, the specified Motion SFC program does not exist.	The specified Motion SFC program does not start. When it started by GSUB, the start source Motion SFC also stop to execute.	Check the Motion SFC program No., and correct a program, or create the non-created Motion SFC program.
16006	—	Double start error	At a Motion SFC program start by automatic start setting or GSUB, the same Motion SFC program is already starting.		Double start should be managed on the user side. Provide the user's starting signal as an interlocks in the transition condition.
16007	—	Online change	The Motion SFC program which is rewriting the Motion SFC chart by online change was started.		Start after the completion of online change.

(2) Motion SFC interpreter detection errors (16100 to 16199)

<table>
<tr><th rowspan="2">Error code</th><th rowspan="2">Details code</th><th colspan="2">Error factor</th><th rowspan="2">Error Processing</th><th rowspan="2">Corrective Action</th></tr>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>16100</td><td>—</td><td rowspan="4">Motion SFC program error (grammatical error)</td><td>• The code exists but is grammatically erroneous.
• Though not within branch-coupling, a label/jump code within selective branch-coupling or a label/jump code within parallel branch-coupling exists.</td><td rowspan="14">Stop to execute the applicable Motion SFC program No. For the subroutine called program, the call source program also stops to execute.</td><td rowspan="10">The Motion SFC program code is corrupted. Turn PLC ready flag (M2000) OFF and write the Motion SFC program again.
Or, replace the external battery if it passed over a life.</td></tr>
<tr><td>16101</td><td>—</td><td>Selective branch destinations are all headed by other than SFT or WAIT transitions.</td></tr>
<tr><td>16102</td><td>—</td><td>WAITON/WAITOFF is not followed by a motion control step. (However, this is permitted to a pointer (Pn) or jump (Pn).)</td></tr>
<tr><td>16103</td><td>—</td><td>A parallel branch is followed by an END step without a parallel coupling.</td></tr>
<tr><td>16104</td><td>—</td><td>Motion SFC code error</td><td>An impossible code is used.
The internal code is corrupted.</td></tr>
<tr><td>16105</td><td>—</td><td>Jump code error 1</td><td>Internal code (list code) error in jump destination information</td></tr>
<tr><td>16106</td><td>—</td><td>Jump code error 2</td><td>Internal code (label information) error in jump destination information</td></tr>
<tr><td>16107</td><td>—</td><td>Jump code error 3</td><td>Internal code (label No.) error in jump destination information</td></tr>
<tr><td>16108</td><td>—</td><td>Jump code error 4</td><td>Internal code (label address) error in jump destination information</td></tr>
<tr><td>16109</td><td>—</td><td>Jump destination error</td><td>The specified pointer does not exist at the jump destination.</td></tr>
<tr><td>16110</td><td>—</td><td>GSUB setting error 1</td><td>The self program was called/started by GSUB.</td><td rowspan="2">GSUB cannot call its own or main program. Correct the Motion SFC program.</td></tr>
<tr><td>16111</td><td>—</td><td>GSUB setting error 2</td><td>The main program was called/started by GSUB.</td></tr>
<tr><td>16112</td><td>—</td><td>Parallel branch nesting excess</td><td>Nesting of parallel branches within a parallel branch route exceeded four levels.</td><td>The nesting of parallel branch is up to four levels.
Subroutine the branch destination processing and correct the program.</td></tr>
<tr><td>16113</td><td>—</td><td>Executed task error</td><td>An attempt was made to execute a motion control step K with an event or NMI task.</td><td>Motion control steps cannot be executed in the Motion SFC programs executed by the event and NMI tasks.</td></tr>
<tr><td>16120</td><td>—</td><td>Simultaneously active step count excess</td><td>The number of simultaneously active steps exceeded 256 during execution.</td><td></td><td>Number of simultaneously active steps is maximum 256. Re-examine the Motion SFC program.</td></tr>
</table>

(3) Motion SFC program run errors (16200 to 16299)

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
16200	—	No specified program (Kn)	The servo program (Kn) specified with the motion control step does not exist.	Stop to execute the applicable Motion SFC program No. For the subroutine called program, the call source program also stops to execute.	Create the specified servo program.
16201	—	No specified program (Fn/FSn)	The operation control program (Fn/FSn) specified with the operation control step does not exist.		Create the specified operation control program.
16202	—	No specified program (Gn)	The program (Gn) specified with the transition does not exist.		Create the specified transition program.
16203	—	No specified program (Motion SFC)	The Motion SFC program specified with the clear step does not exist.		Correct the specified Motion SFC program name or create the specified Motion SFC program.
16204	—	No setting of operation expression/conditional expression	The program (Gn) specified with the transition does not have a conditional expression setting.		Be sure to set a conditional expression in the last block of the transition program.
16205	—	Fn/FSn program code error	Internal code error in the operation control program (Fn/FSn)		The Motion SFC program code is corrupted. Turn PLC ready flag (M2000) OFF and write the Motion SFC program again. Or, replace the external battery if it passed over a life.
16206	—	Gn program code error	Internal code error in the transition program (Gn)		
16207	—	Specified the invalid device	The invalid device (T, C) or shared device outside range in the program is set.		Correct the program which does set the effective device.
16208	—	Limited count for repeat control over	The repeated control instructions (FOR) are repeatedly executed exceeding the limited count for repeat control set in the parameter in one operation control program or transition program.		Review the program so that the repeated control instruction (FOR) does not exceed the limited count for repeat control.
16209	—	Program control instruction block execution error	An operation error (indirectly specified device read error/assignment execution error) occurred when the block is executed to the program control instruction (IF/CASE/FOR/NEXT).		• Correct the program so that the device No. which is indirectly specified is proper. • Correct the program so that the assignment (S) data is within the data (D).

(4) Operation control/transition execution errors (16300 to 16599)

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16301	—	Event task enable (EI) execution error	Event task enable was executed at except for the normal task.	The block processing on executing is stopped and the next block is executed.	Event task enable may be executed in the normal task only. Correct the program.
16302	—	Event task disable (DI) execution error	Event task disable was executed at except for the normal task.		Event task disable may be executed in the normal task only. Correct the program.
16303	—	Block transfer (BMOV) execution error	• The cam data of the cam No. specified with (D) or (S) is not yet registered to the Motion controller. • The resolution of the cam No. specified with (D) or (S) differs from the number of transferred words specified with (n). • (S) to (S)+(n-1) is outside the device range. • (D) to (D)+(n-1) is outside the device range. • (n) is 0 or a negative number. • (S) is a bit device and the device number is not a multiple of 16. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S) to (S)+(n-1). • PX/PY is set in (D) to (D)+(n-1). • When it is advanced synchronous control method, the cam No. is set in the (D) or (S).		• Correct the program so that cam data is that of the already registered cam No. • Correct the program to match (n) with the cam resolution. • Change (n) so that the block transfer range is within the device range. • Change (n) to a positive number. • When (S) or (D) is a bit device, set the device number to be multiple of 16. • When (S) or (D) is a bit device, do not set PX/PY. • When it is advanced synchronous control method, do not set the cam No. in the (D) or (S).
16304	—	Time to wait (TIME) execution error	• The device No. which indirectly specifies (S) is illegal. • The (S) data is outside the range 0 to 2147483647.		• Correct the program so that the device No. which indirectly specifies (S) is proper. • Correct the program so that the (S) data is within the range of 0 to 2147483647.
16305	—	Same data block transfer (FMOV) execution error	• (D) to (D)+(n-1) is outside the device range. • (n) is 0 or a negative number. • (S) is a bit device and the device number is not a multiple of 16. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S). • PX/PY is set in (D) to (D)+(n-1).		• Change (n) so that the block transfer range is within the device range. • When (S) or (D) is a bit device, set the device number to be multiple of 16. • When (S) or (D) is a bit device, do not set PX/PY.
16308	—	Speed change request (CHGV) execution error	The specified axis No. is outside the range.		Correct the program so that the specified axis No. is within the range.
16309	—	Torque limit value change request (CHGT) execution error			
16310	—	Target position change request (CHGP) execution error	• The specified axis No. of (S1) is outside the range. • (S2) is outside the range of 0 to 1. • The device number of (S3) is odd-numbered. • (S3) to ((S3)+7) is outside the device range.		• Correct the specified axis No. of (S1) so that it is within the range. • Correct the program so that the (S2) is within the range of 0 to 1. • Correct the program so that the (S3) is even-numbered device. • Correct the program so that the devices from (S3) to ((S3)+7) are within the range.
16311	—	Torque limit value individual change request (CHGT2) execution error	The specified axis No. is outside the range.		Correct the program so that the specified axis No. is within the range.
16316	—	Assignment (=) execution error	• The (S) data is outside the range of the data type of (D). • The device No. which indirectly specifies (D) is illegal.		• Correct the program so that the (S) data is within the range of the data type of (D). • Correct the program so that the device No. which indirectly specifies (D) is proper.

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16320	—	Operation (/) execution error	The divisor is 0.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the divisor is other than 0.
16321	—	Remainder (%) execution error			
16322	—	Device set (SET) execution error	• The device No. which indirectly specifies (D) is illegal. • (D) is a device which is write-disabled.		• Correct the program so that the device No. which indirectly specifies (D) is proper. • Correct the program to set a write-enabled device at (D).
16333	—	Device reset (RST) execution error			
16334	—	Device set (SET=) execution error			
16335	—	Device reset (RST=) execution error			
16336	—	Device output (DOUT) execution error			
16337	—	Device input (DIN) execution error	The device No. which indirectly specifies (D) is illegal.		Correct the program so that the device No. which indirectly specifies (D) is proper.
16338	—	Bit device output (OUT=) execution error			
16368	—	Direct specification 16 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error	Multiple CPU area device number is outside the range that set by the parameter.		Correct the program so that Multiple CPU area device number is within the range set in the parameter.
16369	—	Direct specification 32 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error			
16370	—	Direct specification 64 bit Multiple CPU area device for CPU No.1 (U3E0\G10000 to) read error			
16371	—	Direct specification 16 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16372	—	Direct specification 32 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16373	—	Direct specification 64 bit Multiple CPU area device for CPU No.2 (U3E1\G10000 to) read error			
16374	—	Direct specification 16 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error			
16375	—	Direct specification 32 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error			

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16376	—	Direct specification 64 bit Multiple CPU area device for CPU No.3 (U3E2\G10000 to) read error	Multiple CPU area device number is outside the range that set by the parameter.	The block processing on executing is stopped and the next block is executed.	Correct the program so that Multiple CPU area device number is within the range set in the parameter.
16377	—	Direct specification 16 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			
16378	—	Direct specification 32 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			
16379	—	Direct specification 64 bit Multiple CPU area device for CPU No.4 (U3E3\G10000 to) read error			
16380	—	Signed 16-bit integer value conversion (SHORT) execution error	The (S) data is outside the signed 16-bit integer value range.		Correct the program so that the (S) data is within the signed 16-bit integer value range.
16381	—	Unsigned 16-bit integer value conversion (USHORT) execution error	The (S) data is outside the unsigned 16-bit integer value range.		Correct the program so that the (S) data is within the unsigned 16-bit integer value range.
16382	—	Signed 32-bit integer value conversion (LONG) execution error	The (S) data is outside the signed 32-bit integer value range.		Correct the program so that the (S) data is within the signed 32-bit integer value range.
16383	—	Unsigned 32-bit integer value conversion (ULONG) execution error	The (S) data is outside the unsigned 32-bit integer value range.		Correct the program so that the (S) data is within the signed 32-bit integer value range.
16386	—	32-bit→64-bit floating-point type data conversion (DFLT) execution error	The (S) data is not a valid 32-bit floating-point type.		Correct the program so that the (S) data is valid as a 32-bit floating-point type.
16387	—	64-bit→32-bit floating-point type data conversion (SFLT) execution error	The (S) data is not in a valid 64-bit floating-point type. Or the converted value exceeded the 32-bit floating-point type range.		Correct the program so that the (S) data is valid as a 64-bit floating-point type and the converted value is within the range of 32-bit floating-point type.
16398	—	Tangent (TAN) execution error	(S) is 90+(180*n). (n is an integer)		Correct the program so that (S) is not 90+(180*n). (n is an integer)
16399	—	Arcsine (ASIN) execution error	(S) is outside the range of -1.0 to 1.0.		Correct the program so that (S) is within the range of -1.0 to 1.0.
16400	—	Arccosine (ACOS) execution error			
16402	—	Square root (SQRT) execution error	(S) is a negative number.		Correct the program so that (S) is a positive number.
16403	—	BCD→BIN conversion (BIN) execution error	Any digit of (S) has a value other than 0 to 9.		Correct the program so that each digit of (S) is 0 to 9.
16404	—	BIN→BCD conversion (BCD) execution error	The (S) value is outside the range where BIN data can be converted into BCD data.		Correct the program so that the (S) value is within the range.
16405	—	Natural logarithm (LN) execution error	(S) is 0 or a negative number.		Correct the program so that (S) is a positive number.
16407	—	Absolute value (ABS) execution error	The value of (S) is outside the range of the absolute value conversion.		Correct the program so that the (S) value is within the range.

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
16412	—	16-bit integer type scaling (SCL) execution error	• (S1) is outside the range of 0 to 3. • The device number of (S3) is odd-numbered. • (S3) to (S3) + (2N + 1) is outside the device range. • The number of points is outside the range of 2 to 4000. • In the sequential search ((S1) is 0 or 1), the points corresponding to the input values (for positive conversion: X_0 to X_{N-1}/for inverse conversion: Y_0 to Y_{N-1}) are not in ascending order. • The conversion results are outside the range of the data type (D).	The block processing on executing is stopped and the next block is executed.	• Correct the program so that the (S1) is within the range of 0 to 3. • Correct the program so that the (S3) is even-numbered device. • Correct the program so that the devices from (S3) to (S3) + (2N + 1) are within the range. • Correct the program so that the number of points is within the range of 2 to 4000. • Correct the program so that the points corresponding to the input values (for positive conversion: X_0 to X_{N-1}/for inverse conversion: Y_0 to Y_{N-1}) increases monotonically. • Correct the program so that the conversion results are within the data type (D).
16413	—	32-bit integer type scaling (DSCL) execution error	• (S1) is outside the range of 0 to 3. • The device numbers of (S2), (S3), and (D) are odd-numbered. • (S3) to (S3) + (4N + 1) is outside the device range. • The number of points is outside the range of 2 to 2000. • In the sequential search ((S1) is 0 or 1), the points corresponding to the input values (for positive conversion: X_0 to X_{N-1}/for inverse conversion: Y_0 to Y_{N-1}) are not in ascending order. • The conversion results are outside the range of the data type (D).		• Correct the program so that the (S1) is within the range of 0 to 3. • Correct the program so that the (S2), (S3), and (D) are even-numbered devices. • Correct the program so that the devices from (S3) to (S3) + (4N + 1) are within the range. • Correct the program so that the number of points is within the range of 2 to 2000. • Correct the program so that the points corresponding to the input values (for positive conversion: X_0 to X_{N-1}/for inverse conversion: Y_0 to Y_{N-1}) increases monotonically. • Correct the program so that the conversion results are within the data type (D).
16414	1	Cam data read (CAMRD) execution error	Cam No. specified by (S1) is outside the range of 1 to 256.		Correct the program so that the cam No. is within the range of 1 to 256.
	2		Cam data which No. is specified by (S1) does not exist in cam open area.		Correct the program to specify a cam No. in which cam data exists.
	3		• In the stroke ratio data format cam, the cam data first position specified by (S2) is outside the range of 1 to cam resolution. • In the coordinate data format cam, the cam data first position specified by (S2) is outside the range of 0 to (coordinate number - 1).		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to cam resolution. • In the coordinate data format cam, correct the program so that it is within the range of 0 to (coordinate number - 1).
	4		• In the stroke ratio data format cam, cam data operation points is outside the range of 1 to 4096. • In the coordinate data format cam, cam data operation points is outside the range of 1 to 2048.		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to 4096. • In the coordinate data format cam, correct the program so that it is within the range of 1 to 2048.
	5		The end device number which stores the cam data is outside the range.		Correct the operation points so that the end device number which stores the cam data is within the range.
	6		The first number of the storage device for (D) cam data is not even-numbered.		Correct the program so that the device number is an even-numbered.
	7		Read the cam data with "Read/write protection" password set.		Execute the cam data read operation after the password is disabled.

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16415	1	Cam data write (CAMWR) execution error	Cam No. specified by (S1) is outside the range of 1 to 256.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the cam No. is within the range of 1 to 256.
	2		• In the stroke ratio data format cam, the cam data first position specified by (S2) is outside the range of 1 to cam resolution. • In the coordinate data format cam, the cam data first position specified by (S2) is outside the range of 0 to (coordinate number - 1).		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to cam resolution. • In the coordinate data format cam, correct the program so that it is within the range of 0 to (coordinate number - 1).
	3		• In the stroke ratio data format cam, cam data operation points is outside the range of 1 to 4096. • In the coordinate data format cam, cam data operation points is outside the range of 1 to 2048. • In the cam data write operation, set a first position and operation points that was outside the setting range for cam resolution or coordinate number.		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to 4096. • In the coordinate data format cam, correct the program so that it is within the range of 1 to 2048. • Correct the program to guarantee that "cam data first position + (cam data operation points - 1)" is not outside the range for cam resolution or coordinate number.
	4		The end device number which stores the cam data operation points specified by (n) is outside the range.		Correct the program so that the end device number which stores the cam data is within the range.
	5		The first number of the storage device for (S3) cam data is not even-numbered.		Correct the program so that the device number is an even-numbered.
	6		Cam data format specified by (S3) has been set other than 1 or 2.		Correct the program to set the value as 1 or 2 in the cam data format.
	7		• In the stroke ratio data format cam, the cam resolution has been set a value outside the range of "256/512/1024/2048/4096/8192/16384/32768". • In the coordinate data format cam, the coordinate number has been set a value outside the range of "2 to 16384".		• In the stroke ratio data format cam, correct the program so that the value is within the range of "256/512/1024/2048/4096/8192/16384/32768". • In the coordinate data format cam, correct the program so that the value is within the range of "2 to 16384".
	8		The cam data starting point is outside the range of 0 to (cam resolution - 1) in stroke ratio data format cam.		Correct the program so that the value is within the range of 0 to (cam resolution - 1).
	9		• There is not enough free area in cam storage area when write to cam data. • It is impossible to write according to the free area.		• Decrease the cam data number (cam number, cam resolution, coordinate number). • Delete the cam data and write again.
	10		• There is not enough free area in cam open area when write to cam data. • It is impossible to write according to the free area.		Decrease the cam data number (cam number, cam resolution, coordinate number).
	11		• Coordinate data input value is negative value. • Coordinate data input value is not "$X_n < X_{n+1}$".		• Correct the setting value to set the input value of coordinate data above 0. • Correct the setting value to set the input value of coordinate data as "$X_n < X_{n+1}$".
	12		Read the cam data with "Write protection" or "Read/write protection" password set.		Execute the cam data read operation after the password is disabled.
	13		Execute CAMWR instruction when writing cam data (CAMWR instruction, CAMWR2 instruction or CAMMK instruction) by Motion SFC program.		Correct the program to set the CAMWR instruction not execute when the cam data write flag (SM505) is ON.

Error code	Details code	Error factor Name	Error factor Description	Error Processing	Corrective Action
16416	1	Cam data write (Cam open area) (CAMWR2) execution error	Cam No. specified by (S1) is outside the range of 1 to 256.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the cam No. is within the range of 1 to 256.
	2		• In the stroke ratio data format cam, the cam data first position specified by (S2) is outside the range of 1 to cam resolution. • In the coordinate data format cam, the cam data first position specified by (S2) is outside the range of 0 to (coordinate number - 1).		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to cam resolution. • In the coordinate data format cam, correct the program so that it is within the range of 0 to (coordinate number - 1).
	3		• In the stroke ratio data format cam, cam data operation points is outside the range of 1 to 4096. • In the coordinate data format cam, cam data operation points is outside the range of 1 to 2048. • In the cam data write operation, set a first position and operation points that was outside the setting range for cam resolution or coordinate number.		• In the stroke ratio data format cam, correct the program so that it is within the range of 1 to 4096. • In the coordinate data format cam, correct the program so that it is within the range of 1 to 2048. • Correct the program to guarantee that "cam data first position + (cam data operation points - 1)" is not outside the range for cam resolution or coordinate number.
	4		The end device number which stores the cam data operation points specified by (n) is outside the range.		Correct the program so that the end device number which stores the cam data is within the range.
	5		The first number of the storage device for (S3) cam data is not even-numbered.		Correct the program so that the device number is an even-numbered.
	6		Cam data format specified by (S3) has been set other than 1 or 2.		Correct the program to set the value as 1 or 2 in the cam data format.
	7		• In the stroke ratio data format cam, the cam resolution has been set a value outside the range of "256/512/1024/2048/4096/8192/16384/32768". • In the coordinate data format cam, the coordinate number has been set a value outside the range of "2 to 16384".		• In the stroke ratio data format cam, correct the program so that the value is within the range of "256/512/1024/2048/4096/8192/16384/32768". • In the coordinate data format cam, correct the program so that the value is within the range of "2 to 16384".
	8		The cam data starting point is outside the range of 0 to (cam resolution - 1) in stroke ratio data format cam.		Correct the program so that the value is within the range of 0 to (cam resolution - 1).
	9		• There is not enough free area in cam storage area when write to cam data. • It is impossible to write according to the free area.		• Decrease the cam data number (cam number, cam resolution, coordinate number). • Delete the cam data and write again.
	10		• There is not enough free area in cam open area when write to cam data. • It is impossible to write according to the free area.		Decrease the cam data number (cam number, cam resolution, coordinate number).
	11		• Coordinate data input value is negative value. • Coordinate data input value is not "X_n < X_{n+1}".		• Correct the setting value to set the input value of coordinate data above 0. • Correct the setting value to set the input value of coordinate data as "X_n < X_{n+1}".
	12		Read the cam data with "Write protection" or "Read/write protection" password set.		Execute the cam data read operation after the password is disabled.
	13		Execute CAMWR2 instruction when writing cam data (CAMWR instruction, CAMWR2 instruction or CAMMK instruction) by Motion SFC program.		Correct the program to set the CAMWR2 instruction not execute when the cam data write flag (SM505) is ON.

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
16417	1	Cam auto-generation (CAMMK) execution error	Cam No. specified by (S1) is outside the range of 1 to 256.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the cam No. is within the range of 1 to 256.
	2		Auto-generation type specified by (S2) has been set to a value that does not correspond to an auto-generation type.		Correct the program to specify a value that corresponds to an auto-generation type.
	3		• There is not enough free area in cam storage area. • It is impossible to write according to the free area.		• Decrease the cam data number (cam number, cam resolution, coordinate number). • Delete the cam data and write again.
	4		• There is not enough free area in cam open area when write to cam data. • It is impossible to write according to the free area.		Decrease the cam data number (cam number, cam resolution, coordinate number).
	5		The end device number for auto-generation data which the first number of storage auto-generation data is specified by (S3) is outside the range.		Correct the program so that the end device number for auto-generation data is within the setting range.
	6		The first number of the storage device for (S3) auto-generation data is not even-numbered.		Correct the program so that the device number is an even-numbered.
	7		Some data outside the range has been set in auto-generation data.		Correct the setting value of auto-generation data within the setting range.
	8		For the cam rotary cutter, a value has been set as sheet synchronization width ≥ sheet length in the auto-generation parameter.		Review the value of auto-generation parameter so sheet synchronization width < sheet length.
	9		For the cam rotary cutter, asynchronous speed will be reduced when the auto-generate data is set as synchronous axis length (synchronous axis diameter × π) < sheet length.		Review the setting value for auto-generation data so that the asynchronous speed cannot be reduced.
	10		For the cam rotary cutter, asynchronous speed is 655.35 times or larger than synchronous speed by auto-generation data.		Review the setting value of auto-generation data so that the asynchronous speed is not higher than 655.35 times the synchronous speed.
	11		Execute cam auto-generation when "Write protection" or "Read/write protection" of cam data is set with password.		Execute the auto-generation operation after the password is disabled.
	12		Execute CAMMK instruction when writing cam data (CAMWR instruction, CAMWR2 instruction or CAMMK instruction) by Motion SFC program.		Correct the program to set the CAMMK instruction not execute when the cam data write flag (SM505) is ON.
	20		For the easy stroke ratio cam, the end points for each section are not in ascending order.		Set a value for each end point that is larger than the previous end point.
	21		For the easy stroke ratio cam, the end point of the final section is less than the cam axis length per cycle.		Set the end point of the final section as the cam axis length per cycle.

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16418	1	Cam position calculation (CAMPSCL) execution error	Cam No. specified by (S1) is outside the range of 1 to 256.	The block processing on executing is stopped and the next block is executed.	Correct the program so that the cam No. is within the range of 1 to 256.
	2		Cam data which No. is specified by (S1) does not exist in cam open area.		Correct the program to specify a cam No. in which cam data exists.
	3		The end device number for cam position calculation control data which the first number of storage cam position calculation control data is specified by (S2) is outside the range.		Correct the program so that the end device number for cam position calculation control data is within the setting range.
	4		The first number of the storage device for cam position calculation control data specified by (S2) is not even-numbered.		Correct the program so that the device number is an even-numbered.
	5		Cam position calculation type specified by cam position calculation data has been set other than 0 or 1.		Correct the program to set the cam position calculation type to 0 or 1.
	6		Cam axis length per cycle is outside the range of 1 to 2147483647.		Set the value within the range of 1 to 2147483647.
	7		Cam axis current value per cycle is outside the range of 0 to (Cam axis length per cycle).		Set the value within the range of 0 to cam axis length per cycle.
	8		The device No. which stores the result of cam position calculation specified by (D) is outside the range.		Correct the program so that the device number is within the range.
	9		Odd-numbered device has been set in the device No. which stores the result of cam position calculation specified by (D).		Correct the program so that the device number is an even-numbered.
	10		Cam axis current value per cycle cannot be calculated for those axes.		To control into the reciprocated cam pattern stroke, set the cam stroke amount of cam position calculation control data, cam reference position and cam axis current feed value.
16420	—	Write device data to CPU shared memory of the self CPU (MULTW) execution error	• Number of words (n) to be written is outside the range of 1 to 256. • The CPU shared memory address (D) of self CPU of the writing destination device is outside the range (800H to FFFH) of the CPU shared memory address. • The CPU shared memory address (D) of self CPU of the writing destination device + number of words (n) to be written is outside the range (800H to FFFH) of the CPU shared memory address. • Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range. • MULTW instruction was executed again before MULTW instruction is executed and complete bit device is turned on. • (D1) is a write-disabled device. • (S) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S) to (S)+(n-1).		• Correct the program so that the number of words (n) to be written is within the range of 1 to 256. • Correct the program so that the CPU shared memory address (D) of self CPU of the writing destination is within the range of CPU shared memory address. • Correct the program so that the CPU shared memory address (D) of self CPU of the writing destination + number of words (n) to be written is within the range of CPU shared memory address. • Correct the program so that start device No. (S) which writing data are stored + number of words (n) to be written is within the device range. • Execute MULTW instruction again after the complete bit device of MULTW instruction is turned on. • Correct the program to set a write-enabled device at (D1). • When (S) is a bit device, set the device number to be multiple of 16. • When (S) is a bit device, do not set PX/PY.

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16421	—	Read device data from CPU shared memory (MULTR) execution error	• Number of words (n) to be read is outside the range of 1 to 256. • The CPU shared memory first address (S2) of the data which it will be read is outside the range (000H to FFFH) of the CPU shared memory address. • The CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is outside the range (000H to FFFH) of the CPU shared memory address. • Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range. • Except 3E0H/3E1H/3E2H/3E3H is set at (S1). • The CPU which reads is resetting. • The errors are detected in the CPU which read. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (D) to (D)+(n-1).	The block processing in execution is stopped and the next block is executed.	• Correct the program so that the number of words (n) to be read is within the range of 1 to 256. • Correct the program so that the CPU shared memory first address (S2) of the data which it will be read is within the device range of CPU shared memory address. • Correct the program so that the CPU shared memory first address (S2) of the data which it will be read + number of words (n) to be read is within the range of CPU shared memory address. • Correct the program so that start device No. (D) which stores the reading data + number of words (n) to be read is within the device range. • Correct the program so that 3E0H/3E1H/3E2H/3E3H is set at (S1). • Check that the resetting flag (SM240 to SM243) is OFF, then correct the program to execute the MULTR instruction. • If the errors are detected in the CPU which read, exchange the CPU. • When (D) is a bit device, set the device number to be multiple of 16. • When (D) is a bit device, do not set PX/PY.
16422	—	Write device data to intelligent function module (TO) execution error	• Number of words (n) to be written is outside the range of 1 to 256. • Motion CPU cannot communicate with intelligent function module at the instruction execution. • Abnormalities of the intelligent function module were detected at the instruction execution. • I/O No.s specified with (D1) differ from the intelligent function module controlled by the self CPU. • The address specified with (D2) is outside the buffer memory range. • Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range. • (S) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (S) to (S)+(n-1).		• Correct the program so that the number of words (n) to be written is within the range of 1 to 256. • Replace the intelligent function module if there is a fault. • Correct the program so that the first I/O No.s specified with (D1) is intelligent function module controlled by the self CPU. • Correct the program so that the address specified with (D2) is within the buffer memory range. • Correct the program so that start device No. (S) which writing data are stored + number of words (n) to be written is within the device range. • When (S) is a bit device, set the device number to be multiple of 16. • When (S) is a bit device, do not set PX/PY.
16423	—	Read device data from intelligent function module (FROM) execution error	• Number of words (n) to be read is outside the range of 1 to 256. • Motion CPU cannot communicate with intelligent function module at the instruction execution. • Abnormalities of the intelligent function module were detected at the instruction execution. • I/O No.s specified with (S1) differ from the intelligent function module controlled by the self CPU. • The address specified with (S2) is outside the range buffer memory. • Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range. • (D) is a bit device and the device number is not a multiple of 16. • PX/PY is set in (D) to (D)+(n-1).		• Correct the program so that the number of words (n) to be read is within the range of 1 to 256. • Replace the intelligent function module if there is a fault. • Correct the program so that the first I/O No.s specified with (S1) is intelligent function module controlled by the self CPU. • Correct the program so that the address specified with (S2) is within the buffer memory range. • Correct the program so that start device No. (D) which stores the reading data + number of words(n) to be read is within the device range. • When (D) is a bit device, set the device number to be multiple of 16. • When (D) is a bit device, do not set PX/PY.

<table>
<tr><th rowspan="2">Error code</th><th rowspan="2">Details code</th><th colspan="2">Error factor</th><th rowspan="2">Error Processing</th><th rowspan="2">Corrective Action</th></tr>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>16424</td><td>—</td><td>Write buffer memory data to head module (RTO) execution error</td><td>• Number of words (n) to be written is outside the range 1 to 240.
• The SSCNETⅢ/H head module axis No. specified with (D1) is outside the range 1 to 8.
• The SSCNETⅢ/H head module is not connected at the instruction execution.
• Start device No. (S) which writing data are stored + number of words (n) to be written is outside the device range.
• (S) is a bit device and device number is not a multiple of 16.
• PX/PY is set in (S) to (S)+(n-1).
• RTO instruction was executed again before RTO instruction is executed and complete bit is turned on.</td><td rowspan="10">The block processing in execution is stopped and the next block is executed.</td><td>• Correct the program so that number of words (n) to be written is within the range 1 to 240.
• Correct the program so that the SSCNETⅢ/H head module axis No. specified with (D1) is within the range 1 to 8.
• Connect the SSCNETⅢ/H head module.
• Correct the program so that start device No. (S) which writing data are stored + number of words (n) to be written is within the device range.
• When (S) is a bit device, set the device number to be multiple of 16.
• When (S) is a bit device, do not set PX/PY.
• Execute RTO instruction again after the complete bit of RTO instruction is turned on.</td></tr>
<tr><td>16425</td><td>—</td><td>Read buffer memory data from head module (RFROM) execution error</td><td>• Number of words (n) to be read is outside the range 1 to 240.
• The SSCNETⅢ/H head module axis No. specified with (S1) is outside the range 1 to 8.
• The SSCNETⅢ/H head module is not connected at the instruction execution.
• Start device No. (D) which stores the reading data + number of words (n) to be read is outside the device range.
• (D) is a bit device and device number is not a multiple of 16.
• PX/PY is set in (D) to (D)+(n-1).
• RFROM instruction was executed again before RFROM instruction is executed and complete bit is turned on.</td><td>• Correct the program so that number of words (n) to be read is within the range 1 to 240.
• Correct the program so that the SSCNETⅢ/H head module axis No. specified with (S1) is within the range 1 to 8.
• Connect the SSCNETⅢ/H head module.
• Correct the program so that start device No. (D) which stores the reading data + number of words (n) to be read is within the device range.
•When (D) is a bit device, set the device number to be multiple of 16.
• When (D) is a bit device, do not set PX/PY.
• Execute RFROM instruction again after the complete bit of RFROM instruction is turned on.</td></tr>
<tr><td>16441</td><td>—</td><td>Indirect specified 16-bit SD(SD(n)) read error</td><td>The indirectly specified device No. is outside the range.</td><td rowspan="8">Correct the program so that the indirectly specified device No. is proper.</td></tr>
<tr><td>16442</td><td>—</td><td>Indirect specified 32-bit SD(SD(n)L) read error</td><td rowspan="2">The indirectly specified device No. is outside the range or an odd number.</td></tr>
<tr><td>16443</td><td>—</td><td>Indirect specified 64-bit SD(SD(n)F) read error</td></tr>
<tr><td>16462</td><td>—</td><td>Indirect specified 16-bit motion register (#(n)) read error</td><td>The indirectly specified device No. is outside the range.</td></tr>
<tr><td>16463</td><td>—</td><td>Indirect specified 32-bit motion register (#(n)L) read error</td><td rowspan="2">The indirectly specified device No. is outside the range or an odd number.</td></tr>
<tr><td>16464</td><td>—</td><td>Indirect specified 64-bit motion register (#(n)F) read error</td></tr>
<tr><td>16465</td><td>—</td><td>Indirect specified 16-bit data register (D(n)) read error</td><td>The indirectly specified device No. is outside the range.</td></tr>
<tr><td>16466</td><td>—</td><td>Indirect specified 32-bit data register (D(n)L) read error</td><td>The indirectly specified device No. is outside the range or an odd number.</td></tr>
</table>

Error code	Details code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
16467	—	Indirect specified 64-bit data register (D(n)F) read error	The indirectly specified device No. is outside the range or an odd number.	The block processing in execution is stopped and the next block is executed.	Correct the program so that the indirectly specified device No. is proper.
16468	—	Indirect specified 16-bit link register (W(n)) read error	The indirectly specified device No. is outside the range.		
16469	—	Indirect specified 32-bit link register (W(n)L) read error	The indirectly specified device No. is outside the range or an odd number.		
16470	—	Indirect specified 64-bit link register (W(n):F) read error			
16475	—	Indirect specified SM(SM(n)) read error	The indirectly specified device No. is outside the range.		
16482	—	Direct specified Multiple CPU area device bit specified for CPU No.1 (U3E0\G10000.0 to) read error	Multiple CPU area device number is outside the range set in the parameter.		Correct the program so that Multiple CPU area device number is within the range set in the parameter.
16483	—	Direct specified Multiple CPU area device bit specified for CPU No.2 (U3E1\G10000.0 to) read error	Multiple CPU area device number is outside the range set in the parameter.		
16484	—	Direct specified Multiple CPU area device bit specified for CPU No.3 (U3E2\G10000.0 to) read error			
16485	—	Direct specified Multiple CPU area device bit specified for CPU No.4 (U3E3\G10000.0 to) read error			
16486	—	Indirect specified input relay (X(n)) read error	The indirectly specified device No. is outside the range.		Correct the program so that the indirectly specified device No. is proper.
16487	—	Indirect specified output relay (Y(n)) read error			
16488	—	Indirect specified internal relay (M(n)) read error			
16489	—	Indirect specified link relay (B(n)) read error			
16490	—	Annunciator (F(n)) read error			
16516	—	Indirect specified 16-bit batch input relay (X(n)) read error	The indirectly specified device No. is outside the range or is not a multiple of 16.		
16517	—	Indirect specified 32-bit batch input relay (X(n)) read error			
16518	—	Indirect specified 16-bit batch output relay (Y(n)) read error			
16519	—	Indirect specified 32-bit batch output relay (Y(n)) read error			

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
16520	—	Indirect specified 16-bit batch internal relay (M(n)) read error	The indirectly specified device No. is outside the range or is not a multiple of 16.	The block processing in execution is stopped and the next block is executed.	Correct the program so that the indirectly specified device No. is proper.
16521	—	Indirect specified 32-bit batch internal relay (M(n)) read error			
16522	—	Indirect specified 16-bit batch internal/latch relay (B(n)) read error			
16523	—	Indirect specified 32-bit batch internal/latch relay (B(n)) read error			
16524	—	Indirect specified 16-bit batch annunciator (F(n)) read error			
16525	—	Indirect specified 32-bit batch annunciator (F(n)) read error			
16538	—	Indirect specified 16-bit batch SM(SM(n)) read error			
16539	—	Indirect specified 32-bit batch SM(SM(n)) read error			

12.4 Motion SFC Parameter Errors

Motion SFC parameters are checked using MT Developer2.

(1) Leading edge of PLC ready flag (M2000) errors (17000 to 17009)

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
17000	—	Normal task consecutive transition count error	The normal task's consecutive transition count of the Motion SFC program started by the normal task is outside the range 1 to 30.	The initial value of 3 is used for control.	Turn PLC ready flag (M2000) OFF, make correction to set the value within the range, and write it to the CPU.
17001	—	Event task consecutive transition count error	The set number of consecutive transitions of the Motion SFC program started by the event task is outside the range 1 to 10.	The initial value of 1 is used for control.	
17002	—	NMI task consecutive transition count error	The set number of consecutive transitions of the Motion SFC program started by the NMI task is outside the range 1 to 10.		
17003	—	Motion SFC parameter unregistered error	Motion SFC parameter is not written or parameter is corrupted.	The initial value of Motion SFC parameter is used for control.	Turn PLC ready flag (M2000) OFF and write the Motion SFC parameter.
17004	—	Event task operation cycle setting error	The fixed cycle task 0.2ms is set when the operation cycle setting is 0.4ms or more.	The specified Motion SFC program does not start.	Turn PLC ready flag (M2000) OFF, set the operation cycle setting to 0.2ms or correct the timing of the fixed cycle task to 0.4ms or more, and write the value to the CPU.

(2) SFC Program start errors (17010 to 17019)

Error code	Details code	Error factor		Error Processing	Corrective Action
		Name	Description		
17010	—	Executed task setting is illegal	Executed task setting is illegal	The initial value (normal task) is used for control.	Turn PLC ready flag (M2000) OFF, make correction, and write a correct value to the CPU.

12.5 Vision System Errors

Error code	Error factor: Name	Error factor: Description	Error Processing	Corrective Action
18000	Argument range error	• Any of the specified argument in the instruction is outside the range. • The indirectly specified device No. of the specified argument in the instruction is outside the range or the device No. of 32-bit type or 64-bit type is odd-numbered.	The block processing in execution is stopped and the next block is executed.	• Check the argument value and correct the program so that the value is within the range. • Correct the program so that the indirectly specified device No. is proper.
18001	Double open error	MVOPEN instruction was executed to the logged on vision system.		Review the conditions to execute the MVOPEN instruction and correct the program.
18002	Unopen error	The instruction is executed to the vision system which is not logged on.		Correct the program to execute the instruction after confirming the logon by the MVOPEN instruction has been completed normally.
18003	Open error	The communication line with the vision system cannot be opened.	Keep executing Motion SFC.	• Check the connection of the Ethernet cable. • Correct the IP address and the port No. of the Ethernet communication line setting to the same setting as the connected vision system.
18007	Log on error	The user name or password for logging on the vision system is in error.		Correct the user name and the password of the Ethernet communication line setting to the same setting as the connected vision system.
18008	Communication error	• The communication with the vision system is disconnected. • The communication is shut off by the MVCLOSE instruction during command execution. • The TCP/IP port cannot be opened.		• Check the connection of the Ethernet cable. • Correct the program to execute the MVCLOSE instruction after the instruction is completed. • Correct the TCP/IP port No. of the Ethernet communication line setting to the same No. as the TCP/IP protocol setting of the connected vision system. For not using the TCP/IP protocol, delete the TCP/IP port No.
18010	No vision program	The specified vision program (job) does not exist in the vision system.		For the vision program name of the vision program operation setting, specify the job name existing in the vision system.
18011	Read value error	Data in the read value cell is not an integer value.		Check that the cell/tag contents specified in the read value cell of the vision program operation setting are integer. To read the floating-point type data, correct the program to use the TCP/IP protocol or the MVIN instruction.
18012	Execute timeout	Execution of the vision system dedicated function does not finish within the specified time.		Review the time-out period specified by the vision system dedicated function and correct the program.
18013	Trigger response timeout	There is no response for the image request within the specified time.		• Review the time-out period specified by the vision system dedicated function and correct the program. • Check if there is an execution error of job in the vision system side by In-Sight® Explorer and correct the job.
18014	Offline error	The vision system is in the "Offline" status.		Set the vision system in "Online" status by In-Sight® Explorer.
18015	Control authority error	User authorities to control the vision system are not enough.		For the user name which can be specified by the Ethernet communication line setting, specify the user whose access level is "Full access" or "Protect" in the user list setting of In-Sight® Explorer. Also, for "Protect", "Online/Offline switching available" needs to be valid.

Error code	Error factor		Error Processing	Corrective Action
	Name	Description		
18016	Vision system No. error	There is no Ethernet communication line setting corresponding to the specified vision system No.	The block processing in execution is stopped and the next block is executed.	Correct the program to use the vision system No. which has the Ethernet communication line setting.
18018	Double start error	The vision system dedicated function is already being executed for the same vision system.		Correct the program to execute the following vision system dedicated function after confirming the status storage device of the vision system not in "In execution".
18019	Vision program load incomplete error	Trigger is issued before the vision program is loaded.		Correct the program to issue a trigger after confirming the status storage device of the vision program has become "1".
18020	Native mode command send error	Send command character string specified in (S2) of the MVCOM instruction are outside the range of 1 to 191 bytes.		Correct the program to make the length of the send command character string specified in (S2) from 1 to 191 bytes.
18021	Native mode command reception error	The data length received by the MVCOM exceeds 256 bytes. Or the storage device space specified in (D) is insufficient.	Keep executing Motion SFC.	• Do not use the native mode command where the data length of the result exceeds 256 bytes. • When the data length of the result is within 256 bytes, correct the device No. of (D).
18022	Cell/tag name error	The character string length of cell/tag name specified in (S2) of the MVIN/MVOUT instruction is outside the range of 1 to 32 bytes.	The block processing in execution is stopped and the next block is executed.	• Correct the program to make the length of the cell/tag name specified in (S2) from 1 to 32 bytes. • Check that the cell/tag specified in (S2) is defined in the vision system side. • Correct the program so that the data type and data range of (S3) match the vision system side at MVOUT instruction.
		The cell/tag specified in (S2) of the MVIN/MVOUT instruction does not exist in the vision system. Or the data type of the cell/tag specified in (S2) of the MVOUT instruction and the transfer data type/range specified in (S3) do not match.	Keep executing Motion SFC.	
18023	Received data conversion error	• Data received by the MVIN instruction cannot be recognized as numeric data. • At the BIN conversion mode of the MVCOM instruction, received data cannot be recognized as numeric data.		To acquire the data other than numerical value (string, etc.), specify "0: ASCII" to (S3) of the MVCOM instruction or correct the program to use the MC protocol.
18024	Transferred data error	The transferred data specified by the MVOUT instruction (S3) is incorrect.	The block processing in execution is stopped and the next block is executed.	When the floating point type is specified in the data of (S3), correct the program so that the data is within the range of the 32-bit floating point type.

APPENDICES

APPENDIX 1 Processing Times

The processing time for the individual instructions are shown below.
Operation processing times can vary substantially depending on the nature of the source and destinations of the instructions, and the values contained in the following tables should therefore be taken as a set of general guidelines to processing time rather than as being strictly accurate.

APPENDIX 1.1 Processing time of operation control/Transition instruction

(1) Operation instructions

Processing time of operation instructions

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Binary operation	=	Substitution	#0=#1	1.0	1.5
			D800=D801		
			U3E1\G10000=U3E1\G10001	2.0	2.0
			#0L=#2L	1.5	1.5
			D800L=D802L		
			U3E1\G10000L=U3E1\G10002L	2.0	2.0
			#0F=#4F	1.5	2.0
			D800F=D804F		
			U3E1\G10000F=U3E1\G10004F	2.5	3.0
	+	Addition	#0=#1+#2	1.5	2.0
			D800=D801+D802		
			U3E1\G10000=U3E1\G10001+U3E1\G10002	3.0	3.0
			#0L=#2L+#4L	2.0	2.5
			D800L=D802L+D804L		
			U3E1\G10000L=U3E1\G10002L+U3E1\G10004L	2.5	3.5
			#0F=#4F+#8F	2.0	3.0
			D800F=D804F+D808F		
			U3E1\G10000F=U3E1\G10004F+U3E1\G10008F	4.0	4.5
	-	Subtraction	#0=#1-#2	2.0	2.5
			D800=D801-D802		
			U3E1\G10000=U3E1\G10001-U3E1\G10002	3.0	3.5
			#0L=#2L-#4L	2.0	2.5
			D800L=D802L-D804L		
			U3E1\G10000L=U3E1\G10002L-U3E1\G10004L	2.5	3.5
			#0F=#4F-#8F	2.0	3.0
			D800F=D804F-D808F		
			U3E1\G10000F=U3E1\G10004F-U3E1\G10008F	4.0	5.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Binary operation	*	Multiplication	#0=#1*#2	1.5	2.5
			D800=D801*D802		
			U3E1\G10000=U3E1\G10001*U3E1\G10002	3.0	4.0
			#0L=#2L*#4L	1.5	2.5
			D800L=D802L*D804L		
			U3E1\G10000L=U3E1\G10002L*U3E1\G10004L	2.5	4.0
			#0F=#4F*#8F	2.0	3.5
			D800F=D804F*D808F		
			U3E1\G10000F=U3E1\G10004F*U3E1\G10008F	3.5	5.0
	/	Division	#0=#1/#2	2.0	2.5
			D800=D801/D802		
			U3E1\G10000=U3E1\G10001/U3E1\G10002	3.0	3.5
			#0L=#2L/#4L	2.0	2.5
			D800L=D802L/D804L		
			U3E1\G10000L=U3E1\G10002L/U3E1\G10004L	3.0	3.5
			#0F=#4F/#8F	2.0	3.5
			D800F=D804F/D808F		
			U3E1\G10000F=U3E1\G10004F/U3E1\G10008F	4.0	4.5
	%	Remainder	#0=#1%#2	2.0	2.5
			D800=D801%D802		
			U3E1\G10000=U3E1\G10001%U3E1\G10002	3.0	3.0
			#0L=#2L%#4L	2.0	2.5
			D800L=D802L%D804L		
			U3E1\G10000L=U3E1\G10002L%U3E1\G10004L	3.0	3.5
Bit operation	~	Bit inversion (complement)	#0=~#1	1.5	1.5
			D800=~D801		
			U3E1\G10000=~U3E1\G10001	2.0	2.0
			#0L=~#2L	1.5	1.5
			D800L=~D802L		
			U3E1\G10000L=~U3E1\G10002L	2.0	2.5
	&	Bit logical AND	#0=#1&#2	1.5	2.5
			D800=D801&D802		
			U3E1\G10000=U3E1\G10001&U3E1\G10002	3.0	3.5
			#0L=#2L&#4L	2.0	2.0
			D800L=D802L&D804L		
			U3E1\G10000L=U3E1\G10002L&U3E1\G10004L	2.5	3.5
	|	Bit logical OR	#0=#1|#2	2.0	2.0
			D800=D801|D802		
			U3E1\G10000=U3E1\G10001|U3E1\G10002	2.5	3.0
			#0L=#2L|#4L	2.0	2.0
			D800L=D802L|D804L		
			U3E1\G10000L=U3E1\G10002L|U3E1\G10004L	2.5	3.0
	^	Bit exclusive OR	#0=#1^#2	1.5	2.0
			D800=D801^D802		
			U3E1\G10000=U3E1\G10001^U3E1\G10002	3.0	3.0
			#0L=#2L^#4L	1.5	2.0
			D800L=D802L^D804L		
			U3E1\G10000L=U3E1\G10002L^U3E1\G10004L	3.0	3.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Bit operation	>>	Bit right shift	#0=#1>>#2	1.5	2.5
			D800=D801>>D802		
			U3E1\G10000`=U3E1\G10001>>U3E1\G10002	2.5	3.5
			#0L=#2L>>#4L	2.0	2.5
			D800L=D802L>>D804L		
			U3E1\G10000L=U3E1\G10002L>>U3E1\G10004L	3.0	3.0
	<<	Bit left shift	#0=#1<<#2	2.0	2.0
			D800=D801<<D802		
			U3E1\G10000=U3E1\G10001<<U3E1\G10002	3.0	3.5
			#0L=#2L<<#4L	2.0	2.0
			D800L=D802L<<D804L		
			U3E1\G10000L=U3E1\G10002L<<U3E1\G10004L	2.5	3.0
Sign	-	Sign inversion (complement of 2)	#0=-#1	1.5	1.5
			D800=-D812		
			U3E1\G10000=-U3E1\G10001	2.0	2.5
			#0L=-#2L	1.5	2.0
			D800L=-D802L		
			U3E1\G10000L=-U3E1\G10002L	2.0	2.5
			#0F=-#4F	1.5	2.0
			D800F=-D804F		
			U3E1\G10000F=-U3E1\G10004F	2.5	3.0
Standard function	SIN	Sine	#0F=SIN(#4F)	4.0	4.5
			D800F=SIN(D804F)		
			U3E1\G10000F=SIN(U3E1\G10004F)	5.0	5.5
	COS	Cosine	#0F=COS(#4F)	3.0	4.5
			D800F=COS(D804F)		
			U3E1\G10000F=COS(U3E1\G10004F)	4.0	5.5
	TAN	Tangent	#0F=TAN(#4F)	6.0	6.0
			D800F=TAN(D804F)		
			U3E1\G10000F=TAN(U3E1\G10004F)	7.0	7.0
	ASIN	Arcsine	#0F=ASIN(#4F)	9.0	12.5
			D800F=ASIN(D804F)		
			U3E1\G10000F=ASIN(U3E1\G10004F)	10.5	14.5
	ACOS	Arccosine	#0F=ACOS(#4F)	7.0	10.5
			D800F=ACOS(D804F)		
			U3E1\G10000F=ACOS(U3E1\G10004F)	7.5	11.5
	ATAN	Arctangent	#0F=ATAN(#4F)	3.5	4.5
			D800F=ATAN(D804F)		
			U3E1\G10000F=ATAN(U3E1\G10004F)	4.0	6.0
	SQRT	Square root	#0F=SQRT(#4F)	1.5	2.5
			D800F=SQRT(D804F)		
			U3E1\G10000F=SQRT(U3E1\G10004F)	2.5	3.5
	LN	Natural logarithm	#0F=LN(#4F)	4.5	5.5
			D800F=LN(D804F)		
			U3E1\G10000F=LN(U3E1\G10004F)	5.5	5.5
	EXP	Exponential operation	#0F=EXP(#4F)	3.0	4.0
			D800F=EXP(D804F)		
			U3E1\G10000F=EXP(U3E1\G10004F)	4.0	4.5

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Standard function	ABS	Absolute value	#0F=ABS(#4F)	1.5	2.0
			D800F=ABS(D804F)		
			U3E1\G10000F=ABS(U3E1\G10004F)	2.5	3.0
	RND	Round-off	#0F=RND(#4F)	2.0	2.5
			D800F=RND(D804F)		
			U3E1\G10000F=RND(U3E1\G10004F)	3.0	3.5
	FIX	Round-down	#0F=FIX(#4F)	2.0	2.5
			D800F=FIX(D804F)		
			U3E1\G10000F=FIX(U3E1\G10004F)	2.5	3.5
	FUP	Round-up	#0F=FUP(#4F)	2.5	2.5
			D800F=FUP(D804F)		
			U3E1\G10000F=FUP(U3E1\G10004F)	3.0	3.5
	BIN	BCD→BIN conversion	#0=BIN(#1)	1.5	2.0
			D800=BIN(D801)		
			U3E1\G10000=BIN(U3E1\G10001)	2.5	2.5
			#0L=BIN(#2L)	2.0	2.5
			D800L=BIN(D802L)		
			U3E1\G10000L=BIN(U3E1\G10002L)	2.5	3.0
	BCD	BIN→BCD conversion	#0=BCD(#1)	2.0	2.0
			D800=BCD(D801)		
			U3E1\G10000=BCD(U3E1\G10001)	2.5	3.0
			#0L=BCD(#2L)	2.5	2.5
			D800L=BCD(D802L)		
			U3E1\G10000L=BCD(U3E1\G10002L)	3.0	3.5
Type conversion	SHORT	Converted into 16-bit integer type (signed)	#0=SHORT(#2L)	2.0	2.0
			D800=SHORT(D802L)		
			U3E1\G10000=SHORT(U3E1\G10002L)	2.5	2.5
			#0=SHORT(#4F)	2.5	2.5
			D800=SHORT(D804F)		
			U3E1\G10000=SHORT(U3E1\G10004F)	3.0	3.5
	USHORT	Converted into 16-bit integer type (unsigned)	#0=USHORT(#2L)	2.0	2.0
			D800=USHORT(D802L)		
			U3E1\G10000=USHORT(U3E1\G10002L)	2.0	2.5
			#0=USHORT(#4F)	2.0	2.5
			D800=USHORT(D804F)		
			U3E1\G10000=USHORT(U3E1\G10004F)	3.0	3.5
	LONG	Converted into 32-bit integer type (signed)	#0L=LONG(#2)	1.5	2.0
			D800L=LONG(D802)		
			U3E1\G10000L=LONG(U3E1\G10002)	2.0	2.5
			#0L=LONG(#4F)	2.0	3.0
			D800L=LONG(D804F)		
			U3E1\G10000L=LONG(U3E1\G10004F)	3.0	3.5
	ULONG	Converted into 32-bit integer type (unsigned)	#0L=ULONG(#2)	2.0	2.0
			D800L=ULONG(D802)		
			U3E1\G10000L=ULONG(U3E1\G10002)	2.0	2.5
			#0L=ULONG(#4F)	2.5	3.0
			D800L=ULONG(D804F)		
			U3E1\G10000L=ULONG(U3E1\G10004F)	3.0	4.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Type conversion	FLOAT	Converted into 64-bit floating point type (signed)	#0F=FLOAT(#4)	1.5	2.0
			D800F=FLOAT(D804)		
			U3E1\G10000F=FLOAT(U3E1\G10004)	1.5	2.5
			#0F=FLOAT(#4L)	2.0	2.0
			D800F=FLOAT(D804L)		
			U3E1\G10000F=FLOAT(U3E1\G10004L)	2.5	3.0
	UFLOAT	Converted into 64-bit floating point type (unsigned)	#0F=UFLOAT(#4)	1.5	2.0
			D800F=UFLOAT(D804)		
			U3E1\G10000F=UFLOAT(U3E1\G10004)	2.0	2.5
			#0F=UFLOAT(#4L)	1.5	2.0
			D800F=UFLOAT(D804L)		
			U3E1\G10000F=UFLOAT(U3E1\G10004L)	2.0	2.5
	DFLT	Floating-point value conversion 32-bit into 64-bit	#0F=DFLT(#4L)	2.0	3.5
			D2000F=DFLT(D2004L)		
			U3E1\G10000F=DFLT(U3E1\G10004L)	2.0	3.5
	SFLT	Floating-point value conversion 64-bit into 32-bit	#0L=SFLT(#2F)	2.5	3.5
			D2000L=SFLT(D2002F)		
			U3E1\G10000L=SFLT(U3E1\G10002F)	3.0	3.5
Bit device status	(None)	ON (normally open contact) (Completion of condition)	SET M1000 = M0	2.5	3.0
			SET M1000 = X100		
			SET M1000 = PX0	5.0	5.0
			SET M1000 = U3E1\G10000.0	3.5	3.5
	!	OFF (normally closed contact) (Completion of condition)	SET M1000 = !M0	2.5	3.0
			SET M1000 = !X100		
			SET M1000 = !PX0	4.5	4.5
			SET M1000 = !U3E1\G10000.0	2.5	5.0
Bit device control	SET	Device set	SET M1000	2.0	2.5
			SET Y100		
			SET PY0	2.0	2.0
			SET U3E1\G11000.0	1.5	2.5
	RST	Device reset	RST M1000	2.0	2.5
			RST Y100		
			RST PY0	2.0	2.0
			RST U3E1\G11000.0	2.5	3.0
	DOUT	Device output	DOUT M0,#0	2.5	2.5
			DOUT Y100,#0		
			DOUT PY0,#0	2.0	2.5
			DOUT M0,#0L	3.5	3.5
			DOUT Y100,#0L		
			DOUT PY0,#0L	3.0	3.5
	DIN	Device input	DIN #0,M0	2.0	2.5
			DIN #0,X0		
			DIN #0,PX0	4.5	4.5
			DIN #0L,M0	2.5	3.0
			DIN #0L,X0		
			DIN #0L,PX0	6.5	9.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Bit device control	OUT	Bit device output	OUT M100 = M0	2.0	2.5
			OUT Y0 = M0		
			OUT PY0 = M0	2.0	2.5
			OUT U3E1\G10000.0 = M0	2.5	3.5
Logical operation	*	Logical AND	SET M1000 = M0*M1	2.5	3.5
			SET M1000 = X100*X101		
			SET M1000 = PX0*PX1	6.0	6.5
			SET M1000 = U3E1\G10000.0*U3E1\G10000.1	2.5	3.5
	+	Logical OR	SET M1000 = M0+M1	2.5	3.5
			SET M1000 = X100+X101		
			SET M1000 = PX0+PX1	6.5	9.0
			SET M1000 = U3E1\G10000.0+U3E1\G10000.1	2.5	3.5
Comparison operation	==	Equal to (Completion of condition)	SET M1000 = #0==#1	2.5	3.5
			SET M1000 = D800==D801		
			SET M1000 = U3E1\G10000==U3E1\G10001	3.5	4.5
			SET M1000 = #0L==#2L	2.5	4.0
			SET M1000 = D800L==D802L		
			SET M1000 = U3E1\G10000L==U3E1\G10002L	3.5	4.5
			SET M1000 = #0F==#4F	3.0	4.5
			SET M1000 = D800F==D804F		
			SET M1000 = U3E1\G10000F==U3E1\G10004F	4.5	6.0
	!=	Not equal to (Completion of condition)	SET M1000 = #0!=#1	2.5	4.0
			SET M1000 = D800!=D801		
			SET M1000 = U3E1\G10000!=U3E1\G10001	3.5	4.5
			SET M1000 = #0L!=#2L	3.0	4.0
			SET M1000 = D800L!=D802L		
			SET M1000 = U3E1\G10000L!=U3E1\G10002L	3.0	4.5
			SET M1000 = #0F!=#4F	3.0	4.5
			SET M1000 = D800F!=D804F		
			SET M1000 = U3E1\G10000F!=U3E1\G10004F	4.5	6.0
	<	Less than (Completion of condition)	SET M1000 = #0<#1	3.0	4.0
			SET M1000 = D800<D801		
			SET M1000 = U3E1\G10000<U3E1\G10001	4.0	4.5
			SET M1000 = #0L<#2L	3.5	4.0
			SET M1000 = D800L<D802L		
			SET M1000 = U3E1\G10000L<U3E1\G10002L	4.0	4.5
			SET M1000 = #0F<#4F	3.5	4.5
			SET M1000 = D800F<D804F		
			SET M1000 = U3E1\G10000F<U3E1\G10004F	5.0	6.0
	<=	Less than or equal to (Completion of condition)	SET M1000 = #0<=#1	3.5	3.5
			SET M1000 = D800<=D801		
			SET M1000 = U3E1\G10000<=U3E1\G10001	4.5	4.5
			SET M1000 = #0L<=#2L	3.5	4.0
			SET M1000 = D800L<=D802L		
			SET M1000 = U3E1\G10000L<=U3E1\G10002L	4.0	4.5
			SET M1000 = #0F<=#4F	3.5	4.5
			SET M1000 = D800F<=D804F		
			SET M1000 = U3E1\G10000F<=U3E1\G10004F	5.0	6.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Comparison operation	>	More than (Completion of condition)	SET M1000 = #0>#1	3.0	4.0
			SET M1000 = D800>D801		
			SET M1000 = U3E1\G10000>U3E1\G10001	4.5	4.5
			SET M1000 = #0L>#2L	3.0	4.0
			SET M1000 = D800L>D802L		
			SET M1000 = U3E1\G10000L>U3E1\G10002L	4.0	4.5
			SET M1000 = #0F>#4F	3.5	4.5
			SET M1000 = D800F>D804F		
			SET M1000 = U3E1\G10000F>U3E1\G10004F	5.0	6.0
	>=	More than or equal to (Completion of condition)	SET M1000 = #0>=#1	3.5	4.0
			SET M1000 = D800>=D801		
			SET M1000 = U3E1\G10000>=U3E1\G10001	4.5	4.5
			SET M1000 = #0L>=#2L	3.5	4.0
			SET M1000 = D800L>=D802L		
			SET M1000 = U3E1\G10000L>=U3E1\G10002L	4.0	5.0
			SET M1000 = #0F>=#4F	3.5	4.5
			SET M1000 = D800F>=D804F		
			SET M1000 = U3E1\G10000F>=U3E1\G10004F	5.0	6.0
Motion dedicated function	CHGV	Speed change request	CHGV(K1,#0)	3.0	3.5
			CHGV(K1,D800)		
			CHGV(K1,U3E1\G10000)	4.0	4.5
			CHGV(K1,#0L)	3.0	3.5
			CHGV(K1,D800L)		
			CHGV(K1,U3E1\G10000L)	3.5	3.5
	CHGVS	Command generation axis speed change request	CHGVS(K1,#0)	2.5	
			CHGVS(K1,D800)		
			CHGVS(K1,U3E1\G10000)	3.5	
			CHGVS(K1,#0L)	2.5	
			CHGVS(K1,D800L)		
			CHGVS(K1,U3E1\G10000L)	3.5	
	CHGT	Torque limit value change request	CHGT(K1,#0)	1.5	2.0
			CHGT(K1,D800)		
			CHGT(K1,U3E1\G10000)	2.5	2.5
			CHGT(K1,#0L)	2.0	2.5
			CHGT(K1,D800L)		
			CHGT(K1,U3E1\G10000L)	2.5	3.0
	CHGT2	Torque limit value individual change request	CHGT2(K1,#0,#1)	2.0	
			CHGT2(K1,D800,D801)		
			CHGT2(K1,U3E1\G10000,U3E1\G10001)	3.0	
			CHGT2(K1,#0L,#2L)	2.5	
			CHGT2(K1,D800L,D802L)		
			CHGT2(K1,U3E1\G10000L,U3E1\G10002L)	3.5	
	CHGP	Target position change request	CHGP(K1,K1,#0) (Note-1)	3.0	
			CHGP(K1,K1,D800) (Note-1)		
			CHGP(K1,K1,U3E1\G10000) (Note-1)	4.0	
			CHGP(K1,K1,#0) (Note-2)	3.5	
			CHGP(K1,K1,D800) (Note-2)		
			CHGP(K1,K1,U3E1\G10000) (Note-2)	5.0	

(Note-1): 1-axis linear positioning control
(Note-2): 4-axes linear interpolation control

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Others	EI	Event task enable	EI	0.5	0.5
	DI	Event task disable	DI	0.5	0.5
	NOP	No operation	NOP	0.5	0.5
	BMOV	Block transfer	BMOV #0,#100,K10	4.5	5.5
			BMOV D800,D100,K10		
			BMOV U3E1\G10000,U3E1\G10100,K10	7.5	7.5
			BMOV #0,#100,K100	19.0	19.0
			BMOV D800,D100,K100		
			BMOV U3E1\G10000,U3E1\G10100,K100	28.0	28.0
			BMOV N1,#0,K512	123.5	123.5
			BMOV N1,D800,K512		
			BMOV N1,U3E1\G10000,K512	250.0	250.5
	FMOV	Same data block transfer	FMOV #0,#100,K10	3.0	3.5
			FMOV D800,D100,K10		
			FMOV U3E1\G10000,U3E1\G10100,K10	2.5	4.0
			FMOV #0,#100,K100	7.5	7.5
			FMOV D800,D100,K100		
			FMOV U3E1\G10000,U3E1\G10100,K100	2.5	5.0
	MULTW	Write device data to CPU shared memory of the self CPU	MULTW H800,#0,K1,M0	4.0	4.0
			MULTW H800,D800,K1,M0		
			MULTW H800,U3E1\G10000,K1,M0	5.0	5.0
			MULTW H800,#0,K10,M0	5.5	5.5
			MULTW H800,D800,K10,M0		
			MULTW H800,U3E1\G10000,K10,M0	9.5	9.5
			MULTW H800,#0,K100,M0	23.5	23.5
			MULTW H800,D800,K100,M0		
			MULTW H800,U3E1\G10000,K100,M0	61.0	61.0
			MULTW H800,#0,K256,M0	58.0	58.0
			MULTW H800,D800,K256,M0		
			MULTW H800,U3E1\G10000,K256,M0	151.5	151.5
	MULTR	Read device data from CPU shared memory	MULTR #0,H3E0,H800,K1	18.0	20.5
			MULTR D800,H3E0,H800,K1		
			MULTR U3E1\G10000,H3E0,H800,K1	18.5	22.0
			MULTR #0,H3E0,H800,K10	27.0	30.5
			MULTR D800,H3E0,H800,K10		
			MULTR U3E1\G10000,H3E0,H800,K10	27.5	31.5
			MULTR #0,H3E0,H800,K100	139.5	140.5
			MULTR D800,H3E0,H800,K100		
			MULTR U3E1\G10000,H3E0,H800,K100	148.5	152.0
			MULTR #0,H3E0,H800,K256	326.5	412.0
			MULTR D800,H3E0,H800,K256		
			MULTR U3E1\G10000,H3E0,H800,K256	350.0	435.0

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Others	TO	Write device data to intelligent function module	TO H0,H0,#0,K1	12.5	15.5
			TO H0,H0,D800,K1		
			TO H0,H0,U3E1\G10000,K1	13.5	16.0
			TO H0,H0,#0,K10	15.0	18.5
			TO H0,H0,D800,K10		
			TO H0,H0,U3E1\G10000,K10	19.0	22.0
			TO H0,H0,#0,K100	80.0	84.0
			TO H0,H0,D800,K100		
			TO H0,H0,U3E1\G10000,K100	117.0	121.5
			TO H0,H0,#0,K256	181.5	224.0
			TO H0,H0,D800,K256		
			TO H0,H0,U3E1\G10000,K256	277.0	358.5
	FROM	Read device data from intelligent function module	FROM #0,H0,H0,K1	11.0	14.5
			FROM D800,H0,H0,K1		
			FROM U3E1\G10000,H0,H0,K1	12.5	16.5
			FROM #0,H0,#0,K10	20.0	22.5
			FROM D800,H0,H0,K10		
			FROM U3E1\G10000,H0,H0,K10	21.0	23.0
			FROM #0,H0,#0,K100	132.0	132.0
			FROM D800,H0,H0,K100		
			FROM U3E1\G10000,H0,H0,K100	141.0	144.5
			FROM #0,H0,H0,K256	319.5	405.0
			FROM D800,H0,H0,K256		
			FROM U3E1\G10000,H0,H0,K256	343.0	432.0
	RTO	Write buffer memory data to head module	RTO #4000,#4001,#4002,#0,K1,M0	5.0 (Note-3)	
			RTO D2000,D2001,D2002,D800,K1,M0		
			RTO U3E1\G12000,U3E1\G12001,U3E1\G12002, U3E1\G10000,K1,M0	7.5 (Note-3)	
			RTO #4000,#4001,#4002,#0,K10,M0	5.5 (Note-3)	
			RTO D2000,D2001,D2002,D800,K10,M0		
			RTO U3E1\G12000,U3E1\G12001,U3E1\G12002, U3E1\G10000,K10,M0	6.0 (Note-3)	
			RTO #4000,#4001,#4002,#0,K100,M0	5.5 (Note-3)	
			RTO D2000,D2001,D2002,D800,K100,M0		
			RTO U3E1\G12000,U3E1\G12001,U3E1\G12002, U3E1\G10000,K100,M0	6.5 (Note-3)	
			RTO #4000,#4001,#4002,#0,K240,M0	6.0 (Note-3)	
			RTO D2000,D2001,D2002,D800,K240,M0		
			RTO U3E1\G12000,U3E1\G12001,U3E1\G12002, U3E1\G10000,K240,M0	6.5 (Note-3)	

(Note-3): This is the Motion CPU processing time, and does not include the time to complete data transfer.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Others	RFROM	Read buffer memory data from head module	RFROM #0,#4000,#4001,#4002,K1,M0	3.5 (Note-3)	
			RFROM D800,D2000,D2001,D2002,K1,M0		
			RFROM U3E1\G10000,U3E1\G12000, U3E1\G12001,U3E1\G12002,K1,M0	7.0 (Note-3)	
			RFROM #0,#4000,#4001,#4002,K10,M0	4.0 (Note-3)	
			RFROM D800,D2000,D2001,D2002,K10,M0		
			RFROM U3E1\G10000,U3E1\G12000, U3E1\G12001,U3E1\G12002,K10,M0	5.0 (Note-3)	
			RFROM #0,#4000,#4001,#4002,K100,M0	4.0 (Note-3)	
			RFROM D800,D2000,D2001,D2002,K100,M0		
			RFROM U3E1\G10000,U3E1\G12000, U3E1\G12001,U3E1\G12002,K100,M0	5.0 (Note-3)	
			RFROM #0,#4000,#4001,#4002,K240,M0	4.0 (Note-3)	
			RFROM D800,D2000,D2001,D2002,K240,M0		
			RFROM U3E1\G10000,U3E1\G12000, U3E1\G12001,U3E1\G12002,K240,M0	5.5 (Note-3)	
	TIME	Time to wait	TIME K1	2.5	2.5
			TIME #0	2.0	2.5
			TIME D800		
			TIME U3E1\G10000	3.5	3.5
Vision system dedicated function	MVOPEN	Open line	MVOPEN K1,K1000	3.0	5.5
			MVOPEN #0,#1	4.5	7.0
			MVOPEN D2000,D2001		
			MVOPEN U3E1\G10000,U3E1\G10001	5.0	7.5
	MVLOAD	Load a program	MVLOAD K1,K1000	3.0	5.0
			MVLOAD #0,#1	3.5	5.5
			MVLOAD D2000,D2001		
			MVLOAD U3E1\G10000,U3E1\G10001	5.0	7.0
	MVTRG	Send an image acquisition trigger	MVTRG K1,K1000	1.5	4.5
			MVTRG #0,#1	3.0	5.0
			MVTRG D2000,D2001		
			MVTRG U3E1\G10000,U3E1\G10001	3.5	6.5
	MVPST	Start a program	MVPST K1,K1000	3.0	5.0
			MVPST #0,#1	4.5	6.5
			MVPST D2000,D2001		
			MVPST U3E1\G10000,U3E1\G10001	5.0	6.5
	MVIN	Input data	MVIN K1,"A1",#0L,K1000	4.0	7.5
			MVIN D2000,D2001,#0L,K1000 (Note-4)	7.0	11.5
			MVIN D2000,D2001,#0L,K1000 (Note-5)	12.5	17.5
			MVIN U3E1\G10000,U3E1\G100001, U3E1\G10020L,K1000 (Note-5)	35.0	39.5

(Note-3): This is the Motion CPU processing time, and does not include the time to complete data transfer.
(Note-4): (S2) in MVIN (S1), (S2), (D) and (S3) are set by 2 bytes character sequence.
(Note-5): (S2) in MVIN (S1), (S2), (D) and (S3) are set by 32 bytes character sequence.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Vision system dedicated function	MVOUT	Output data	MVOUT K1,"A1",#0L,K1000	8.0	14.5
			MVOUT D2000,D2001,#0L,K1000 (Note-6)	15.0	22.0
			MVOUT D2000,D2001,#0L,K1000 (Note-7)	18.0	25.0
			MVOUT U3E1\G10000,U3E1\G10001, U3E1\G10020L,K1000 (Note-7)	41.5	48.0
	MVFIN	Reset a status storage device	MVFIN K1	2.0	3.0
			MVFIN #0	3.0	4.0
			MVFIN D2000		
			MVFIN U3E1\G10000	3.0	4.0
	MVCLOSE	Close line	MVCLOSE K1	129.5	176.0
			MVCLOSE #0	136.0	183.0
			MVCLOSE D2000		
			MVCLOSE U3E1\G10000	129.5	184.5
	MVCOM	Send a command for native mode	MVCOM K1,"GO",#0,K0,K1000	7.0	9.5
			MVCOM D2000,D2001,#0,D2100,K1000 (Note-8)	12.0	13.5
			MVCOM D2000,D2001,#0,D2100,K1000 (Note-9)	56.5	64.5
			MVCOM U3E1\G10000,U3E1\G10002, U3E1\G11000,U3E1\G10001,K1000 (Note-9)	183.0	191.5
Data control	SCL	16-bit integer type scaling	SCL K0,K2000,#0,#2002 (Note-10)	7.0	
			SCL K0,K2000,D2000,D4002 (Note-10)		
			SCL K0,K2000,U3E1\G10000,U3E1\G12002 (Note-10)	15.5	
			SCL K0,K2000,#0,#2002 (Note-11)	37.0	
			SCL K0,K2000,D2000,D4002 (Note-11)		
			SCL K0,K2000,U3E1\G10000,U3E1\G12002 (Note-11)	104.0	
			SCL K0,K2000,#0,#2002 (Note-12)	334.0	
			SCL K0,K2000,D2000,D4002 (Note-12)		
			SCL K0,K2000,U3E1\G10000,U3E1\G12002 (Note-12)	1030.5	
			SCL K2,K1,#0,#2002 (Note-10)	6.5	
			SCL K2,K1,D2000,D4002 (Note-10)		
			SCL K2,K1,U3E1\G10000,U3E1\G12002 (Note-10)	12.0	

(Note-6): (S2) in MVOUT (S1), (S2), (S3) and (S4) are set by 2 bytes character sequence.
(Note-7): (S2) in MVOUT (S1), (S2), (S3) and (S4) are set by 32 bytes character sequence.
(Note-8): (S2) in MVCOM (S1), (S2), (D), (S3) and (S4) are set by 2 bytes character sequence.
(Note-9): (S2) in MVCOM (S1), (S2), (D), (S3) and (S4) are set by 191 bytes character sequence.
(Note-10): Number of searches of data conversion for scaling is 10 times.
(Note-11): Number of searches of data conversion for scaling is 100 times.
(Note-12): Number of searches of data conversion for scaling is 1000 times.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Data control	DSCL	32-bit integer type scaling	DSCL K0,K2000L,#0,#4002L (Note-10) DSCL K0,K2000L,D2000,D6002L (Note-10)	7.5	
			DSCL K0,K2000L,U3E1\G10000,U3E1\G14002L (Note-10)	15.5	
			DSCL K0,K2000L,#0,#4002L (Note-11) DSCL K0,K2000L,D2000,D6002L (Note-11)	37.5	
			DSCL K0,K2000L,U3E1\G10000,U3E1\G14002L (Note-11)	104.5	
			DSCL K0,K2000L,#0,#4002L (Note-12) DSCL K0,K2000L,D2000,D6002L (Note-12)	334.5	
			DSCL K0,K2000L,U3E1\G10000,U3E1\G14002L (Note-12)	1031.5	
			DSCL K2,K1L,#0,#4002L (Note-10) DSCL K2,K1L,D2000,D6002L (Note-10)	7.0	
			DSCL K2,K1L,U3E1\G10000,U3E1\G14002L (Note-10)	12.5	
Program control	IF - ELSE - IEND	Conditional branch control	IF #0 == #1 (Note-13) #2 = #3 ELSE #4 = #5 IEND IF D800 == D801 (Note-13) #2 = #3 ELSE #4 = #5 IEND	2.0	3.5
			IF U3E1\G10000 == U3E1\G10001 (Note-13) #2 = #3 ELSE #4 = #5 IEND	3.5	4.5
			IF #0 == #1 (Note-14) #2 = #3 ELSE #4 = #5 IEND IF D800 == D801 (Note-14) #2 = #3 ELSE #4 = #5 IEND	2.0	3.5
			IF U3E1\G10000 == U3E1\G10001 (Note-14) #2 = #3 ELSE #4 = #5 IEND	3.0	4.5

(Note-10): Number of searches of data conversion for scaling is 10 times.
(Note-11): Number of searches of data conversion for scaling is 100 times.
(Note-12): Number of searches of data conversion for scaling is 1000 times.
(Note-13): (S) in IF - ELSE - IEND are set by true data.
(Note-14): (S) in IF - ELSE - IEND are set by false data.

Processing time of operation instructions (Continued)

<table>
<tr><th>Classifi-cations</th><th>Symbol</th><th>Instruction</th><th>Operation expression</th><th>Q173DSCPU/
Q172DSCPU
Unit [μs]</th><th>Q173DCPU(-S1)/
Q172DCPU(-S1)
Unit [μs]</th></tr>
<tr><td rowspan="4">Program control</td><td rowspan="4">SELECT - CASE - SEND</td><td rowspan="4">Selective branch control</td><td>SELECT (Note-15)
CASE #0 == K1
#2 = #3
CEND
CASE #1 == K1
#4 = #5
CEND
CELSE
#6 = #7
CEND
SEND</td><td rowspan="2">2.0</td><td rowspan="2">4.0</td></tr>
<tr><td>SELECT (Note-15)
CASE D800 == K1
#2 = #3
CEND
CASE D801 == K1
#4 = #5
CEND
CELSE
#6 = #7
CEND
SEND</td></tr>
<tr><td>SELECT (Note-15)
CASE U3E1\G10000 == K1
#2 = #3
CEND
CASE U3E1\G10001 == K1
#4 = #5
CEND
CELSE
#6 = #7
CEND
SEND</td><td>2.5</td><td>4.5</td></tr>
<tr><td>SELECT (Note-16)
CASE #0 == K1
#2 = #3
CEND
CASE #1 == K1
#4 = #5
CEND
CELSE
#6 = #7
CEND
SEND</td><td>3.0</td><td>5.0</td></tr>
</table>

(Note-15): For SELECT - CASE(S1) - CEND CASE(S2) - CEND CELSE - CEND SEND, (S1) are set by true data.

(Note-16): For SELECT - CASE(S1) - CEND CASE(S2) - CEND CELSE - CEND SEND, (S1) are set by false data and (S2) are set by true data.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Program control	SELECT - CASE - SEND	Selective branch control	SELECT (Note-16) CASE D800 == K1 #2 = #3 CEND CASE D801 == K1 #4 = #5 CEND CELSE #6 = #7 CEND SEND	3.0	5.0
			SELECT (Note-16) CASE U3E1\G10000 == K1 #2 = #3 CEND CASE U3E1\G10001 == K1 #4 = #5 CEND CELSE #6 = #7 CEND SEND	4.0	6.0
			SELECT (Note-17) CASE #0 == K1 #2 = #3 CEND CASE #1 == K1 #4 = #5 CEND CELSE #6 = #7 CEND SEND	3.0	5.5
			SELECT (Note-17) CASE D800 == K1 #2 = #3 CEND CASE D801 == K1 #4 = #5 CEND CELSE #6 = #7 CEND SEND		

(Note-16): For SELECT - CASE(S1) - CEND CASE(S2) - CEND CELSE - CEND SEND, (S1) are set by false data and (S2) are set by true data.

(Note-17): For SELECT - CASE(S1) - CEND CASE(S2) - CEND CELSE - CEND SEND, (S1) and (S2) are set by false data.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Program control	SELECT - CASE - SEND	Selective branch control	SELECT (Note-17) CASE U3E1\G10000 == K1 #2 = #3 CEND CASE U3E1\G10001 == K1 #4 = #5 CEND CELSE #6 = #7 CEND SEND	4.0	6.5
	FOR - NEXT	Repeat control with specified count	FOR #0 = K1 TO 10 #1 = #1 + 1 NEXT	32.0	58.5
			FOR D800 = K1 TO 10 #1 = #1 + 1 NEXT		
			FOR U3E1\G10000 = K1 TO 10 #1 = #1 + 1 NEXT	41.0	71.0
Synchronous control dedicated function	CAMRD	Cam data read	CAMRD #0,#2L,K256,#4 (Note-18)	28.0	
			CAMRD D2000,D2002L,K256,D2004 (Note-18)		
			CAMRD U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-18)	24.0	
			CAMRD #0,#2L,K1024,#4 (Note-18)	88.0	
			CAMRD D2000,D2002L,K1024,D2004 (Note-18)		
			CAMRD U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-18)	69.5	
			CAMRD #0,#2L,K2048,#4 (Note-18)	169.0	
			CAMRD D2000,D2002L,K2048,D2004 (Note-18)		
			CAMRD U3E1\G10000,U3E1\G10002L,K2048, U3E1\G10004 (Note-18)	131.5	
			CAMRD #0,#2L,K256,#4 (Note-19)	47.0	
			CAMRD D2000,D2002L,K256,D2004 (Note-19)		
			CAMRD U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-19)	38.5	
			CAMRD #0,#2L,K512,#4 (Note-19)	87.5	
			CAMRD D2000,D2002L,K512,D2004 (Note-19)		
			CAMRD U3E1\G10000,U3E1\G10002L,K512, U3E1\G10004 (Note-19)	69.0	
			CAMRD #0,#2L,K1024,#4 (Note-19)	168.0	
			CAMRD D2000,D2002L,K1024,D2004 (Note-19)		
			CAMRD U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-19)	130.0	

(Note-17): For SELECT - CASE(S1) - CEND CASE(S2) - CEND CELSE - CEND SEND, (S1) and (S2) are set by false data.
(Note-18): The cam data is in the stroke ratio data format.
(Note-19): The cam data is in the coordinate data format.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Synchronous control dedicated function	CAMWR	Cam data write	CAMWR #0,#2L,K256,#4 (Note-18)	62.5	
			CAMWR D2000,D2002L,K256,D2004 (Note-18)		
			CAMWR U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-18)	104.0	
			CAMWR #0,#2L,K1024,#4 (Note-18)	207.5	
			CAMWR D2000,D2002L,K1024,D2004 (Note-18)		
			CAMWR U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-18)	370.0	
			CAMWR #0,#2L,K2048,#4 (Note-18)	417.0	
			CAMWR D2000,D2002L,K2048,D2004 (Note-18)		
			CAMWR U3E1\G10000,U3E1\G10002L,K2048, U3E1\G10004 (Note-18)	757.0	
			CAMWR #0,#2L,K256,#4 (Note-19)	116.5	
			CAMWR D2000,D2002L,K256,D2004 (Note-19)		
			CAMWR U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-19)	189.0	
			CAMWR #0,#2L,K512,#4 (Note-19)	221.5	
			CAMWR D2000,D2002L,K512,D2004 (Note-19)		
			CAMWR U3E1\G10000,U3E1\G10002L,K512, U3E1\G10004 (Note-19)	375.0	
			CAMWR #0,#2L,K1024,#4 (Note-19)	447.0	
			CAMWR D2000,D2002L,K1024,D2004 (Note-19)		
			CAMWR U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-19)	776.0	
	CAMWR2	Cam data write (Cam open area)	CAMWR2 #0,#2L,K256,#4 (Note-18)	35.5	
			CAMWR2 D2000,D2002L,K256,D2004 (Note-18)		
			CAMWR2 U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-18)	74.0	
			CAMWR2 #0,#2L,K1024,#4 (Note-18)	121.0	
			CAMWR2 D2000,D2002L,K1024,D2004 (Note-18)		
			CAMWR2 U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-18)	264.0	
			CAMWR2 #0,#2L,K2048,#4 (Note-18)	249.5	
			CAMWR2 D2000,D2002L,K2048,D2004 (Note-18)		
			CAMWR2 U3E1\G10000,U3E1\G10002L,K2048, U3E1\G10004 (Note-18)	536.0	
			CAMWR2 #0,#2L,K256,#4 (Note-19)	70.0	
			CAMWR2 D2000,D2002L,K256,D2004 (Note-19)		
			CAMWR2 U3E1\G10000,U3E1\G10002L,K256, U3E1\G10004 (Note-19)	143.0	
			CAMWR2 #0,#2L,K512,#4 (Note-19)	134.0	
			CAMWR2 D2000,D2002L,K512,D2004 (Note-19)		
			CAMWR2 U3E1\G10000,U3E1\G10002L,K512, U3E1\G10004 (Note-19)	287.5	
			CAMWR2 #0,#2L,K1024,#4 (Note-19)	279.5	
			CAMWR2 D2000,D2002L,K1024,D2004 (Note-19)		

(Note-18): The cam data is in the stroke ratio data format.
(Note-19): The cam data is in the coordinate data format.

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Synchronous control dedicated function	CAMWR2	Cam data write (Cam open area)	CAMWR2 U3E1\G10000,U3E1\G10002L,K1024, U3E1\G10004 (Note-19)	565.5	
	CAMMK	Cam auto-generation	CAMMK #0,#1,#2 (Note-20)	192.5	
			CAMMK D2000,D2001,D2002 (Note-20)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-20)	207.5	
			CAMMK #0,#1,#2 (Note-21)	5905.0	
			CAMMK D2000,D2001,D2002 (Note-21)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-21)	5908.5	
			CAMMK #0,#1,#2 (Note-22)	23753.5	
			CAMMK D2000,D2001,D2002 (Note-22)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-22)	23755.5	
			CAMMK #0,#1,#2 (Note-23)	170.5	
			CAMMK D2000,D2001,D2002 (Note-23)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-23)	187.5	
			CAMMK #0,#1,#2 (Note-24)	4662.5	
			CAMMK D2000,D2001,D2002 (Note-24)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-24)	4680.5	
			CAMMK #0,#1,#2 (Note-25)	19034.0	
			CAMMK D2000,D2001,D2002 (Note-25)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-25)	19060.0	
			CAMMK #0,#1,#2 (Note-26)	202.5	
			CAMMK D2000,D2001,D2002 (Note-26)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-26)	242.0	
			CAMMK #0,#1,#2 (Note-27)	4611.5	
			CAMMK D2000,D2001,D2002 (Note-27)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-27)	4642.5	

(Note-19): The cam data is in the coordinate data format.
(Note-20): The cam resolution is 256, and the auto-generation option is set to the S-curve acceleration/deceleration system.
(Note-21): The cam resolution is 8192, and the auto-generation option is set to the S-curve acceleration/deceleration system.
(Note-22): The cam resolution is 32768, and the auto-generation option is set to the S-curve acceleration/deceleration system.
(Note-23): The cam auto-generation type is set to easy stroke ratio cam, 8 sections are set, the cam resolution is 256, the cam curve is distorted sine.
(Note-24): The cam auto-generation type is set to easy stroke ratio cam, 8 sections are set, the cam resolution is 8192, the cam curve is distorted sine.
(Note-25): The cam auto-generation type is set to easy stroke ratio cam, 8 sections are set, the cam resolution is 32768, the cam curve is distorted sine.
(Note-26): The cam auto-generation type is set to easy stroke ratio cam, 32 sections are set, the cam resolution is 256, the cam curve is distorted sine.
(Note-27): The cam auto-generation type is set to easy stroke ratio cam, 32 sections are set, the cam resolution is 8192, the cam curve is distorted sine.

Processing time of operation instructions (Continued)

Classifications	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Synchronous control dedicated function	CAMMK	Cam auto-generation	CAMMK #0,#1,#2 (Note-28)	18403.5	
			CAMMK D2000,D2001,D2002 (Note-28)		
			CAMMK U3E1\G10000,U3E1\G10001, U3E1\G10002 (Note-28)	18473.5	
	CAMPSCL	Cam position calculation	CAMPSCL #0,#2,#14L (Note-29), (Note-31)	6.5	
			CAMPSCL D2000,D2002,D2014L (Note-29), (Note-31)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-29), (Note-31)	11.0	
			CAMPSCL #0,#2,#14L (Note-29), (Note-32)	6.5	
			CAMPSCL D2000,D2002,D2014L (Note-29), (Note-32)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-29), (Note-32)	9.0	
			CAMPSCL #0,#2,#14L (Note-29), (Note-33)	7.5	
			CAMPSCL D2000,D2002,D2014L (Note-29), (Note-33)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-29), (Note-33)	11.0	
			CAMPSCL #0,#2,#14L (Note-29), (Note-34)	7.0	
			CAMPSCL D2000,D2002,D2014L (Note-29), (Note-34)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-29), (Note-34)	11.0	

(Note-28): The cam auto-generation type is set to easy stroke ratio cam, 32 sections are set, the cam resolution is 32768, the cam curve is distorted sine.

(Note-29): The cam position calculation type is set to the cam axis current feed value calculation.

(Note-30): The cam position calculation type is set to the cam axis current value per cycle calculation.

(Note-31): The cam data is in the stroke ratio data format, the cam resolution is 256, and the calculation is performed with the midpoint (128).

(Note-32): The cam data is in the stroke ratio data format, the cam resolution is 8192, and the calculation is performed with the midpoint (4096).

(Note-33): The cam data is in the coordinate data format, the coordinates number is 256, and the calculation is performed with the midpoint (128).

(Note-34): The cam data is in the coordinate data format, the coordinates number is 8192, and the calculation is performed with the midpoint (4096).

Processing time of operation instructions (Continued)

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Synchronous control dedicated function	CAMPSCL	Cam position calculation	CAMPSCL #0,#2,#14L (Note-30), (Note-31)	27.5	
			CAMPSCL D2000,D2002,D2014L (Note-30), (Note-31)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-30), (Note-31)	32.5	
			CAMPSCL #0,#2,#14L (Note-30), (Note-32)	631.0	
			CAMPSCL D2000,D2002,D2014L (Note-30), (Note-32)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-30), (Note-32)	644.0	
			CAMPSCL #0,#2,#14L (Note-30), (Note-33)	17.0	
			CAMPSCL D2000,D2002,D2014L (Note-30), (Note-33)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-30), (Note-33)	22.5	
			CAMPSCL #0,#2,#14L (Note-30), (Note-34)	250.0	
			CAMPSCL D2000,D2002,D2014L (Note-30), (Note-34)		
			CAMPSCL U3E1\G10000,U3E1\G10002, U3E1\G10014L (Note-30), (Note-34)	327.0	

(Note-30): The cam position calculation type is set to the cam axis current value per cycle calculation.

(Note-31): The cam data is in the stroke ratio data format, the cam resolution is 256, and the calculation is performed with the midpoint (128).

(Note-32): The cam data is in the stroke ratio data format, the cam resolution is 8192, and the calculation is performed with the midpoint (4096).

(Note-33): The cam data is in the coordinate data format, the coordinates number is 256, and the calculation is performed with the midpoint (128).

(Note-34): The cam data is in the coordinate data format, the coordinates number is 8192, and the calculation is performed with the midpoint (4096).

(2) Transition conditional expressions

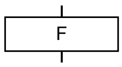
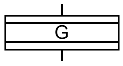
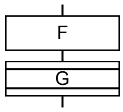
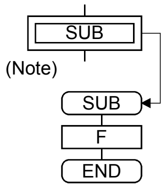
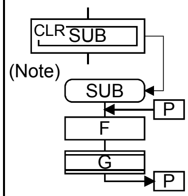
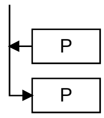
Processing time of transition conditional expressions

Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Bit device status	(None)	ON (Normally open contact) (Completion of condition)	M0	1.0	1.0
			X100		
			PX0	3.0	3.0
			U3E1\G10000.0	0.5	1.5
	!	OFF (Normally closed contact) (Completion of condition)	!M0	1.0	1.5
			!X100		
			!PX0	3.0	3.5
			!U3E1\G10000.0	0.5	1.5
Logical operation	*	Logical AND	M0*M1	1.5	2.0
			X100*X101		
			PX0*PX1	5.0	5.5
			U3E1\G10000.0*U3E1\G10000.1	1.5	2.5
	+	Logical OR	M0+M1	1.5	1.5
			X100+X101		
			PX0+PX1	6.0	7.0
			U3E1\G10000.0+U3E1\G10000.1	1.5	2.5
Comparison operation	==	Equal to (Completion of condition)	#0==#1	1.0	1.5
			D800==D801		
			U3E1\G10000==U3E1\G10001	1.5	2.5
			#0L==#2L	1.5	2.0
			D800L==D802L		
			U3E1\G10000L==U3E1\G10002L	2.0	2.5
			#0F==#4F	2.0	2.5
			D800F==D804F		
			U3E1\G10000F==U3E1\G10004F	3.0	4.0
	!=	Not equal to (Completion of condition)	#0!=#1	1.5	1.5
			D800!=D801		
			U3E1\G10000!=U3E1\G10001	2.0	2.5
			#0L!=#2L	1.5	1.5
			D800L!=D802L		
			U3E1\G10000L!=U3E1\G10002L	1.5	2.5
			#0F!=#4F	2.0	2.0
			D800F!=D804F		
			U3E1\G10000F!=U3E1\G10004F	3.0	3.5
	<	Less than (Completion of condition)	#0<#1	1.5	1.5
			D800<D801		
			U3E1\G10000<U3E1\G10001	2.5	2.5
			#0L<#2L	2.0	2.0
			D800L<D802L		
			U3E1\G10000L<U3E1\G10002L	2.5	2.5
			#0F<#4F	2.0	2.0
			D800F<D804F		
			U3E1\G10000F<U3E1\G10004F	3.0	3.5

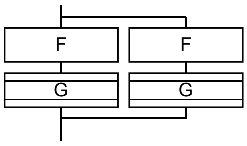
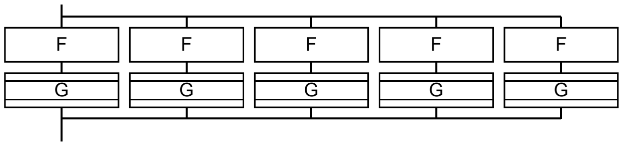
Processing time of transition conditional expressions (Continued)

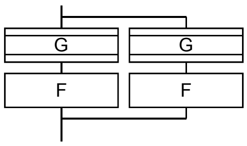
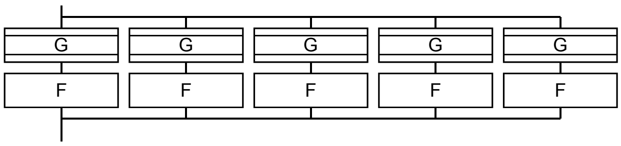
Classifi-cations	Symbol	Instruction	Operation expression	Q173DSCPU/ Q172DSCPU Unit [μs]	Q173DCPU(-S1)/ Q172DCPU(-S1) Unit [μs]
Comparison operation	<=	Less than or equal to (Completion of condition)	#0<=#1	1.5	1.5
			D800<=D801		
			U3E1\G10000<=U3E1\G10001	2.5	2.5
			#0L<=#2L	1.5	1.5
			D800L<=D802L		
			U3E1\G10000L<=U3E1\G10002L	2.0	2.5
			#0F<=#4F	2.5	2.5
			D800<=D804F		
			U3E1\G10000F<=U3E1\G10004F	3.0	3.5
	>	More than (Completion of condition)	#0>#1	1.5	1.5
			D800>D801		
			U3E1\G10000>U3E1\G10001	2.0	2.5
			#0L>#2L	1.5	1.5
			D800L>D802L		
			U3E1\G10000L>U3E1\G10002L	2.0	2.5
			#0F>#4F	2.0	2.0
			D800F>D804F		
			U3E1\G10000F>U3E1\G10004F	3.0	3.5
	>=	More than or equal to (Completion of condition)	#0>=#1	2.0	2.0
			D800>=D801		
			U3E1\G10000>=U3E1\G10001	2.5	2.5
			#0L>=#2L	1.5	1.5
			D800L>=D802L		
			U3E1\G10000L>=U3E1\G10002L	2.5	2.5
			#0F>=#4F	2.0	2.0
			D800F>=D804F		
			U3E1\G10000F>=U3E1\G10004F	3.0	4.0

(3) Processing time by the combination F and G (program described in F/G is NOP)

	F alone	G alone	F+G	GSUB	CLR	JMP/coupling
						
Q173DSCPU Q172DSCPU [μs]	9.0	9.5	10.5	17.0	9.0	4.5
Q173DCPU(-S1) Q172DCPU(-S1)[μs]	14.0	13.5	15.5	22.0	14.5	4.5

(Note): Varies greatly with the started or cleared program.

	Parallel branch (2 Pcs.)		Parallel branch (5 Pcs.)	
				
	At branch	At coupling	At branch	At coupling
Q173DSCPU Q172DSCPU [μs]	16.5	15.0	22.0	22.0
Q173DCPU(-S1) Q172DCPU(-S1)[μs]	23.0	18.5	49.0	32.5

	Selective branch (2 Pcs.)	Selective branch (5 Pcs.)
		
Q173DSCPU Q172DSCPU [μs]	33.5	37.5
Q173DCPU(-S1) Q172DCPU(-S1)[μs]	48.0	55.0

POINT

Long processing time may cause a Motion CPU WDT error or servo fault. Especially for the Motion SFC programs run by event/NMI tasks, take care so that the processing time will not be too long (the processing time will not exceed the operation cycle).

APPENDIX 1.2 Processing time of Motion dedicated PLC instruction

Processing time of Motion dedicated PLC instruction

Classifications	Symbol	Instruction (Condition)		Processing time [μs]					
				Q03UD(E)CPU (Note)		Q04UD(E)HCPU (Note)/ Q06UD(E)HCPU (Note)/ Q10UD(E)HCPU (Note)/ Q13UD(E)HCPU (Note)/ Q20UD(E)HCPU (Note)/ Q26UD(E)HCPU (Note)/ Q50UDEHCPU (Note)/ Q100UDEHCPU (Note)		Q03UDVCPU/ Q04UDVCPU/ Q06UDVCPU/ Q13UDVCPU/ Q26UDVCPU	
				Min.	Max.	Min.	Max.	Min.	Max.
Multiple CPU high speed bus Motion dedicated instruction	D.SFCS	Start request of the specified Motion SFC program		62.0	95.0	60.0	94.0	31.0	66.0
	D.SVST	Start request of the specified servo program		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGA	Current value change request of the specified axis		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGAS	Current value change request of the specified command generation axis		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGV	Speed change request of the specified axis		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGVS	Speed change request of the specified command generation axis		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGT	Torque control value change request of the specified axis		82.0	122.0	80.0	115.0	42.0	76.0
	D.CHGT2	Torque control value individual change request of the specified axis		87.0	127.0	85.0	120.0	42.0	76.0
Multiple CPU high speed bus other CPU access instruction	D.DDWR	Write device data of the self CPU to the device of other CPU	Number of writing data = 1	82.0	133.0	80.0	130.0	34.0	82.0
			Number of writing data = 16	91.0	142.0	89.0	139.0	37.0	84.0
	D.DDRD	Read device data of other CPU to the device of self CPU	Number of reading data = 1	82.0	133.0	80.0	130.0	34.0	81.0
			Number of reading data = 16	82.0	133.0	80.0	130.0	34.0	81.0
	D.GINT	Execute request of an event task of Motion SFC program		50.0	80.0	48.0	78.0	31.0	66.0

(Note): The speed-up of the processing time has been achieved at QnUD(E)(H)CPU-B02 or later (upper five digit of serial No. is "10012" or later).

APPENDIX 2 Sample Program

APPENDIX 2.1 Motion control example by Motion SFC program

(1) The Motion SFC program composition example to execute motion control.
This sample program example using Q173DCPU is described to every following function.

Function list of sample program

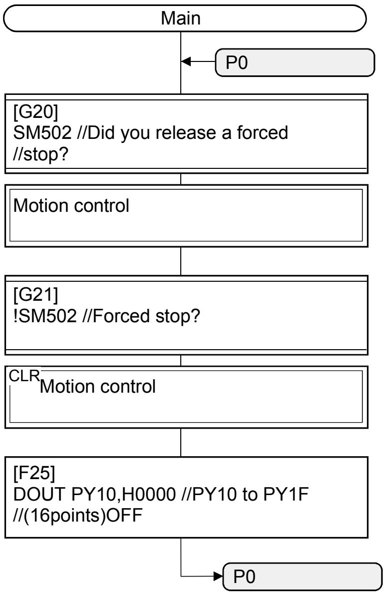
No.	Item	Description
1	Forced stop	When the forced stop input assigned to PX0 is on, all axes turn on, and motion control is executed. When the forced stop input turn off, servo amplifier is made to forced stop, and motion control is suspended, and actual output (PY) turn off.
2	Motion control	Motion control is executed according to the condition of PX and PX2 in each following mode. • PX2 : OFF PX1 : OFF JOG mode • PX2 : OFF PX1 : ON Manual pulse generator mode • PX2 : On PX1 : OFF Home position return mode • PX2 : On PX1 : On Programming operation mode
3	JOG mode	The following JOG operation is executed when each signal of PX3 to PX6 is turned on. • PX3 : 1 axis JOG forward rotation • PX4 : 1 axis JOG reverse rotation • PX5 : 2 axes JOG forward rotation • PX6 : 2 axes JOG reverse rotation
4	Manual pulse generator mode	The following the manual pulse generator operation is executed. • Manual pulse generator operation of 1 axis is executed with the manual pulse generator P1. • Manual pulse generator operation of 2 axes is executed with the manual pulse generator P1.
5	Home position return mode	The following home position return is executed. • When PX3 is on, the home position return of 1 axis is executed. • When PX4 is on, the home position return of 2 axes is executed.
6	Programming operation mode	The following program operation is executed. • When PX3 detects OFF to ON, axis No.1 locates and 1000[ms] standing by, after the location of axis No.2 is executed. • When PX4 turn on, axis No.1, 2 locates of the linear control and in-position check is executed, after the location of axis No.2 is executed, the program stands by until No.1, 2 locates of the linear control is executed at a double speed in the opposition direction and PX4 turns off.

(2) Contents processing of the Motion SFC program

Motion SFC program list

No.	Program name	Task	Automatic operation	Number of connective transitions	Contents of processing
20	Main	Normal	Start	3	• This program starts automatically at the time of run of Motion CPU, and it is always executed. • When a forced stop is cancelled, a subroutine starts a "No.110 : Motion control". • "No.110 : Motion control" is stopped at the time of the forced stop, and real output (PY) is turned off.
110	Motion control	Normal	Not start	3	• All axes servo on. • The call of the subroutine of the following program is executed by the condition of PX1, PX2. 1) PX2 : OFF PX1 : OFF No.120 : JOG 2) PX2 : OFF PX1 : ON No.130 : Manual pulse generator 3) PX2 : ON PX1 : OFF No.140 : Home position return 4) PX2 : ON PX1 : ON No.150 : Programming operation
120	JOG	Normal	Not start	3	• The JOG operation speed of 1 axis and 2 axes is set. • 1 axis JOG forward command is turned on when PX3 is on, and the reverse command is turned on when PX4 is on. • 2 axes JOG forward command is turned on when PX5 is on, and the reverse command is turned on when PX6 is on. • The above are repeated when PX2/PX1 are off, when PX2/PX1 are not off, the JOG forward and reverse command of 1 axis and 2 axes are turned off, and the program is ended.
130	Manual pulse generator	Normal	Not start	3	• 1 pulse input magnification of the 1 axis and 2 axes is set up. • 1 axis is controlled with P1, and set up to control 2 axes with P2, and Manual pulse generator enable flag of P1, P2 is turned on. • When except for PX2 : OFF, PX1 : ON (Manual pulse generator mode), Manual pulse generator enable flag of P1, P2 is turned off, and a program is ended.
140	Home position return	Normal	Not start	3	• "K140 : The home position return of 1 axis" is started when PX3 is on,"K141 : The home position return of 2 axes" is started when PX4 is on. • PX2 : ON, PX1 : The program is ended when they become to except for off (Home position return mode).
150	Programming operation	Normal	Not start	3	• When PX3 detects OFF to ON, after positioning of 1 axis, standing by for 1000[ms] and positioning of 2 axes is executed. • When PX4 turn on, after positioning of linear interpolation in-position check is executed, positioning of axis No. 1, 2 linear interpolation is executed at a double speed in the opposition direction, and it stand by until PX4 turned off. • PX2 : ON, PX1 : The program is fended when they become to except for ON (Programming operation mode).

(a) No.20 : Main



When a forced stop is released, a subroutine starts "No.110 : Motion control". (Because the next step is a shift, it becomes a subroutine start, and the next step is executed at the same time with subroutine practice, too.)

"No.110 : Motion control" is made to stop at the time of the forced stop, and (The program that a subroutine call is executed from No.110 stops, too.) actual output (PY) is turned off.

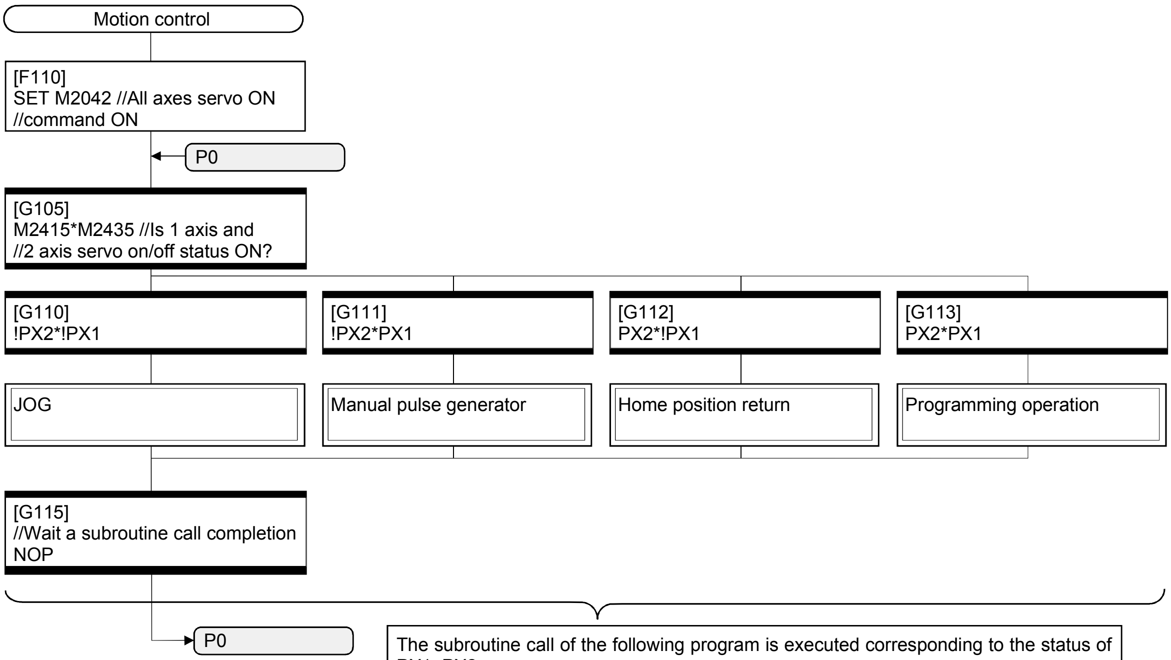
(Note-1) : The program that a subroutine was started is made to stop if necessary when a subroutine start program is added because it does not stop.

(Note-2) : Real output is turned off if necessary.

(Note-3) : The occurrence detection of servo error and so on is added to the stop status with forced stop if necessary.

When a forced stop is released, it is the structure which starts the program which does motion control from the initials again by sample program. Therefore it is the system example that motion control is resumed when a forced stop release is executed after it stops forced for while.

(b) No.110 : Motion control



The subroutine call of the following program is executed corresponding to the status of PX1, PX2.

Selective branch is made the program so that each subroutine may never start it in the plural at the same time.

And, each subroutine makes the next step "WAIT" to become a subroutine call to make it stop when this program is suspended by the clear step of "No.20 : Main", too.

Condition of PX1, PX2		Subroutine call program	
PX2	PX1	No.	Program name
OFF	OFF	120	JOG
OFF	ON	130	Manual pulse generator
ON	OFF	140	Home position return
ON	ON	150	Programming operation

(c) No.120 : JOG

JOG

```
[F120]
//1 axis JOG operation speed =
//100000pulse/s
D640L=K100000
//2 axes JOG operation speed =
//100000pulse/s
D642L=K100000
```

```
[G120]
//1 axis forward rotation JOG start
//SET/RST
SET M3202=PX3 * !M3203
RST M3202=!PX3
//1 axis reverse rotation JOG start
//SET/RST
SET M3203=PX4 * !M3202
RST M3203=!PX4
//2 axis forward rotation JOG start
//SET/RST
SET M3222=PX5 * !M3223
RST M3222=!PX5
//2 axes reverse rotation JOG start
//SET/RST
SET M3223=PX6 * !M3222
RST M3223=!PX6

//Is repeated to the JOG mode
//completion.
!(!PX1*!PX2)
```

When each signal of PX3 to PX6 is turned on/off, which the correspondences JOG command device is SET/RST.
It makes forward rotation JOG start of the same axis and a reverse rotation JOG start from making turned on at the same time.

Signal name	Correspond with JOG command device
PX3	M3202(1 axis forward rotation JOG)
PX4	M3203(1 axis reverse rotation JOG)
PX5	M3222(2 axis forward rotation JOG)
PX6	M3223(2 axis reverse rotation JOG)

(Note) : The ON/OFF distinction of each signal can be described with Y/N transition. But, processing time can be shortened more the number of steps when it was described as the following in the case of the processing which could be described only with SET=/RST= because it is made low.

```
[F122]
//1, 2 axis forward/reverse rotation
//JOG start command is reset.
RST M3202
RST M3203
RST M3222
RST M3223
```

Forward rotation/reverse rotation JOG status of 1, 2 axis is turned off at the time of the JOG mode completion not to continue a JOG movement after it moves to other mode of the safety.

END

(d) No.130 : Manual pulse generator

Manual pulse generator

```
[F130]
D720=100 //1-pulse input magnification
//setting of 1 axis
D721=100 //1-pulse input magnification
//setting of 2 axes
D714L=H00000001 //P1 is controlled
//1 axis.
D716L=H00000002 //P2 is controlled
//2 axes
SET M2051 //P1 Manual pulse generator
//enable flag is ON
SET M2052 //P2 Manual pulse generator
//enable flag is ON
```

The setup of the following is executed to do manual pulse generator operation of P1 with 1 axis/P2 with 2 axis.

- Setting of 1-pulse input magnification of the 1 axis and 2 axis.
- Manual pulse generator axis No. setting register is setup to control of P1 with 1 axis/P2 with 2 axis.
- Manual pulse generator axis enable flag of P1, P2 is turned on.

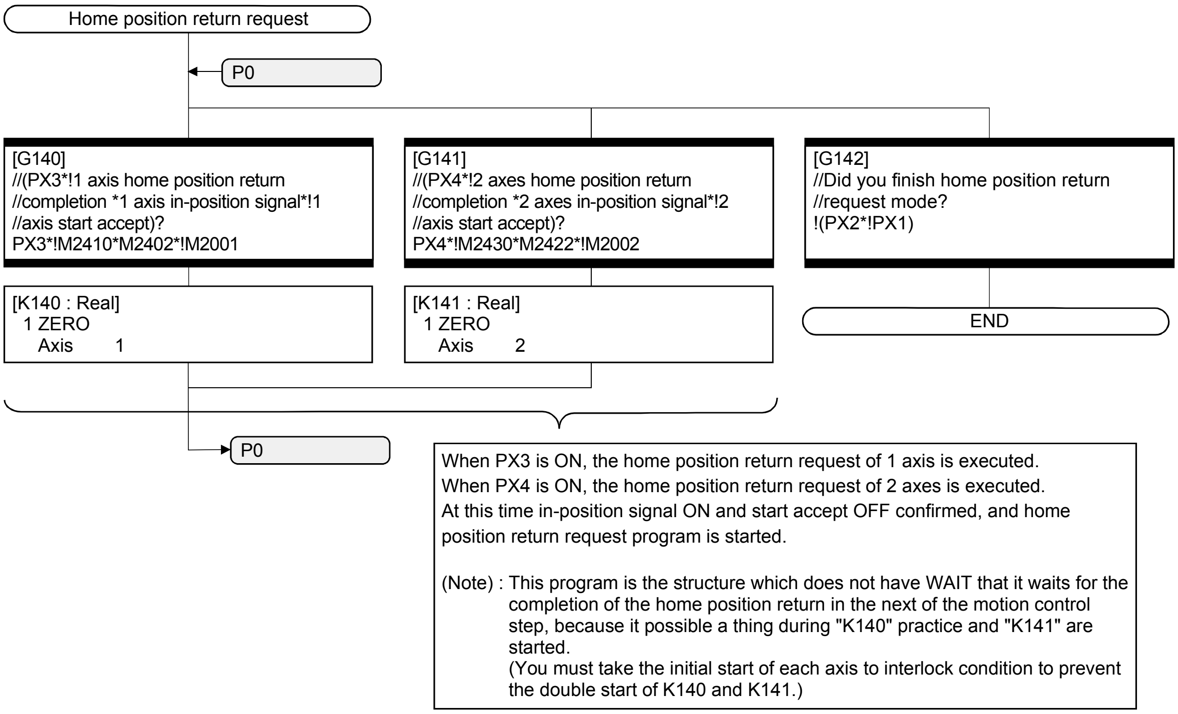
```
[G130]
!(!PX2*PX1)//Did you complete a manual
//pulse generator mode?
```

```
[F131]
RST M2051 //P1 Manual pulse generator
//enable flag is OFF
RST M2052 //P2 Manual pulse generator
//enable flag is OFF
```

1, 2 axis Manual pulse generator enable flag turned off at the time of the JOG mode completion not to continue a manual pulse generator operation after it moves to other mode of the safety.

END

(e) No.140 : Home position return



(f) No.150 : Programming operation

Programming operation

P0

```
[G150]
//(OFF to ON)detection of PX3.

//PX3 turns on M0 in on when M1 (last time
//condition of PX3) is off.
RST M0
SET M0=PX3 * !M1
//Condition was stored in M1 last time
//of PX3.
RST M1
SET M1=PX3
//When M0 is ON(OFF to ON of PX3 is
//detected.), it transition to the next steps.
M0
```

Detection for leading edge of bit device (PX3). Condition was memorized in M1 last time, and OFF to ON of PX3 was detected.

```
[K150:Real]
 1 INC-1
   Axis        1,   1000000 pulse
   Speed            500000 pulse/s
```

```
[G153]
//It waits 1000[ms] after the motion control
//completion.TIME K1000
```

```
[K152:Real]
 1 INC-1
   Axis       2,   2000000 pulse
   Speed           1000000 pulse/s
```

```
[G155]
NOP //Waits for the motion control
//completion.
```

When OFF to ON of PX3 is detected, 1000[ms] after 1 axis positioning completion stands by, and the positioning of 2 axes is executed, and stands by to the positioning completion of 2 axes.

P0

```
[G151]
//Did you turn on PX4?
PX4
```

```
[F150]
#0L=1000000//1 axis positioning address
//set
#2L=2000000//2 axes positioning address
//set
#4L=500000//Positioning speed
```

```
[K151:Real]
1 INC-2
  Axis              1,  #  0 pulse
  Axis              2,  #  2 pulse
  Vector speed          #  4 pulse/s
```

Positioning address, the indirect designation of the speed.

```
[G154]
//Did you turned on 1 axis and 2 axis
//in-position?
M2402*M2422
```

```
[F151]
#0L=-#0L //1 axis positioning address sign
//inversion
#2L=-#2L //2 axes positioning address sign
//inversion
#4L= #4L*2 //Positioning speed  *2
```

```
[K151:Real]
1 INC-2
  Axis              1,  #  0 pulse
  Axis              2,  #  2 pulse
  Vector speed          #  4 pulse/s
```

Positioning address, the indirect designation of the speed.

```
[G156]
//Did you turn on PX4?
!PX4
```

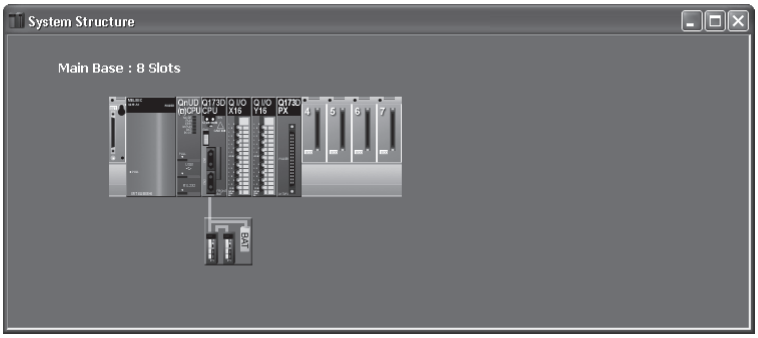
When PX4 is on, the linear interpolation control of the 1 axis and 2 axis is executed at a double speed in the action, the opposition direction after the linear interpolation control of the 1 axis and 2 axis in-position check. And stands by to PX4 is OFF.

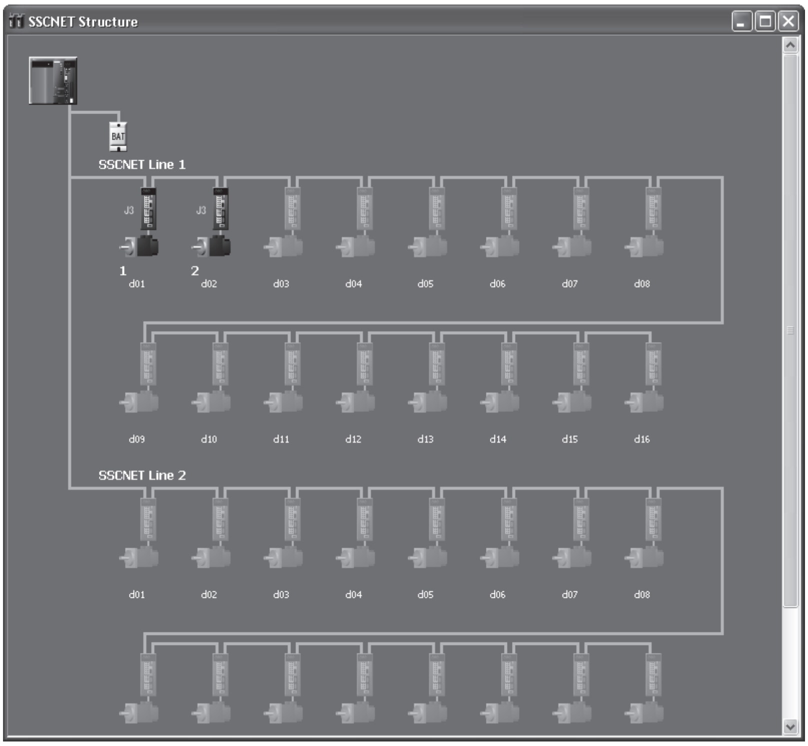
```
[G152]
//Did you finish a programming operation
//mode?
!(PX2*PX1)
```

END

(3) System setting data of the Motion CPU

System setting is shown below.





(a) Module setting

1) Motion module setting

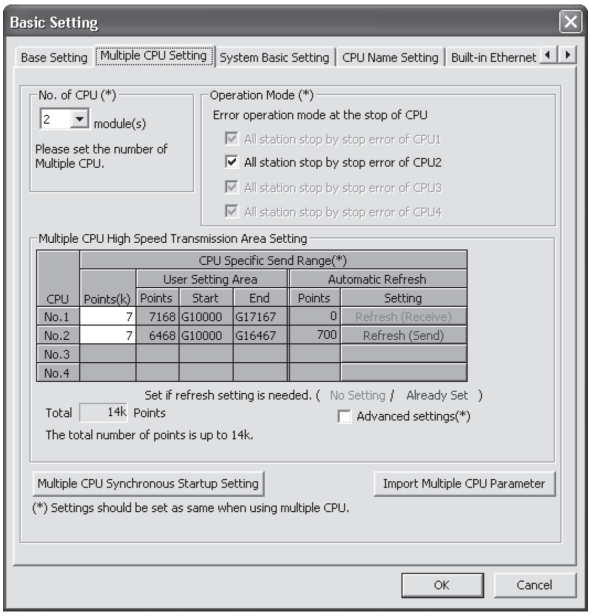
Manual pulse generator interface module (Q173DPX : Slot 3)

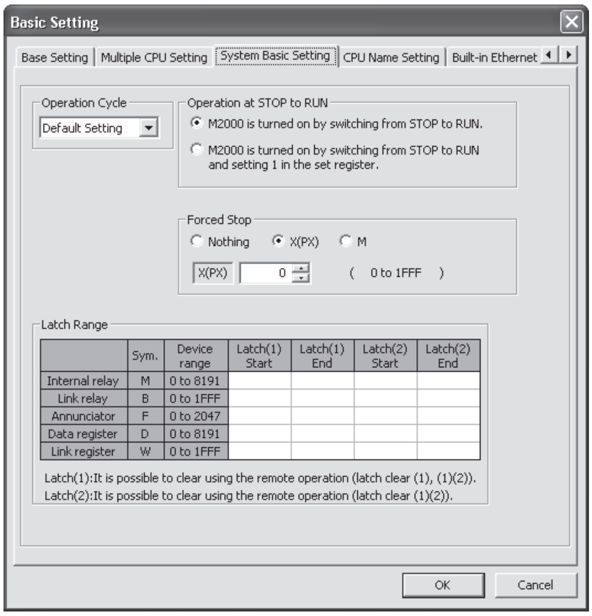
Axis No.	Description
P1	Manual pulse generator, Synchronous encoder (INC)
P2	Manual pulse generator, Synchronous encoder (INC)
I/O response time	0.4[ms]

2) PLC module setting

Module type	Points	Occupied I/O No.	Base	Slot No.	I/O response time
Input	16	000-00F	Main Base	1	10[ms]
Output	16	010-01F	Main Base	2	

(b) Basic setting





1) Multiple CPU setting

Setting items	Description
No. of CPU	2 modules
Operating mode	All station stop by stop error of CPU 1/2
Multiple CPU synchronous startup setting	Set CPU No. 1/2 to synchronous startup

2) Multiple CPU high speed transmission area setting

CPU	Points (k)	CPU specific send range			
		User setting area			Automatic refresh
		Points	Start	End	Points
No.1	7	7168	G10000	G17167	0
No.2	7	6468	G10000	G16467	700
No.3					
No.4					

3) Automatic refresh setting

a) CPU No.1

Setting No.	Automatic refresh		
	Points	Start	End
1			
2			
3			

b) CPU No.2

Setting No.	Automatic refresh		
	Points	Start	End
1	20	M2000	M2319
2	40	M2400	M3039
3	640	D0	D639

4) System basic setting

Setting items	Description
Operation cycle	Default Setting
Operation at STOP to RUN	M2000 is turned on by switching from STOP to RUN
Forced stop	PX0

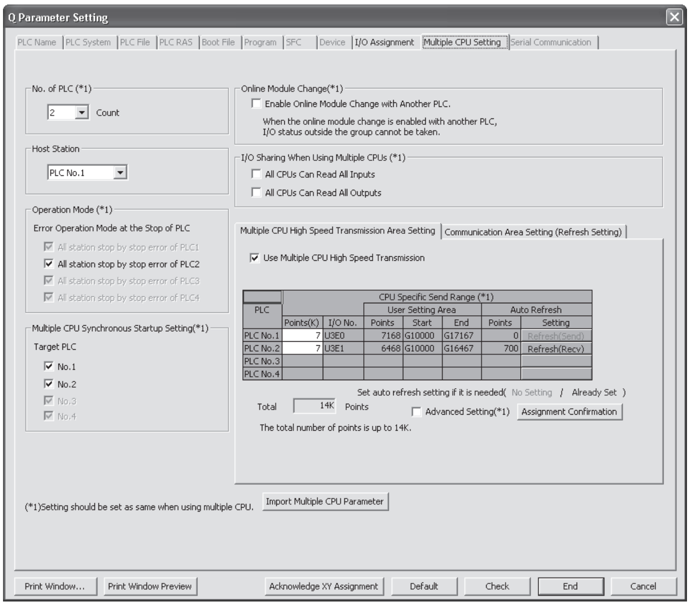
5) Latch range setting

Item	Symbol	Latch (1)		Latch (2)	
		Start	End	Start	End
Internal relay	M				
Link relay	B				
Annunciator	F				
Data register	D				
Link register	W				

Latch (1) : It is possible to clear using remote operation (latch clear (1), latch clear (1) (2)).

Latch (2) : It is possible to clear using remote operation (latch clear (1) (2)).

(4) Parameter setting of the PLC CPU (No.1)



<Screen: GX Works2>

<table>
<tr><th colspan="2">PC parameter item</th><th>Description</th></tr>
<tr><td>1</td><td>No. of PLC</td><td>2 modules</td></tr>
<tr><td>2</td><td>Operating mode</td><td>All station stop by stop error of PLC1/PLC2</td></tr>
<tr><td>3</td><td>Multiple CPU synchronous startup</td><td>Check the PLC No.1/PLC No.2</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">I/O sharing when using Multiple CPUs</td><td>Check the all CPUs can read all inputs</td></tr>
<tr><td>Not check the all CPUs can read all outputs</td></tr>
<tr><td>5</td><td>Multiple CPU high speed communication area setting</td><td>Use multiple CPU high speed communication
<table>
<tr><th rowspan="3">PLC</th><th colspan="8">CPU specific send range</th></tr>
<tr><th colspan="2"></th><th colspan="3">User setting area</th><th colspan="3">Auto refresh</th></tr>
<tr><th>point (K)</th><th>I/O No.</th><th>point</th><th>Start</th><th>End</th><th>point</th><th>Start</th><th>End</th></tr>
<tr><td>CPU No.1</td><td>7</td><td>U3E0</td><td>7168</td><td>G10000</td><td>G17167</td><td>0</td><td>—</td><td>—</td></tr>
<tr><td>CPU No.2</td><td>7</td><td>U3E1</td><td>6468</td><td>G10000</td><td>G16467</td><td>700</td><td>G16468</td><td>G17167</td></tr>
</table></td></tr>
<tr><td>6</td><td>Auto refresh setting</td><td>• PLC No1
<table>
<tr><th rowspan="2">No.</th><th colspan="3">Auto refresh</th><th colspan="2">CPU specific send range</th></tr>
<tr><th>point</th><th>Start</th><th>End</th><th>Start</th><th>End</th></tr>
<tr><td>1</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>2</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>3</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
</table>
• PLC No.2
<table>
<tr><th rowspan="2">No.</th><th colspan="3">Auto refresh</th><th colspan="2">CPU specific send range</th></tr>
<tr><th>point</th><th>Start</th><th>End</th><th>Start</th><th>End</th></tr>
<tr><td>1</td><td>20</td><td>M2000</td><td>M2319</td><td>G16468</td><td>G16487</td></tr>
<tr><td>2</td><td>40</td><td>M2400</td><td>M3039</td><td>G16488</td><td>G16527</td></tr>
<tr><td>3</td><td>640</td><td>D0</td><td>D639</td><td>G16528</td><td>G17167</td></tr>
</table></td></tr>
</table>

APPENDIX 2.2 Continuation execution example at the subroutine re-start by the Motion SFC program

(1) Explanation of the operation

This is the program example which execute continuously from the motion control step which stopped on the way when it re-started after stopping the subroutine program with the clear step during the motion control is running.

The servo is turned on by the forced stop release and the positioning control of the 2 axes linear interpolation is executed when PX4 is ON in this program. One cycle operation is completed after confirmation that PX4 became OFF. When the forced stop is executed during the positioning operating, the positioning operation is interrupted and the servomotor is stopped. It is resumed from the interrupted positioning operation when the forced stop was released next.
Continuation execution of the subroutine re-start is executed by this program example by the following processing.

(a) While motion control with the subroutine is executed, it is memorized whether the positioning of which motion control step was completed in the user device.

(b) The subroutine re-start is resumed from the motion control step of stopping the information memorized by the above (a).

(c) A motion control step should locate absolute to cope with it when it is resumed after it stops on the way of the positioning.

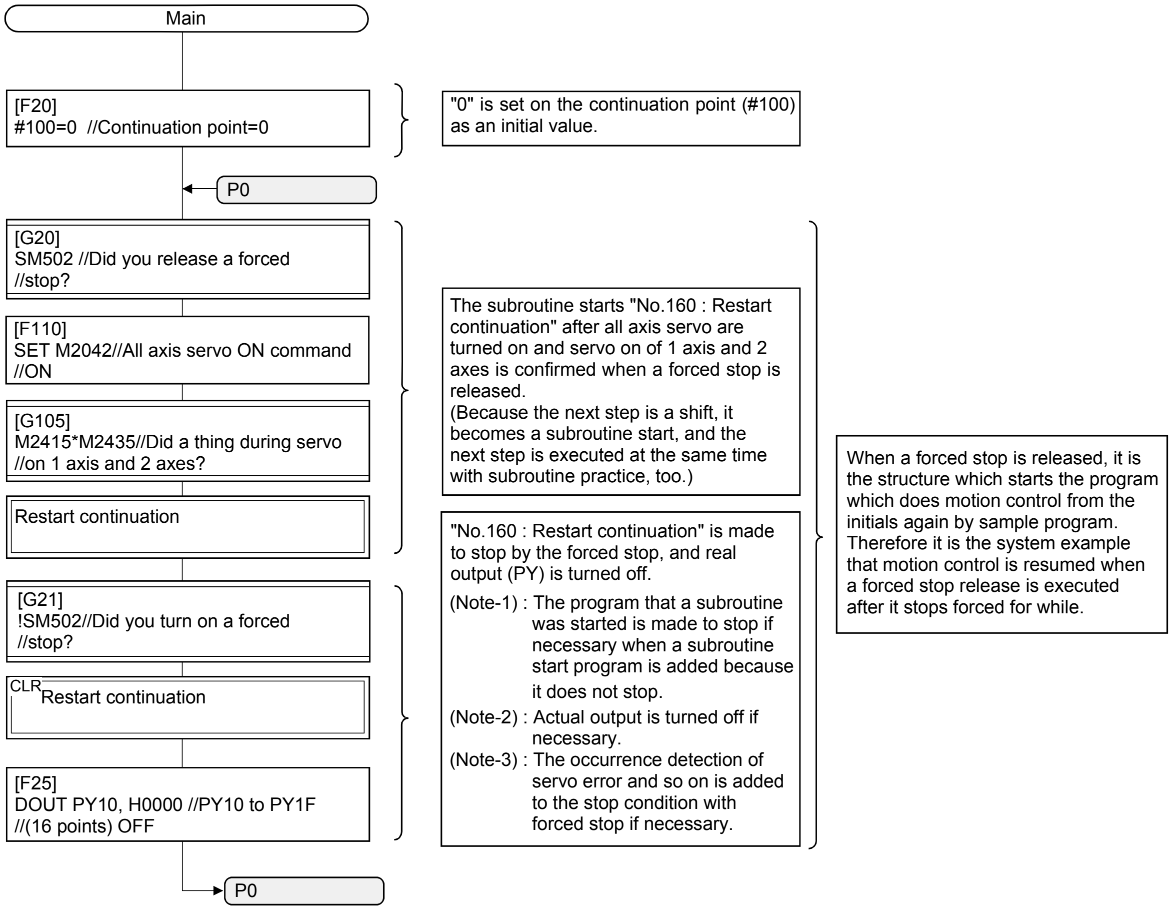
(d) A positioning complete signal (M2401+20n) is used for the decision, whether servomotor is stopped during the positioning.

(2) Contents of processing the Motion SFC program

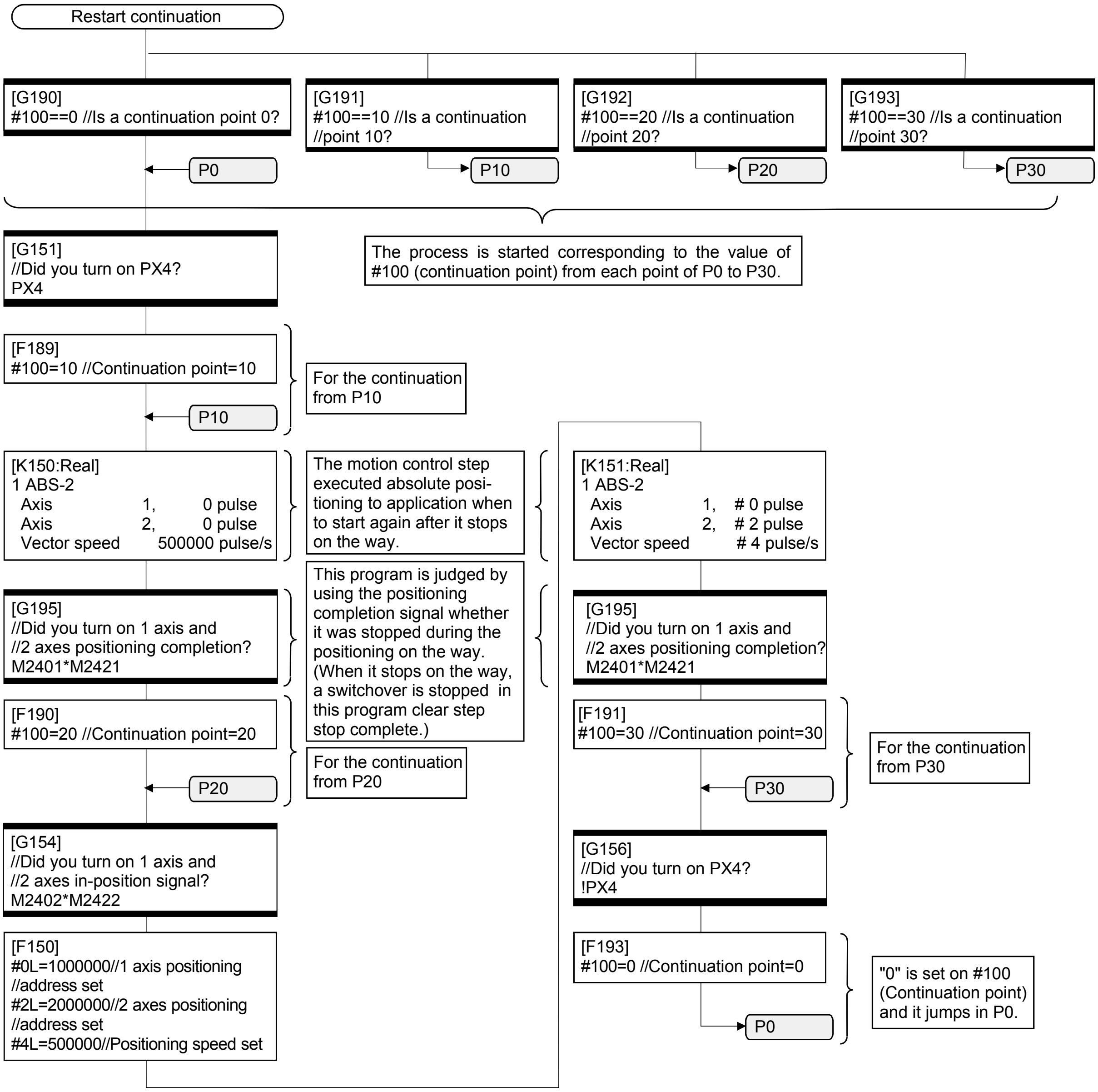
Motion SFC program list

<table>
<tr><th>No.</th><th>Program name</th><th>Task</th><th>Automatic operation</th><th>Number of connective transitions</th><th>Contents of processing</th></tr>
<tr><td>20</td><td>Main</td><td>Normal</td><td>Start</td><td>3</td><td>• This program starts automatically at the time of RUN of Motion CPU, and it is always executed.
• "0" is set on the continuation point (#100 : user device) as an initial value.
• The subroutine starts a "No.160 : Re-start continuation" after all axes servo are turned on and servo on of 1 axis and 2 axes is confirmed when a forced stop is released.
• "No.160 : Re-start continuation" is stopped at the time of the forced stop, and actual output (PY) is turned off.</td></tr>
<tr><td>160</td><td>Restart continuation</td><td>Normal</td><td>Not start</td><td>3</td><td>(1) This program jumps corresponding to the value of the continuation point (#100) of the following (2) 1) to 9).
<table>
<tr><th>#100</th><th>Jump destination</th></tr>
<tr><td>0</td><td>Following (2) 1)</td></tr>
<tr><td>10</td><td>Following (2) 3)</td></tr>
<tr><td>20</td><td>Following (2) 5)</td></tr>
<tr><td>30</td><td>Following (2) 8)</td></tr>
</table>
(2) The following motion control is executed.
1) This program stands by until PX4 is turned on.
2) "10" is set on continuation point (#100).
3) 1 axis, 2 axes are located in (0,0) in the linear control (absolute 2 axes positioning).
4) Positioning completion signal on of 1 axis, 2 axes is confirmed, and "20" is set on the continuation point (#100).
5) In-position on of 1 axis and 2 axes is confirmed.
6) 1 axis, 2 axes are located in (1000000, 2000000) in the linear control (absolute 2 axes positioning).
7) Positioning completion signal on of 1 axis, 2 axes is confirmed, and "30" is set on the continuation point (#100).
8) This program stands by until PX4 is turned off.
9) "0" is set on continuation point (#100).</td></tr>
</table>

(a) No.20 : Main


Main
[F20]
#100=0 //Continuation point=0
"0" is set on the continuation point (#100) as an initial value.
P0
[G20]
SM502 //Did you release a forced //stop?
[F110]
SET M2042//All axis servo ON command //ON
[G105]
M2415*M2435//Did a thing during servo //on 1 axis and 2 axes?
Restart continuation
The subroutine starts "No.160 : Restart continuation" after all axis servo are turned on and servo on of 1 axis and 2 axes is confirmed when a forced stop is released.
(Because the next step is a shift, it becomes a subroutine start, and the next step is executed at the same time with subroutine practice, too.)
[G21]
!SM502//Did you turn on a forced //stop?
CLR Restart continuation
[F25]
DOUT PY10, H0000 //PY10 to PY1F //(16 points) OFF
P0
"No.160 : Restart continuation" is made to stop by the forced stop, and real output (PY) is turned off.
(Note-1) : The program that a subroutine was started is made to stop if necessary when a subroutine start program is added because it does not stop.
(Note-2) : Actual output is turned off if necessary.
(Note-3) : The occurrence detection of servo error and so on is added to the stop condition with forced stop if necessary.
When a forced stop is released, it is the structure which starts the program which does motion control from the initials again by sample program. Therefore it is the system example that motion control is resumed when a forced stop release is executed after it stops forced for while.

(b) No.160 : Restart continuation



APPENDIX 2.3 Continuation execution example after the stop by the Motion SFC program

(1) The explanation of the operation

The program example that the Motion SFC program is stopped by external input signal ON for the forced stop from the input module, and it is executed continuously by external signal OFF for the stop is shown below.

The servo is turned on by the forced stop release and the positioning control of the 2 axes linear interpolation is executed when PX4 is ON in this program. One cycle operation is completed after confirmation that PX4 became OFF.
When PX5 turns ON during the positioning operating, the positioning operation is stopped by the stop instruction and it is resumed from the interrupted positioning operation at turning PX5 on. The transition to the next step is not executed during PX5 is ON in the WAIT transition.
When the forced stop is executed during the positioning operating, the positioning operation is interrupted and the servomotor is stopped. It is resumed from the interrupted positioning operation when the forced stop was released next.
Continuation execution of the stop and stop after is executed by this program example by the following processing.

(a) While PX5 turns it on, it is made to turn on a stop command (M3200+20n) and an internal relay (M100) for the stop.

(b) While PX5 turns it off, it is made to turn off a stop command (M3200+20n) and an internal relay (M100) for the stop.

(c) A motion control step does absolute position to cope with it when it is resumed after it stops on the way of the positioning.

(d) A positioning completion signal (M2401+20n) is used for the decision whether it is stopped during the positioning on the way.

(e) The motion control step is resumed after it waits to turn it off, when it was stepped during positioning.

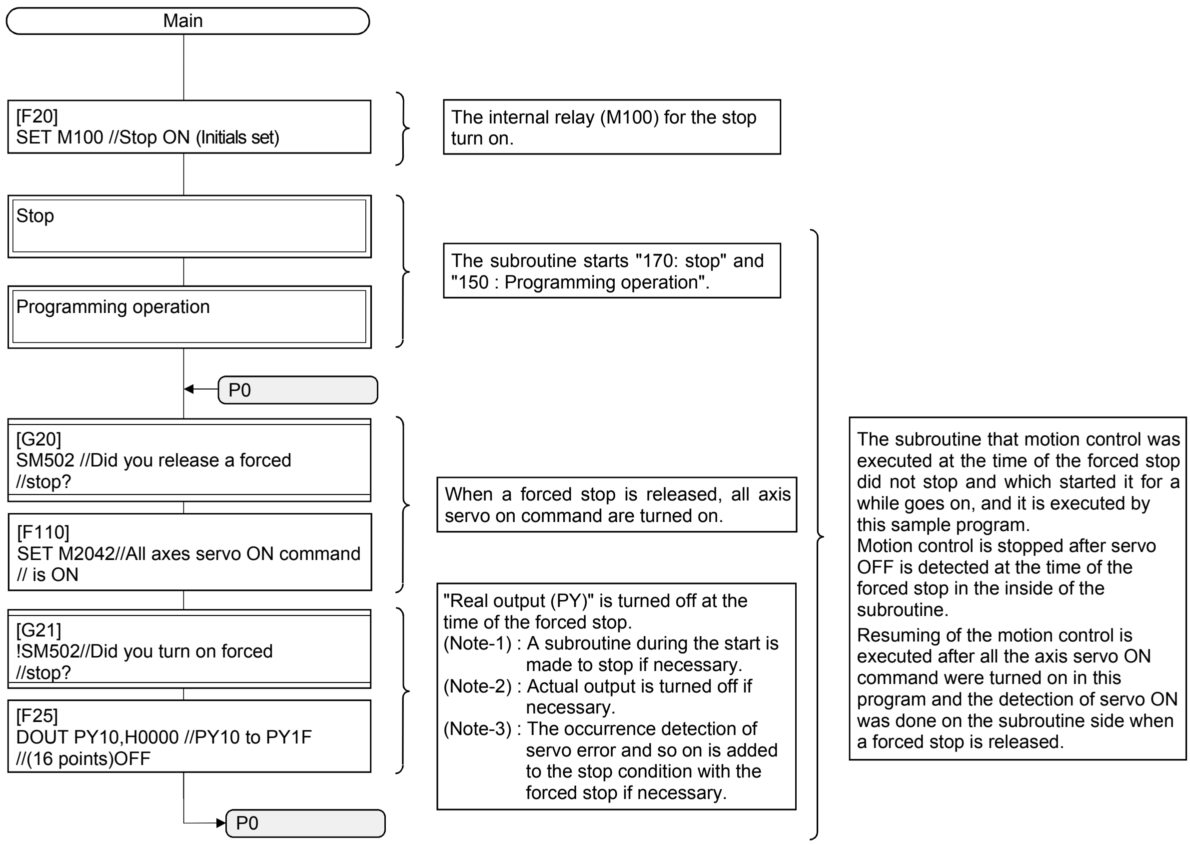
(f) "The internal relay (M100) for the stop turn off." is substituted for the WAIT transition condition that you must stop.

(2) Contents of processing Motion SFC program

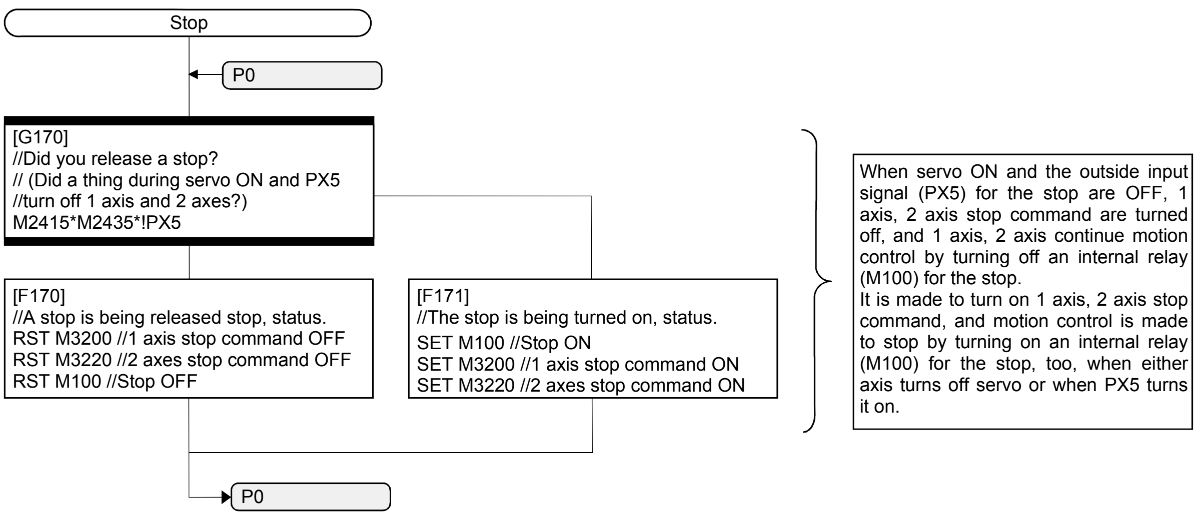
Motion SFC program list

No.	Program name	Task	Automatic operation	Number of connective transitions	Contents of processing
20	Main	Normal	Start	3	• This program starts automatically at the time of RUN of Motion CPU, and it is always executed. • The initials condition of the internal relay (M100) for the stop is turned on. • The subroutine starts "No.170 : Stop". • The subroutine starts "No.150 : Programming operation". • When an forced stop is released, all axes servo are turned on. • Turns off actual output (PY) at the time of the forced stop.
170	Stop	Normal	Not start	3	(1) When a stop input signal (PX5) from the input unit is off, the treatment of the following (2) is executed, and 1 axis and 2 axes executed the following (3) during servo on in the case of the one except for it. (2) 1 axis and 2 axes stop command are turned off, and an internal relay (M100) for the stop is turned off. (3) 1 axis and 2 axes stop command are turned on, and an internal relay (M100) for the stop is turned on.
150	Program operation	Normal	Not start	3	(1) The following motion control is executed. 1) This program stands by until PX4 is turned on. 2) 1 axis and 2 axes are located in (0,0) in the linear interpolation control (absolute 2 axes positioning). 3) Positioning completion signal on of 1 axis and 2 axes are confirmed. 4) In-position on of 1 axis and 2 axes are confirmed. 5) 1 axis and 2 axes are located in (1000000, 2000000) in the linear control (absolute 2 axes positioning). 6) Positioning completion signal on of 1 axis and 2 axes are confirmed. 7) This program stands by until PX4 is turned off. (2) When a positioning completion signal of the above (1) 3) and 6) is off, it waits to turn off, and (When a positioning was suspended on the way.) execute the motion control step (1) 2) or 5) again. (3) Until an internal relay (M100) for the stop turns it on, it does not move to the next step of the above (1) 1) and 7).

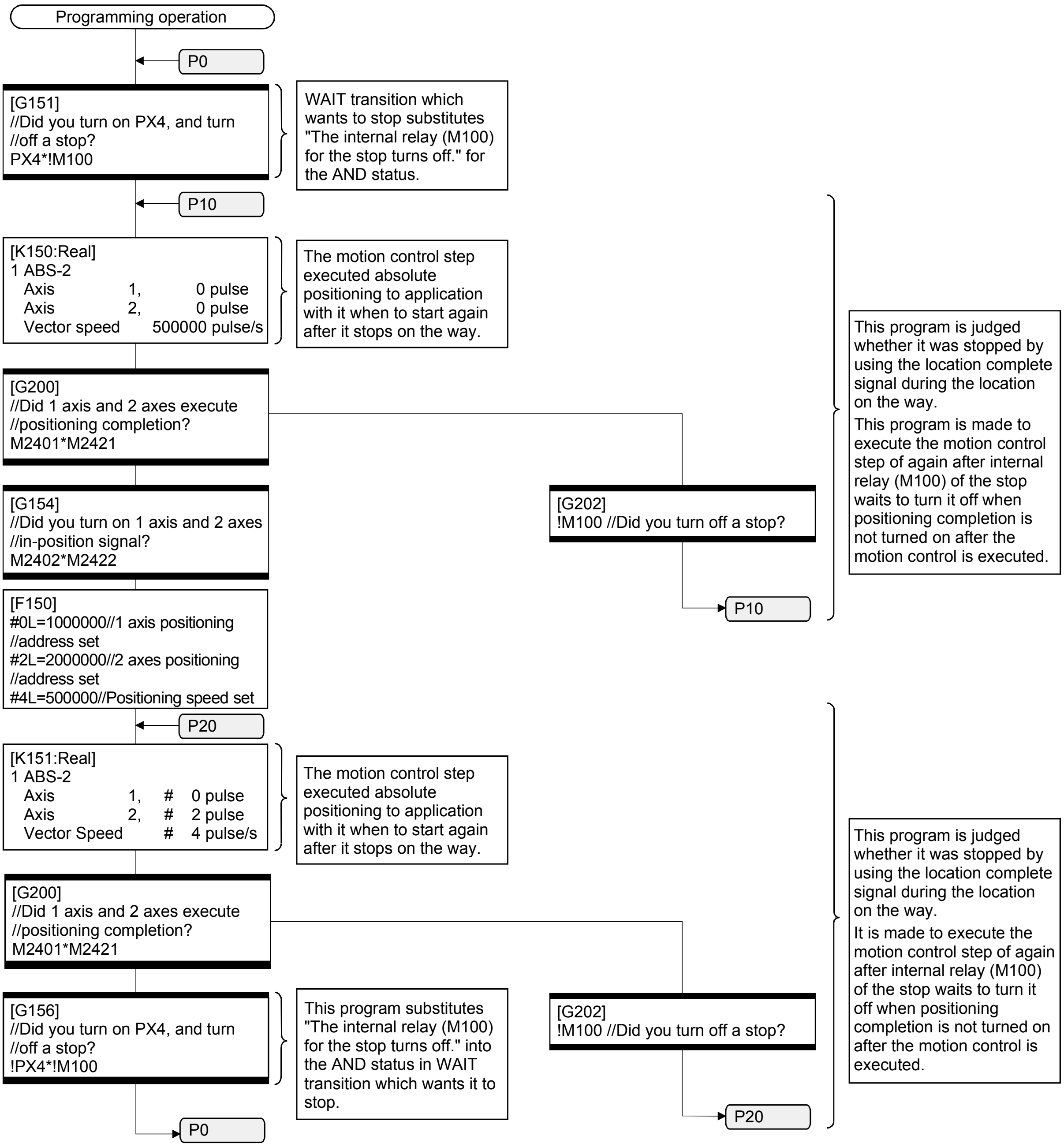
(a) No.20 : Main



(b) No.170 : Stop



(c) No.150 : Programming operation



APPENDIX 3 Vision System Connection Function Ver.!

APPENDIX 3.1 Overview

The Cognex In-Sight® vision system can be connected to the PERIPHERAL I/F of the Motion CPU (Q173DSCPU/Q172DSCPU/Q173DCPU-S1/Q172DCPU-S1).
The vision system dedicated functions have been added to the Motion SFC program making it easy to control the vision system from the Motion SFC program.
There are restrictions to the operating system software and programming software versions when using the Cognex vision system connection function. (Refer to Section 1.3.)
Refer to the manual or help sections provided by Cognex for details on the In-Sight® vision system and Cognex vision system integrating tool In-Sight® Explorer.

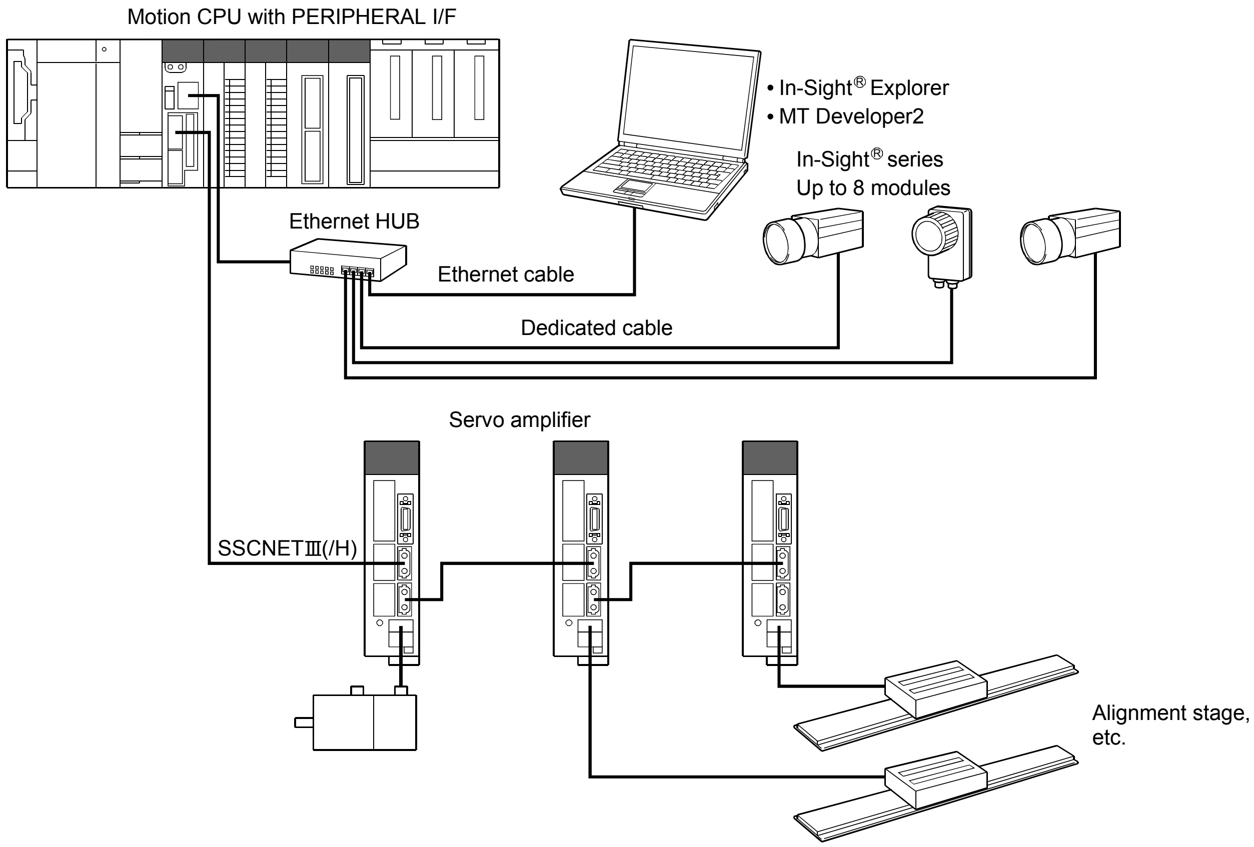
The following terms are used to explain the vision system connection function.

Term	Description
In-Sight® Explorer	Abbreviation for Cognex vision system integrating tool In-Sight® Explorer (Version 4.3.0 or later).
Log on/Log off	Procedure to connect/disconnect the communication to the vision system from the Motion CPU.
PoE	Abbreviation for Power over Ethernet. Method of supplying power via an Ethernet cable.
Native mode	Vision system's communication method used to control the vision system from the Motion CPU.
TCP/IP	One of vision system's communication protocol names.
Job (Vision program)	Program that processes images in the vision system.
Load	The process of developing a job file stored in the vision system into the memory in the vision system, and making it an active job.
Trigger	Start signal for acquiring images.
Vision system status storage device	Device that stores the status of the vision system controlled by the Motion CPU.
Program status storage device	Device that stores the status of jobs controlled by the Motion CPU.
Image data	Various data created by the vision system's image process. (Not images acquired by the trigger.)
Read value	Numeric data retrieved in addition to the image data.
Spreadsheet	A table (400 line × 26 row) in which the job is written. The program is created by writing the functions of various image processes, etc., in each spreadsheet cell.
Tag	Symbolic tag associated with the spreadsheet cells.

Ver.! : Refer to Section 1.3 for the software version that supports this function.

(1) System configuration

This section explains the system configuration and precautions for using the Cognex vision system connection function.



POINT
Depending on the vision system used, power supply by Power over Ethernet (PoE) may be required. The Motion CPU's PERIPHERAL I/F does not support PoE, so a PoE power source equipment (PSE) is required.

(2) Features of vision system connection function

This section explains the features of the Cognex vision system connection function.

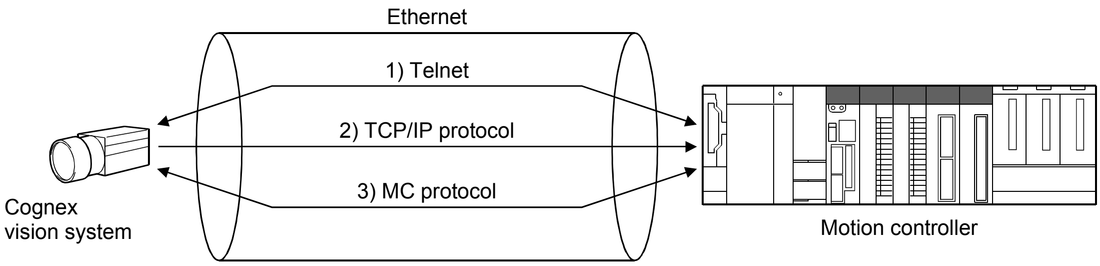
(a) Method of connection with vision system

The Motion CPU and vision system are connected with the Motion CPU's PERIPHERAL IF (Ethernet). A dedicated communication module, etc., is not needed.

POINT

(1) The Motion CPU and vision system are connected via Ethernet. The response of vision system dedicated functions may slow down if several devices (MT Developer2, personal computer for In-Sight® Explorer or GOT, etc.) are connected via an Ethernet HUB.
(2) When simultaneously controlling two or more vision systems, the execution of commands to other vision systems may be delayed during the log on process (MVOPEN) with a specific vision system.
(3) When simultaneously controlling two or more vision systems, if the offline-online state for a specific vision system is switched from an external source, the execution of commands to other vision systems may be delayed.
(4) Execution of the vision system dedicated functions may be delayed if the vision system is in the offline state.
(5) When the vision system is logged onto, communication is established between the Motion CPU and vision system to check the connection state even if the vision system dedicated functions are not used.

The following three communication methods can be used simultaneously with Ethernet.
(Note-1)



(Note-1): The simultaneous communication cannot be used depending on the vision system's model. Refer to the manual or help sections provided by Cognex to confirm the specifications of vision system.

1) Telnet
   The vision system is controlled from the Motion CPU using the native mode. The vision system dedicated functions control the vision system using Telnet.
2) TCP/IP protocol
   By using TCP/IP as the vision system communication protocol and setting the output string, the image data are sent in a batch to the Motion CPU immediately after the job is finished with the vision system.
   High-speed data transfer is possible compared to the other communication methods.
3) MC protocol
   By setting the vision system, data can be easily exchanged between the vision system and Motion CPU device.
   Refer to the manual, etc. provided by Cognex for details on using MC protocol.
   The Motion CPU parameters (built-in Ethernet port open setting) must also be set.

(b) Vision system parameter
Parameters required for Ethernet communication and job execution must be set beforehand with MT Developer2.
The vision system can be controlled just by writing only the vision system dedicated functions in the Motion SFC program.

(c) Priority of the vision system dedicated function

The priority of the vision system dedicated functions in the Motion CPU are shown below.

Process	Description	Priority
Motion operation process	Servo operation process, Servo data communication process, Event task of Motion SFC, etc.	High
Vision system dedicated function	Execution of the communication process with the vision system	↕
Motion main process	Communication process with the peripheral devices, Auto refresh process, Normal task process of Motion SFC	Low

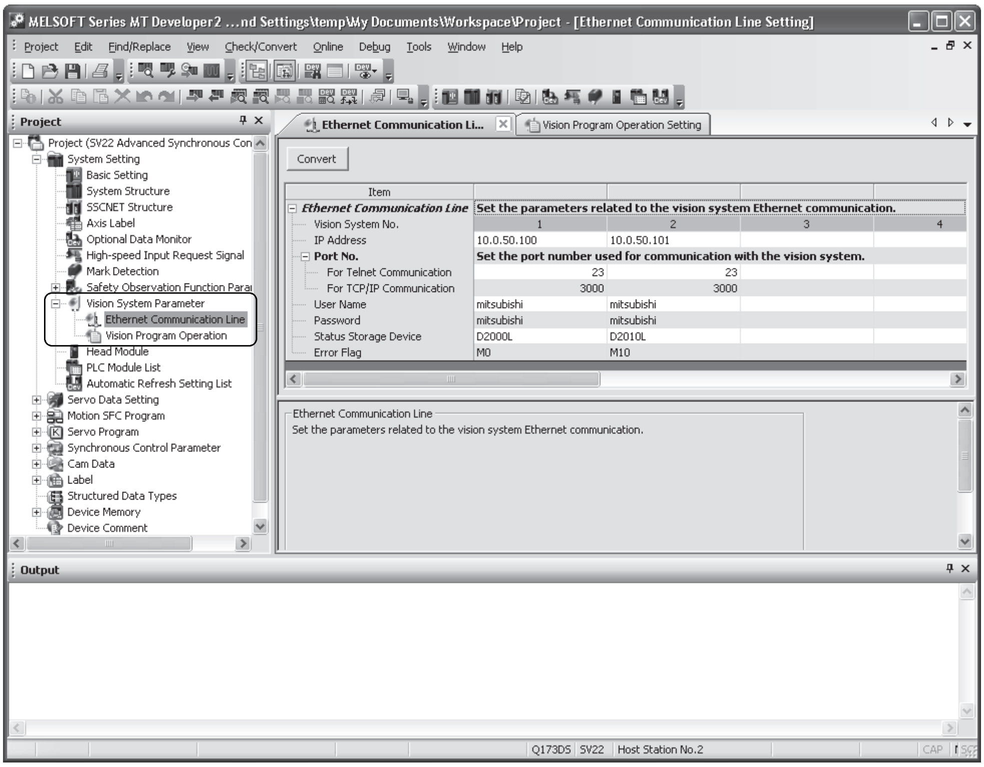
POINT

(1) The communication process with the vision system has a lower priority than the motion operation cycle, so the motion operation processing time is not affected.

(2) Even if the vision system dedicated function is executed from the Motion SFC event task or NMI task, communication process with the vision system is executed after the motion operation process.

APPENDIX 3.2 Vision system parameter setting

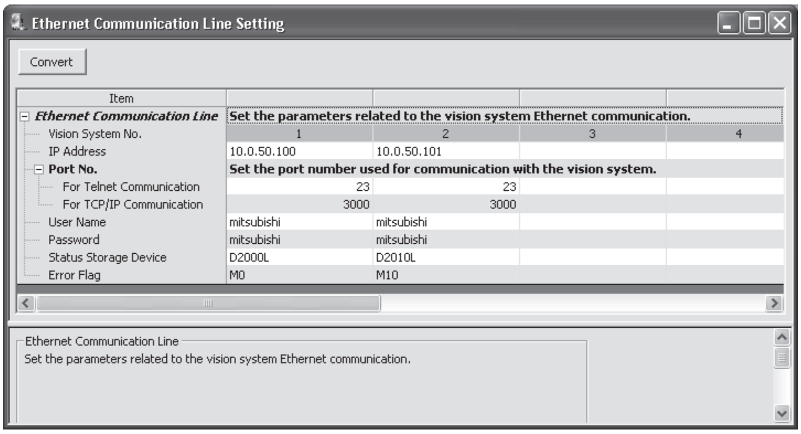
The vision system parameters (Ethernet communication line setting, vision program operation setting) of the MT Developer2 are set by opening an arbitrary SFC program and using the project window [System Setting] - [Vision System Parameter] - [Ethernet Communication Line Setting] or [Vision Program Operation Setting].



POINT
When writing the vision system parameters into the Motion CPU, execute one of the following. • Select the menu bar [Check/Convert] - [Vision System Parameter Check]. • Click [Convert] button of Ethernet communication line setting screen or vision program operation setting screen.

(1) Ethernet Communication Line Setting

Set the parameters related to the vision system Ethernet communication.



(a) Vision System (camera) No. (Not necessary to set)
This number (1 to 8) is used by the vision system dedicated function to identify the vision system.

(b) IP Address
Set the IP address set for each vision system.

(c) Port No.
Set the port number used for communication with the vision system.
Set the same number as the port number set for the vision system with In-Sight® Explorer.

1) For Telnet communication
Set the Telnet connection port number used to control the vision system from the Motion CPU.
If this number is not set, the Telnet default port number (23) will be used.

2) For TCP/IP communication
Set the vision system's TCP/IP server port number used to batch send the vision system job execution results with the format output string setting of TCP/IP protocol.
This does not need to be set when not using the format output string setting of TCP/IP protocol.

(d) User Name
Select a user from those set in the vision system to be used when executing the vision system dedicated functions.

POINT

Select a user name for which the access level is set to "Full Access" or "Protect" in the In-Sight® Explorer user list.

(e) Password
Set the password corresponding to the set user name.

(f) Status Storage Device

Set the word device in which the vision system status and vision system dedicated function error codes are set.

Settable word devices are shown below.

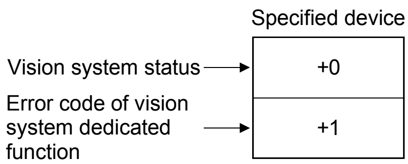
Item	Device No. setting range
Data register	D0 to D8190 (Note-1)
Link register	W0 to W1FFE
Motion register	#0 to #7998

(Note-1): Only the user device range can be set.

POINT

Set the device No. as an even-number.

The vision system status and error code of vision system dedicated function are stored in two successive points of the specified device as shown below.

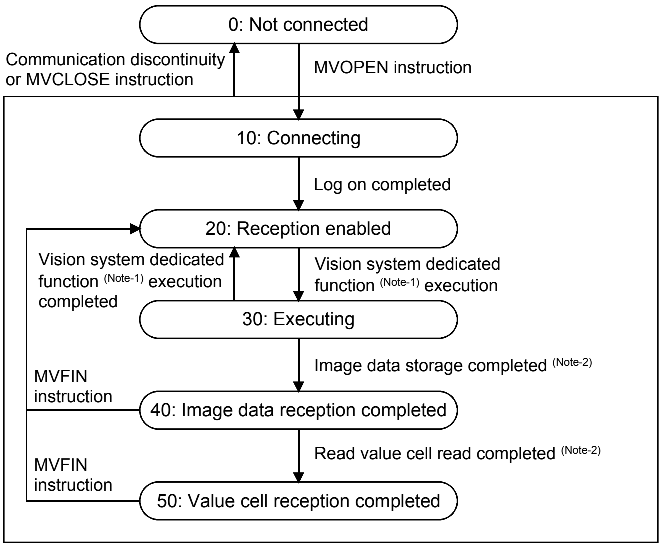


Both of them are set to 0 at the Multiple CPU system's power supply ON.

Refer to Section 12.5 for the error code of vision system dedicated function.

The vision system status is indicated with the following values.

Storage value	Status	
0	Not connected	Status before logging onto the vision system.
10	Connecting	Status while executing log onto the vision system.
20	Reception enabled	Status in which the vision system has been logged onto, and the vision system dedicated functions can be executed.
30	Executing	Status in which vision system dedicated functions are being executed. Other vision system dedicated functions cannot be executed in this status.
40	Image data reception completed	Status in which the vision system job executed by the vision system dedicated function has been completed, and batch send of the image data has been completed. The image data storage device value can be used by the Motion SFC.
50	Value cell reception completed	Status in which the Motion CPU has received the data acquired by the job in the vision system. The read data storage device value can be used by the Motion SFC.



(Note-1): MVLOAD, MVTRG, MVPST, MVIN, MVOUT, MVCLOSE and MVCOM instruction

(Note-2): When the vision system job is started by the MVTRG instruction or MVPST instruction, if the image data storage device or read value storage device is set, the program will jump to line 40 or 50 at a normal completion. If the process ends abnormally, the program will jump to line 20.

(g) Error Flag

Set the bit device that turns ON if an error is detected when the vision system dedicated function is executed.

Settable bit devices are shown below.

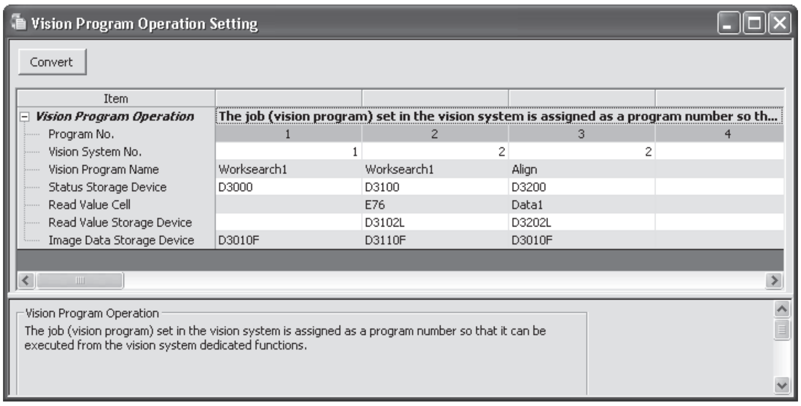
Item	Device No. setting range
Input relay	X0 to X1FFF (Note-1)
Output relay	Y0 to Y1FFF
Internal relay	M0 to M8191 (Note-2)
Link relay	B0 to B1FFF
Annunciator	F0 to F2047

(Note-1): The range of "PXn+4 to PXn+F" cannot be used (fixed at 0) for the input device (PXn+0 to PXn+F) allocated to the built-in interface in Motion CPU (DI). (n: First input No.). QDS

(Note-2): Only the user device range can be set.

(2) Vision Program Operation Setting

The job (vision program) set in the vision system is assigned as a program number so that it can be executed from the vision system dedicated functions.



(a) Program No. (Not necessary to set)
This number (1 to 32) is used by the vision system dedicated function to identify the vision system job.

(b) Vision System (camera) No.
Set the vision system number corresponding to the vision system that is executing the job.

(c) Vision Program Name
Set the name of the job executed by the vision system dedicated function.

(d) Status Storage Device
Set the word device that stores the job's load status and the vision system's online/offline status.
Settable word devices are shown below.

Item	Device No. setting range
Data register	D0 to D8191 (Note-1)
Link register	W0 to W1FFF
Motion register	#0 to #7999

(Note-1): Only the user device range can be set.

Both of them are set to 0 at the Multiple CPU system's power supply ON.
The job's load status is indicated with the following values.

Storage value	Status	
0	Job not loaded or offline	The job is not loaded, or even if loaded is offline. (An error occurs if a trigger is issued.)
1	Job loading completed and online	The job has been loaded and is online. (The job is executed if a trigger is issued.)

POINT

The vision program status storage device value is refreshed at the following timing.
(1) When a job is loaded by the MVLOAD instruction or MVPST instruction. (Refreshed immediately after loading.)
(2) When the vision system's online/offline status changes.
(3) When a job is loaded from a source other than the Motion CPU (In-Sight® Explorer, etc.). (Refreshed several seconds after loading.)
(4) When job is loaded and an online vision system is logged onto with the MVOPEN instruction. (Refreshed several seconds after logging on.)

(e) Read Value Cell/Read Value Storage Device
Set this to store the vision system tag or numeric data of spreadsheet in the Motion CPU device.
This does not need to be set if the numeric data does not need to be referred to.

POINT

(1) The vision system image data can be stored in the image data storage device by setting the format output string setting of TCP/IP protocol.
(2) When the data stored in the set tag or spreadsheet cell is not an integer value, the value after truncation of decimal point is stored in the read value storage device.
(3) If a spreadsheet cell is designated when using the vision system In-Sight® EZ series, an error will occur when the job is executed.

The tag or spreadsheet cell is set in the Read Value Cell.

Setting with tag	Write the symbolic tag name in the original state. (Example) Tag: Job.Pass_count → Set "Job.Pass_count".
Setting with cell	Write the spreadsheet row (A to Z) and line (0 to 399). (Example) Cell: A5 → Set "A5".

The word device storing the value set in the tag or spreadsheet cell is set in the read value storage device.
Settable word devices are shown below.

Item	Device No. setting range
Data register	D0 to D8190 (Note-1)
Link register	W0 to W1FFE
Motion register	#0 to #7998

(Note-1): Only the user device range can be set.

POINT

Set the device No. as an even-number.

The cell or tag value is stored as a 32-bit integer value in two successive points of the set device.

(f) Image Data Storage Device
Set the word device for storing the image data obtained when the job was executed.

POINT
The image data is stored only when the format output string setting of TCP/IP protocol is set in the vision system. (Refer to (3) in this section.)

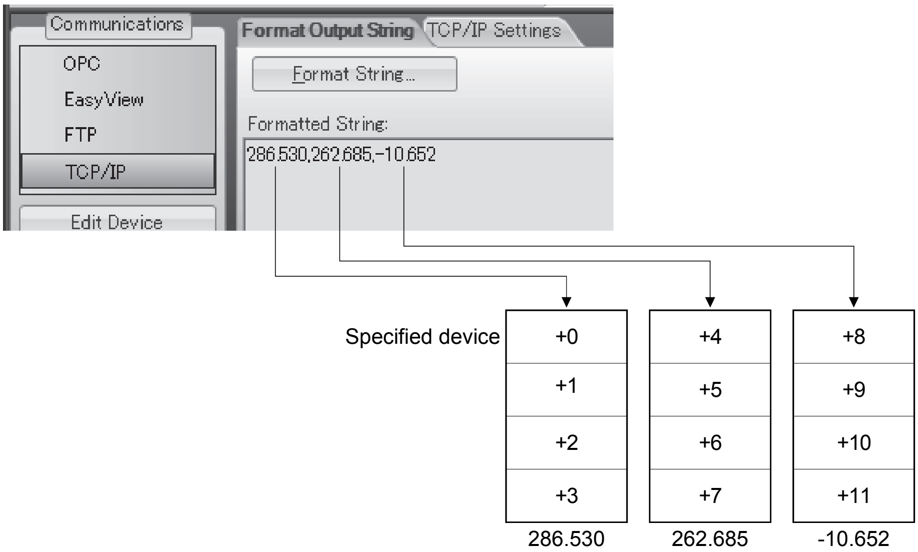
This does not need to be set if the format output string setting of TCP/IP protocol is not set in the vision system.
Settable word devices are shown below.

Item	Device No. setting range
Data register	D0 to D8188 (Note-1)
Link register	W0 to W1FFC
Motion register	#0 to #7996

(Note-1): Only the user device range can be set.

POINT
Set the device No. as an even-number.

The image data is stored as a 64-bit floating point type every four successive points from the specified device equivalent to the output data set with the format output string setting of TCP/IP protocol.
Use the type conversion instruction of Motion SFC according to the application.

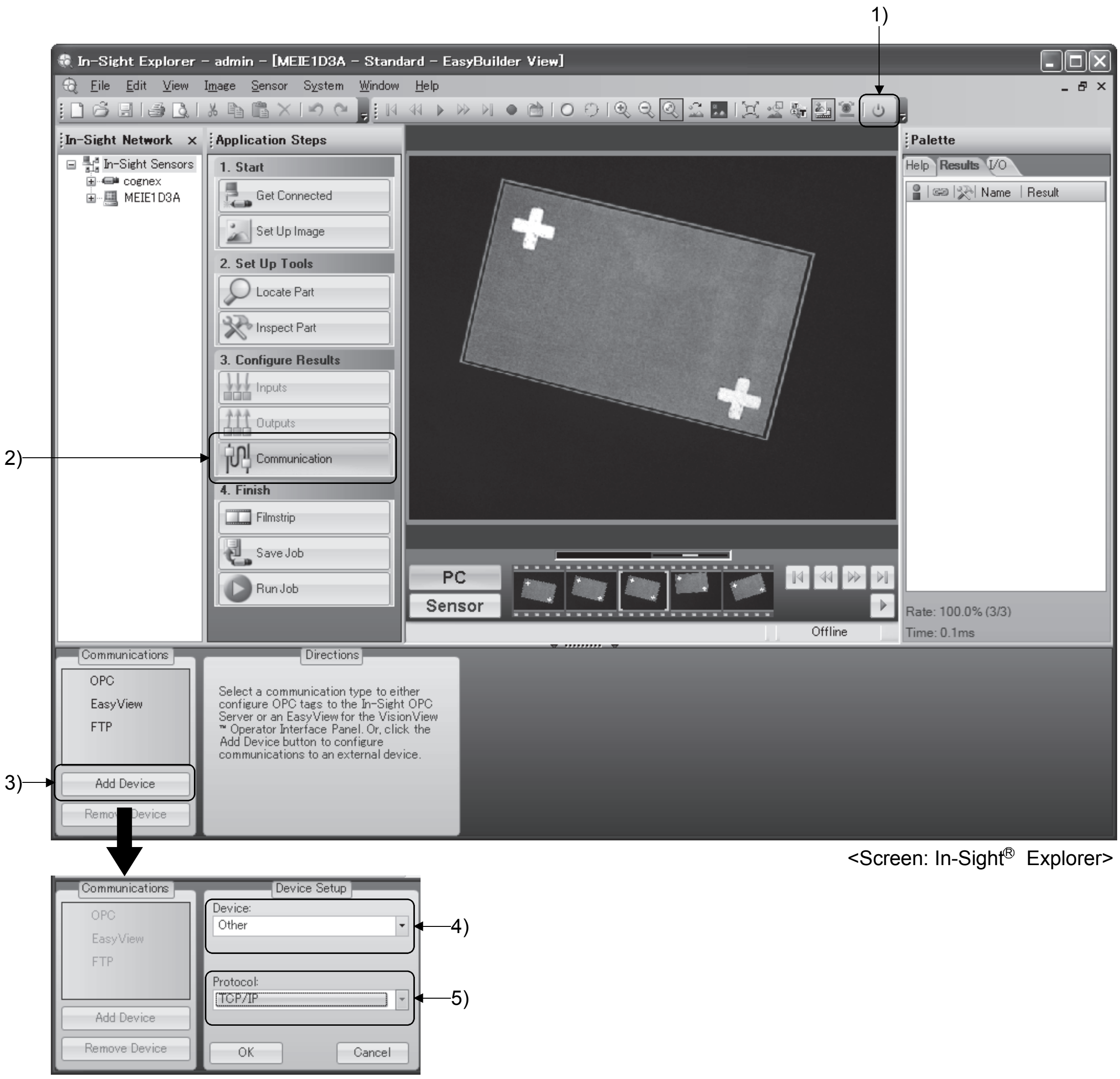


(3) Setting batch send (TCP/IP protocol) of multiple data

By using the format output string setting of TCP/IP protocol, image data after the job is finished can be sent in a batch to the Motion CPU.

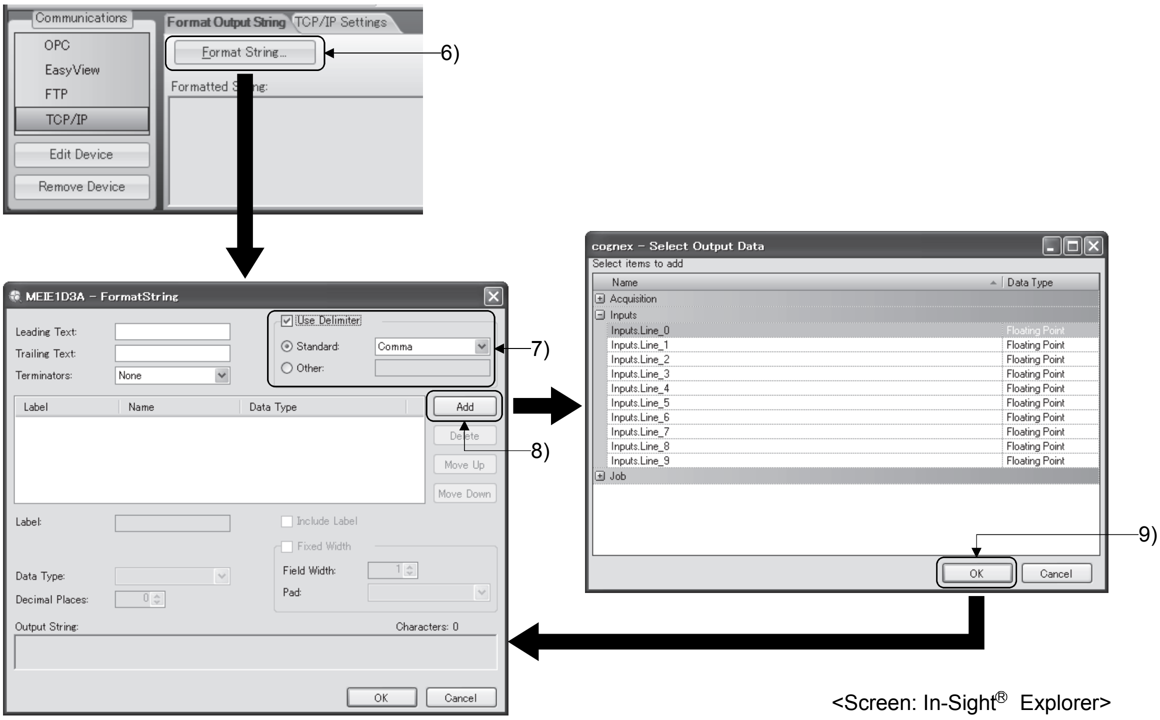
Set with the following procedure using In-Sight® Explorer.

1) Click the [Online] icon on the tool bar, and shift to the offline mode.
2) Click the [Communication] button under Application Steps to display the Communications screen.
3) Click the [Add Device] button to display the screen of Device Setup.
4) Select "Other" for "Device".
5) Select "TCP/IP" for "Protocol", and click the [OK] button.



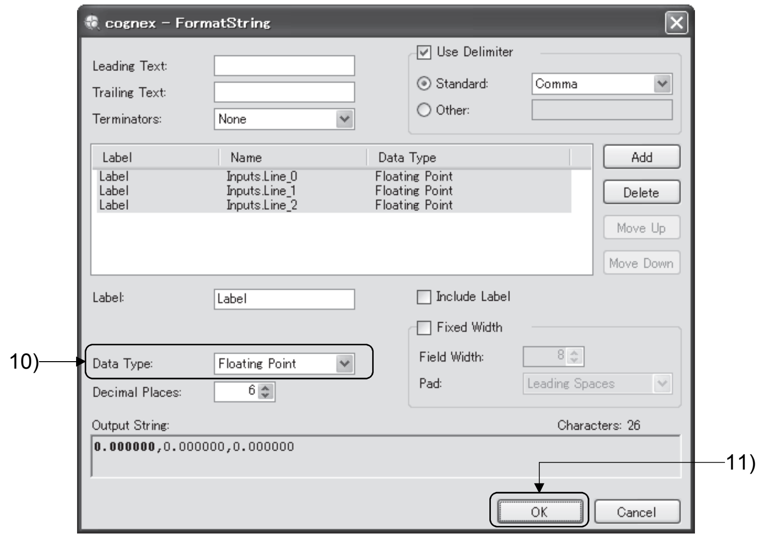
<Screen: In-Sight® Explorer>

6) The “TCP/IP” device will be added. Click the [Format String] button to display the FormatString dialog.
7) Set "Use Delimiter", and set the selectable character with "Standard".
8) Click the [Add] button to display the Select Output Data dialog.
9) Select the data to be sent to the Motion CPU as the result of the job execution, and then click the [OK] button.



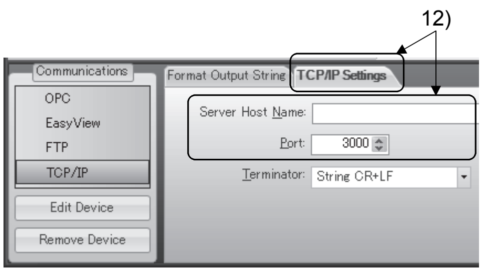
<Screen: In-Sight® Explorer>

10) Set "Data Type" for the added data.
Select "Integer", "Unsigned Integer" or "Floating Point".
The data type stored in the Motion CPU device is always a 64-bit floating point type regardless of the data type set here.
11) Click [OK] button to close the FormatString dialog.



<Screen: In-Sight® Explorer>

12) Check the "TCP/IP Settings".
Leave the Server Host Name blank. (The vision system acts as the TCP/IP server.)
The port number must be the same as the port No. for TCP/IP communication set with the Ethernet communication line setting. (Refer to (1) in this section.)



<Screen: In-Sight® Explorer>

APPENDIX 3.3 Flow of vision system control

This section explains the basic procedures for controlling the vision system from the Motion CPU.

(1) Setting the vision system

Set the vision system network and create a job (vision program) using In-Sight® Explorer.

(2) Setting the Motion CPU parameters

Set the Ethernet communication line setting and the vision program operation setting using MELSOFT MT Works2. (Refer to Appendix 3.2.)

(3) Controlling the vision system with vision system dedicated functions of Motion SFC

1) Log onto the control target vision system using the MVOPEN instruction.
2) Load the job (vision program) to be used using the MVLOAD instruction.
3) Issue a trigger to the vision system using the MVTRG instruction or vision system's image acquire trigger input.
   When the MVPST instruction is used, the job can be loaded and the trigger issued simultaneously.
4) When the vision system finishes executing the job, the job execution results are stored into the device set with the parameters (image data storage device and read value storage device) of Motion CPU.

In addition to the above procedure, data can be acquired from the vision system using the MVIN instruction or MC protocol.
Select the method that suits the required data acquisition time or data type.

Data acquisition method	Communication protocol	Data acquisition time	Output data type of vision system	Storage data type to the device	Batch acquisition of multiple data
Image data storage device	TCP/IP protocol	Fast ↑	Integer value	64-bit floating point type (Automatic conversion)	○
			Floating point value		
Read value storage device	Telnet		Integer value	32-bit integer type (Automatic conversion)	×
MVIN instruction	Telnet		Integer value	32-bit integer type (Automatic conversion)	×
			Floating point value	64-bit floating point type (Automatic conversion)	
MC protocol	MC protocol		Integer value	16-bit integer type or 32-bit integer type (According to vision system output)	○
		↓ Slow	Floating point value	32-bit floating point type (Note-1)	
			Character string	Character string (Note-2)	

○: Enable ×: Disable

(Note-1): Convert to the 64-bit floating point type using the DFLT instruction to use with operation of Motion SFC.
(Note-2): Use the MVCOM instruction (ASCII mode) to acquire the character string data without using MC protocol.

5) Motion control is executed using the data acquired from the vision system.
6) Reset the status storage device using the MVFIN instruction to issue the next trigger.
7) If the job is not changed, repeat steps 3) to 6).
8) If necessary, log off the control target vision system using the MVCLOSE instruction.

POINT

(1) If a different vision system dedicated function is executed for a vision system that is processing a vision system dedicated function, a vision command invalid start (error code: 18018) will occur.
Apply the interlock conditions with the vision system's status storage device value to prevent double startup.
(2) Depending on the status of the vision system and details of the job process, it may take some time to process the vision system dedicated function.
Set the timeout time according to the state.
(3) When logged onto the vision system, the vision system or Multiple CPU system's power supply can be turned OFF without logging off using the MVCLOSE instruction.

APPENDIX 3.4 Sample program

(1) Explanation of the operations

The following section gives an example of a program that executes positioning control using the adjustment data recognized by the vision system as the target data.

(2) Setting the vision system

Complete the following settings with In-Sight® Explorer.
Refer to the manual or help sections provided by Cognex for details on operating and setting In-Sight® Explorer.

(a) Ethernet communication setting

Item		Setting value
IP address		10.0.50.100
Port No.	Telnet	23
	TCP/IP	3000

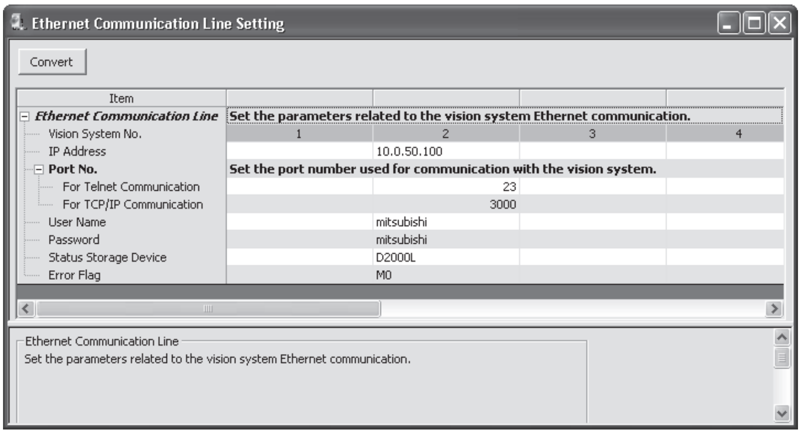
(b) Job setting

Item	Setting value		
Job name	Worksearch1		
TCP/IP protocol - Format output string	1	Pattern_1.fIXTURE.x	Floating point
	2	Pattern_1.fIXTURE.y	Floating point

(3) Vision system parameter setting

Complete the vision system parameter setting of MT Developer2.

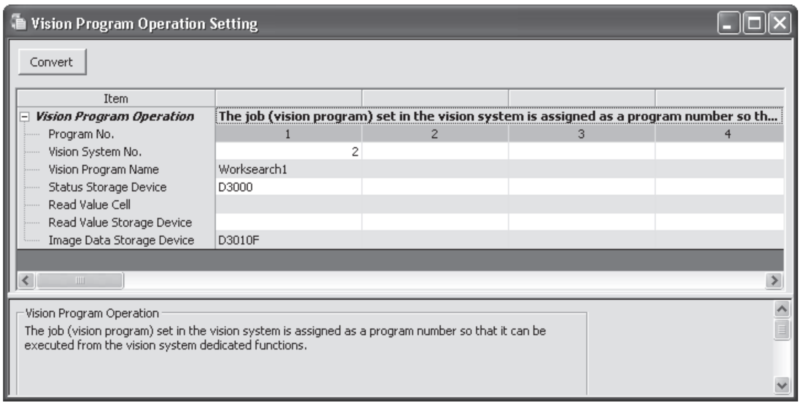
(a) Ethernet Communication Line Setting



Vision System No.2

Setting item		Description
IP address		10.0.50.100
Port No.	Telnet	23
	TCP/IP	3000
User Name		According to the vision system setting
Password		
Status Storage Device		D2000L
Error Flag		M0

(b) Vision program operation setting



Program No.1

<table>
<tr><th>Setting item</th><th>Description</th></tr>
<tr><td>Vision System No.</td><td>2</td></tr>
<tr><td>Vision Program Name</td><td>Worksearch1</td></tr>
<tr><td>Status Storage Device</td><td>D3000</td></tr>
<tr><td>Read Value Cell</td><td rowspan="2">Not necessary to set</td></tr>
<tr><td>Read Value Storage Device</td></tr>
<tr><td>Image Data Storage Device</td><td>D3010F</td></tr>
</table>

(4) Motion SFC program

Program	Description
Alignment ajustment	
[F0] //Log onto vision system MVOPEN K2	Log onto vision system of vision sensor (camera) No. 2.
[G0] //Confirm log on completion (D2000==K20)*!M0	Confirm log on completion (20) with vision system status storage device (D2000). Confirm that there is no error with error flag (M0).
[F1] //Load job MVLOAD K1	Load job "Worksearch1" set in program No. 1.
[G1] //Confirm load completion of job (D3000==K1)*!M0	Confirm completion of job loading and online status (1) with program status storage device (D3000). Confirm that there is no error with error flag (M0).
P0	
[F2] //Reset status storage device MVFIN K2	Set status storage device reset in vision sensor (camera) No. 2 vision system.
[G2] //Wait for trigger request PX0	Wait for trigger request (PX0).
[F3] //Issue trigger to vision system MVTRG K2	Issue trigger to vision system of vision sensor (camera) No. 2. .
[G3] //Wait for complete of image data reception (D2000==40)*!M0	Wait for completion of image data reception (40) with vision system status storage device (D2000). Confirm that there is no error with error flag (M0).

[F4]
//Calculate command position from image data
 	
D4000L=LONG(D3010F*10000.0) //Axis 1 (X)
D4002L=LONG(D3014F*10000.0) //Axis 2 (Y)
	Convert data [mm] stored in image data storage device (D3010 to) into position command unit [0.1 μm].

[K1:Real] 1 ABS-2(Vector-speed) Axis 1, D4000 μm Axis 2, D4002 μm Vector Speed 1000 mm/s	Move to command position.
[G4] //Wait for positioning completion M2402*M2422	Wait for positioning completion (axis 1, axis 2 in-position).
P0	

MEMO

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

We will repair any failure or defect hereinafter referred to as "failure" in our FA equipment hereinafter referred to as the "Product" arisen during warranty period at no charge due to causes for which we are responsible through the distributor from which you purchased the Product or our service provider. However, we will charge the actual cost of dispatching our engineer for an on-site repair work on request by customer in Japan or overseas countries. We are not responsible for any on-site readjustment and/or trial run that may be required after a defective unit is repaired or replaced.

[Gratis Warranty Term]

The term of warranty for Product is thirty six (36) months after your purchase or delivery of the Product to a place designated by you or forty two (42) months from the date of manufacture whichever comes first "Warranty Period". Warranty period for repaired Product cannot exceed beyond the original warranty period before any repair work.

[Gratis Warranty Range]

(1) You are requested to conduct an initial failure diagnosis by yourself, as a general rule.
It can also be carried out by us or our service company upon your request and the actual cost will be charged. However, it will not be charged if we are responsible for the cause of the failure.

(2) This limited warranty applies only when the condition, method, environment, etc. of use are in compliance with the terms and conditions and instructions that are set forth in the instruction manual and user manual for the Product and the caution label affixed to the Product.

(3) Even during the term of warranty, the repair cost will be charged on you in the following cases;
1) A failure caused by your improper storing or handling, carelessness or negligence, etc., and a failure caused by your hardware or software problem
2) A failure caused by any alteration, etc. to the Product made on your side without our approval
3) A failure which may be regarded as avoidable, if your equipment in which the Product is incorporated is equipped with a safety device required by applicable laws and has any function or structure considered to be indispensable according to a common sense in the industry
4) A failure which may be regarded as avoidable if consumable parts designated in the instruction manual, etc. are duly maintained and replaced
5) Any replacement of consumable parts (battery, fan, etc.)
6) A failure caused by external factors such as inevitable accidents, including without limitation fire and abnormal fluctuation of voltage, and acts of God, including without limitation earthquake, lightning and natural disasters
7) A failure generated by an unforeseeable cause with a scientific technology that was not available at the time of the shipment of the Product from our company
8) Any other failures which we are not responsible for or which you acknowledge we are not responsible for

2. Onerous Repair Term after Discontinuation of Production

(1) We may accept the repair at charge for another seven (7) years after the production of the product is discontinued.
The announcement of the stop of production for each model can be seen in our Sales and Service, etc.

(2) Please note that the Product (including its spare parts) cannot be ordered after its stop of production.

3. Service in overseas countries

Our regional FA Center in overseas countries will accept the repair work of the Product; However, the terms and conditions of the repair work may differ depending on each FA Center. Please ask your local FA center for details.

4. Exclusion of Loss in Opportunity and Secondary Loss from Warranty Liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:
(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Change of Product specifications

Specifications listed in our catalogs, manuals or technical documents may be changed without notice.

6. Precautions for Choosing the Products

(1) For the use of our Motion controller, its applications should be those that may not result in a serious damage even if any failure or malfunction occurs in Motion controller, and a backup or fail-safe function should operate on an external system to Motion controller when any failure or malfunction occurs.

(2) Our Motion controller is designed and manufactured as a general purpose product for use at general industries. Therefore, applications substantially influential on the public interest for such as atomic power plants and other power plants of electric power companies, and also which require a special quality assurance system, including applications for railway companies and government or public offices are not recommended, and we assume no responsibility for any failure caused by these applications when used.
In addition, applications which may be substantially influential to human lives or properties for such as airlines, medical treatments, railway service, incineration and fuel systems, man-operated material handling equipment, entertainment machines, safety machines, etc. are not recommended, and we assume no responsibility for any failure caused by these applications when used.
We will review the acceptability of the abovementioned applications, if you agree not to require a specific quality for a specific application. Please contact us for consultation.

Microsoft, Windows, Windows NT, Windows Vista, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Ethernet is a registered trademark of Fuji Xerox Corporation in Japan.
The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.
In some cases, trademark symbols such as '™' or '®' are not specified in this manual.

IB(NA)-0300135-J(1703)MEE
MODEL: Q173D-P-SV13/22-SFCE
MODEL CODE: 1XB929

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.