ASEL

ASEL Controller

Operation Manual Eighth Edition

IAI America, Inc.

Please Read Before Use

Thank you for purchasing our product.

This Operation Manual explains the handling methods, structure and maintenance of this product, among others, providing the information you need to know to use the product safely.

Before using the product, be sure to read this manual and fully understand the contents explained herein to ensure safe use of the product.
The CD or DVD that comes with the product contains operation manuals for IAI products.
When using the product, refer to the necessary portions of the applicable operation manual by printing them out or displaying them on a PC.

After reading the Operation Manual, keep it in a convenient place so that whoever is handling this product can reference it quickly when necessary.

[Important]

? This Operation Manual is original.
? The product cannot be operated in any way unless expressly specified in this Operation Manual. IAI shall assume no responsibility for the outcome of any operation not specified herein.
? Information contained in this Operation Manual is subject to change without notice for the purpose of product improvement.
? If you have any question or comment regarding the content of this manual, please contact the IAI sales office near you.
? Using or copying all or part of this Operation Manual without permission is prohibited.
? The company names, names of products and trademarks of each company shown in the sentences are registered trademarks.

Operator Alarm on Low Battery Voltage

This controller is equipped with the following backup batteries for retention of data in the event of power failure:

[1] System-memory backup battery (optional)
For retention of position data, global variables/flags, error list, strings, etc.

[2] Absolute-encoder backup battery (absolute specification)
For retention of multi-rotation data of the encoder

Since these batteries are not rechargeable, they will be eventually consumed. Unless the batteries are replaced in a timely manner, the voltage will drop to a level where the data can no longer be retained. If a power failure occurs in this condition, the data will be lost. (The life of each battery varies depending on the operating time.)

Once the data is lost, the controller will not operate normally the next time the power is turned on, and recovery will take time.

To prevent this problem, this controller can output a low battery voltage alarm from its I/O port.

You can specify a desired output port to issue a low voltage alarm for the system-memory backup battery.

Set "15" as the input function specification value in the I/O parameter corresponding to the output port number you want to specify.

Setting example)
To specify output port No. 306 to issue a low voltage alarm for the system-memory backup battery, set "15" in I/O parameter No. 52 as the input function specification value.

You can specify a desired output port to issue a low voltage alarm for the absolute-data backup battery.

Set "16" as the input function specification value in the I/O parameter corresponding to the output port number you want to specify.

Setting example)
To specify output port No. 307 to issue a low voltage alarm for the absolute-data backup battery, set "16" in I/O parameter No. 53 as the input function specification value.

It is recommended that this function be utilized to prevent unnecessary problems resulting from low battery voltage (consumption of battery life).

In particular, the person in charge of overall system design should utilize this function to provide a design means for issuing an operator alarm using an output signal from an I/O port, while the person in charge of electrical design should provide an electrical means for achieving the same effect.

For the battery replacement procedure, refer to the applicable section in the operating manual.

It is recommended that you always back up the latest data to a PC in case of voltage drop in the system-memory backup battery or unexpected controller failure.

Optional System-Memory Backup Battery

The ASEL controller can be used with the optional system-memory backup battery.

Caution: When installing the system-memory backup battery, “Other parameter No. 20” must be set to “2.”

Installing the system-memory backup battery will add the following functions to the controller:

- Save SEL global data
  Data of global variables, flags and strings will be retained even after the main power is turned off.
- Save RAM position data
  Position data changed by SEL programs will be retained even after the main power is turned off.
- Save an error list
  An error list containing up to 100 most recent errors will be retained even after the main power is turned off.

If you need any or all of the above functions, you must install the optional system-memory backup battery.

Table of Contents

Safety Guide

“Safety Guide” has been written to use the machine safely and so prevent personal injury or property damage beforehand. Make sure to read it before the operation of this product.

Safety Precautions for Our Products

The common safety precautions for the use of any of our robots in each operation.

No.	Operation Description	Description
1	Model Selection	● This product has not been planned and designed for the application where high level of safety is required, so the guarantee of the protection of human life is impossible. Accordingly, do not use it in any of the following applications. 1) Medical equipment used to maintain, control or otherwise affect human life or physical health. 2) Mechanisms and machinery designed for the purpose of moving or transporting people (For vehicle, railway facility or air navigation facility) 3) Important safety parts of machinery (Safety device, etc.) ● Do not use the product outside the specifications. Failure to do so may considerably shorten the life of the product. ● Do not use it in any of the following environments. 1) Location where there is any inflammable gas, inflammable object or explosive 2) Place with potential exposure to radiation 3) Location with the ambient temperature or relative humidity exceeding the specification range 4) Location where radiant heat is added from direct sunlight or other large heat source 5) Location where condensation occurs due to abrupt temperature changes 6) Location where there is any corrosive gas (sulfuric acid or hydrochloric acid) 7) Location exposed to significant amount of dust, salt or iron powder 8) Location subject to direct vibration or impact ● For an actuator used in vertical orientation, select a model which is equipped with a brake. If selecting a model with no brake, the moving part may drop when the power is turned OFF and may cause an accident such as an injury or damage on the work piece.

No.	Operation Description	Description
2	Transportation	● When carrying a heavy object, do the work with two or more persons or utilize equipment such as crane. ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● When in transportation, consider well about the positions to hold, weight and weight balance and pay special attention to the carried object so it would not get hit or dropped. ● Transport it using an appropriate transportation measure. The actuators available for transportation with a crane have eyebolts attached or there are tapped holes to attach bolts. Follow the instructions in the operation manual for each model. ● Do not step or sit on the package. ● Do not put any heavy thing that can deform the package, on it. ● When using a crane capable of 1t or more of weight, have an operator who has qualifications for crane operation and sling work. ● When using a crane or equivalent equipments, make sure not to hang a load that weighs more than the equipment's capability limit. ● Use a hook that is suitable for the load. Consider the safety factor of the hook in such factors as shear strength. ● Do not get on the load that is hung on a crane. ● Do not leave a load hung up with a crane. ● Do not stand under the load that is hung up with a crane.
3	Storage and Preservation	● The storage and preservation environment conforms to the installation environment. However, especially give consideration to the prevention of condensation. ● Store the products with a consideration not to fall them over or drop due to an act of God such as earthquake.
4	Installation and Start	(1) Installation of Robot Main Body and Controller, etc. ● Make sure to securely hold and fix the product (including the work part). A fall, drop or abnormal motion of the product may cause a damage or injury. Also, be equipped for a fall-over or drop due to an act of God such as earthquake. ● Do not get on or put anything on the product. Failure to do so may cause an accidental fall, injury or damage to the product due to a drop of anything, malfunction of the product, performance degradation, or shortening of its life. ● When using the product in any of the places specified below, provide a sufficient shield. 1) Location where electric noise is generated 2) Location where high electrical or magnetic field is present 3) Location with the mains or power lines passing nearby 4) Location where the product may come in contact with water, oil or chemical droplets

No.	Operation Description	Description
4	Installation and Start	(2) Cable Wiring ● Use our company's genuine cables for connecting between the actuator and controller, and for the teaching tool. ● Do not scratch on the cable. Do not bend it forcibly. Do not pull it. Do not coil it around. Do not insert it. Do not put any heavy thing on it. Failure to do so may cause a fire, electric shock or malfunction due to leakage or continuity error. ● Perform the wiring for the product, after turning OFF the power to the unit, so that there is no wiring error. ● When the direct current power (+24V) is connected, take the great care of the directions of positive and negative poles. If the connection direction is not correct, it might cause a fire, product breakdown or malfunction. ● Connect the cable connector securely so that there is no disconnection or looseness. Failure to do so may cause a fire, electric shock or malfunction of the product. ● Never cut and/or reconnect the cables supplied with the product for the purpose of extending or shortening the cable length. Failure to do so may cause the product to malfunction or cause fire.
		(3) Grounding ● The grounding operation should be performed to prevent an electric shock or electrostatic charge, enhance the noise-resistance ability and control the unnecessary electromagnetic radiation. ● For the ground terminal on the AC power cable of the controller and the grounding plate in the control panel, make sure to use a twisted pair cable with wire thickness 0.5mm^2 (AWG20 or equivalent) or more for grounding work. For security grounding, it is necessary to select an appropriate wire thickness suitable for the load. Perform wiring that satisfies the specifications (electrical equipment technical standards). ● Perform Class D Grounding (former Class 3 Grounding with ground resistance 100Ω or below).

No.	Operation Description	Description
4	Installation and Start	(4) Safety Measures ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● When the product is under operation or in the ready mode, take the safety measures (such as the installation of safety and protection fence) so that nobody can enter the area within the robot's movable range. When the robot under operation is touched, it may result in death or serious injury. ● Make sure to install the emergency stop circuit so that the unit can be stopped immediately in an emergency during the unit operation. ● Take the safety measure not to start up the unit only with the power turning ON. Failure to do so may start up the machine suddenly and cause an injury or damage to the product. ● Take the safety measure not to start up the machine only with the emergency stop cancellation or recovery after the power failure. Failure to do so may result in an electric shock or injury due to unexpected power input. ● When the installation or adjustment operation is to be performed, give clear warnings such as "Under Operation; Do not turn ON the power!" etc. Sudden power input may cause an electric shock or injury. ● Take the measure so that the work part is not dropped in power failure or emergency stop. ● Wear protection gloves, goggle or safety shoes, as necessary, to secure safety. ● Do not insert a finger or object in the openings in the product. Failure to do so may cause an injury, electric shock, damage to the product or fire. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity.
5	Teaching	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● Perform the teaching operation from outside the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the "Stipulations for the Operation" and make sure that all the workers acknowledge and understand them well. ● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency. ● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly. ● Place a sign "Under Operation" at the position easy to see. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity. * Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated.

No.	Operation Description	Description
6	Trial Operation	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● After the teaching or programming operation, perform the check operation one step by one step and then shift to the automatic operation. ● When the check operation is to be performed inside the safety protection fence, perform the check operation using the previously specified work procedure like the teaching operation. ● Make sure to perform the programmed operation check at the safety speed. Failure to do so may result in an accident due to unexpected motion caused by a program error, etc. ● Do not touch the terminal block or any of the various setting switches in the power ON mode. Failure to do so may result in an electric shock or malfunction.
7	Automatic Operation	● Check before starting the automatic operation or rebooting after operation stop that there is nobody in the safety protection fence. ● Before starting automatic operation, make sure that all peripheral equipment is in an automatic-operation-ready state and there is no alarm indication. ● Make sure to operate automatic operation start from outside of the safety protection fence. ● In the case that there is any abnormal heating, smoke, offensive smell, or abnormal noise in the product, immediately stop the machine and turn OFF the power switch. Failure to do so may result in a fire or damage to the product. ● When a power failure occurs, turn OFF the power switch. Failure to do so may cause an injury or damage to the product, due to a sudden motion of the product in the recovery operation from the power failure.

No.	Operation Description	Description
8	Maintenance and Inspection	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● Perform the work out of the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the “Stipulations for the Operation” and make sure that all the workers acknowledge and understand them well. ● When the work is to be performed inside the safety protection fence, basically turn OFF the power switch. ● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency. ● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly. ● Place a sign “Under Operation” at the position easy to see. ● For the grease for the guide or ball screw, use appropriate grease according to the Operation Manual for each model. ● Do not perform the dielectric strength test. Failure to do so may result in a damage to the product. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity. ● The slider or rod may get misaligned OFF the stop position if the servo is turned OFF. Be careful not to get injured or damaged due to an unnecessary operation. ● Pay attention not to lose the cover or untightened screws, and make sure to put the product back to the original condition after maintenance and inspection works. Use in incomplete condition may cause damage to the product or an injury. * Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated.
9	Modification and Dismantle	● Do not modify, disassemble, assemble or use of maintenance parts not specified based at your own discretion.
10	Disposal	● When the product becomes no longer usable or necessary, dispose of it properly as an industrial waste. ● When removing the actuator for disposal, pay attention to drop of components when detaching screws. ● Do not put the product in a fire when disposing of it. The product may burst or generate toxic gases.
11	Other	● Do not come close to the product or the harnesses if you are a person who requires a support of medical devices such as a pacemaker. Doing so may affect the performance of your medical device. ● See Overseas Specifications Compliance Manual to check whether complies if necessary. ● For the handling of actuators and controllers, follow the dedicated operation manual of each unit to ensure the safety.

Alert Indication

The safety precautions are divided into “Danger”, “Warning”, “Caution” and “Notice” according to the warning level, as follows, and described in the Operation Manual for each model.

Level	Degree of Danger and Damage	Symbol
Danger	This indicates an imminently hazardous situation which, if the product is not handled correctly, will result in death or serious injury.	Danger
Warning	This indicates a potentially hazardous situation which, if the product is not handled correctly, could result in death or serious injury.	Warning
Caution	This indicates a potentially hazardous situation which, if the product is not handled correctly, may result in minor injury or property damage.	Caution
Notice	This indicates lower possibility for the injury, but should be kept to use this product properly.	Notice

CE Marking

If a compliance with the CE Marking is required, please follow Overseas Standards Compliance Manual (ME0287) that is provided separately.

Part 1 Installation

Chapter 1 Overview

1. Introduction

Thank you for purchasing the ASEL Controller.

Please read this manual carefully, and handle the product with due care and operate it correctly.
Keep this manual in a safe place and reference relevant items when needed.

When actually starting up your system or if you have encountered a problem, you should also refer to the manuals for the teaching pendant, PC software and other components used with the system, in addition to this manual.

This manual does not cover all possible operations other than normal operations, or unexpected events such as complex signal changes resulting from use of critical timings.
Accordingly, you should consider items not specifically explained in this manual as "prohibited."

* Utmost effort has been made to ensure accuracy and completeness of the information contained in this manual. However, should you find any error in the manual or if you have any comment regarding its content, please contact IAI.
Keep this manual in a convenient place so that you can quickly reference it whenever necessary.

2. Type

Refer to the following table for details on type specification.

Example of type specification

ASEL - C - 2 - 30A - 30AB - NP - 2 - 0
❶ ❷ ❸ ❹ ❹ ❺ ❻ ❼

Type specification table

❶ Series	❷ Controller type	❸ Number of axes	❹ Details of axis 1 to axis 2				❺ Standard I/O	❻ I/O flat cable length	❼ Power-source voltage
			Motor output (W)	Encoder type	Brake	Home sensor			
ASEL	C (Standard specification) CS (Standard specification)	1 (Axis 1) 2 (Axis 2)	20 (20W) 30 (30W)	I (Incremental) A (Absolute)	Blank (Without brake) B (With brake)	Blank (Without home sensor) L (With home sensor)	NP Standard PIO 24 inputs/8 outputs NPN specification PN Standard PIO 24 inputs/8 outputs PNP specification	2：2m (Standard) 3：3m 5：5m 0：None	0: 24 VDC

3. ASEL Controller Functions

The functions provided by the ASEL controller are structured in the following manner.

- ASEL
  - Program mode
  - Positioner mode
    - Standard mode
    - Product switching mode
    - 2-axis independent mode
    - Teaching mode
    - DS-S-C1 compatible mode

The ASEL controller has the “program mode” in which SEL programs are input to operate the actuator(s), and the “positioner mode” in which position numbers are specified from the host PLC to operate the actuator(s).
The positioner mode provides five sub-modes to meet the needs of various applications.
The program mode has been selected at the factory prior to the shipment of the controller (Other parameter No. 25 = 0).

Caution: Two modes cannot be selected at the same time.

This controller can be configured with one axis and two axes. Just like other conventional SEL controllers, this controller can be combined with various actuators. When connecting an actuator, be sure to use a dedicated cable.

- Turn on the I/O power before or simultaneously with the main power (control power + motor power).
- Take the control power and motor power from the same power supply and turn on both powers simultaneously.
- Before performing a check or inserting/removing a connector, turn off the power and wait for at least 10 minutes.
- About actuator duty
  IAI recommends that our actuators be used at a duty of 50% or less as a guideline in view of the relationship of service life and accuracy:

$$\text{Duty (\%)} = \frac{\text{Acceleration / Deceleration Time}}{\text{Motion time} + \text{Inactivity}} \times 100$$

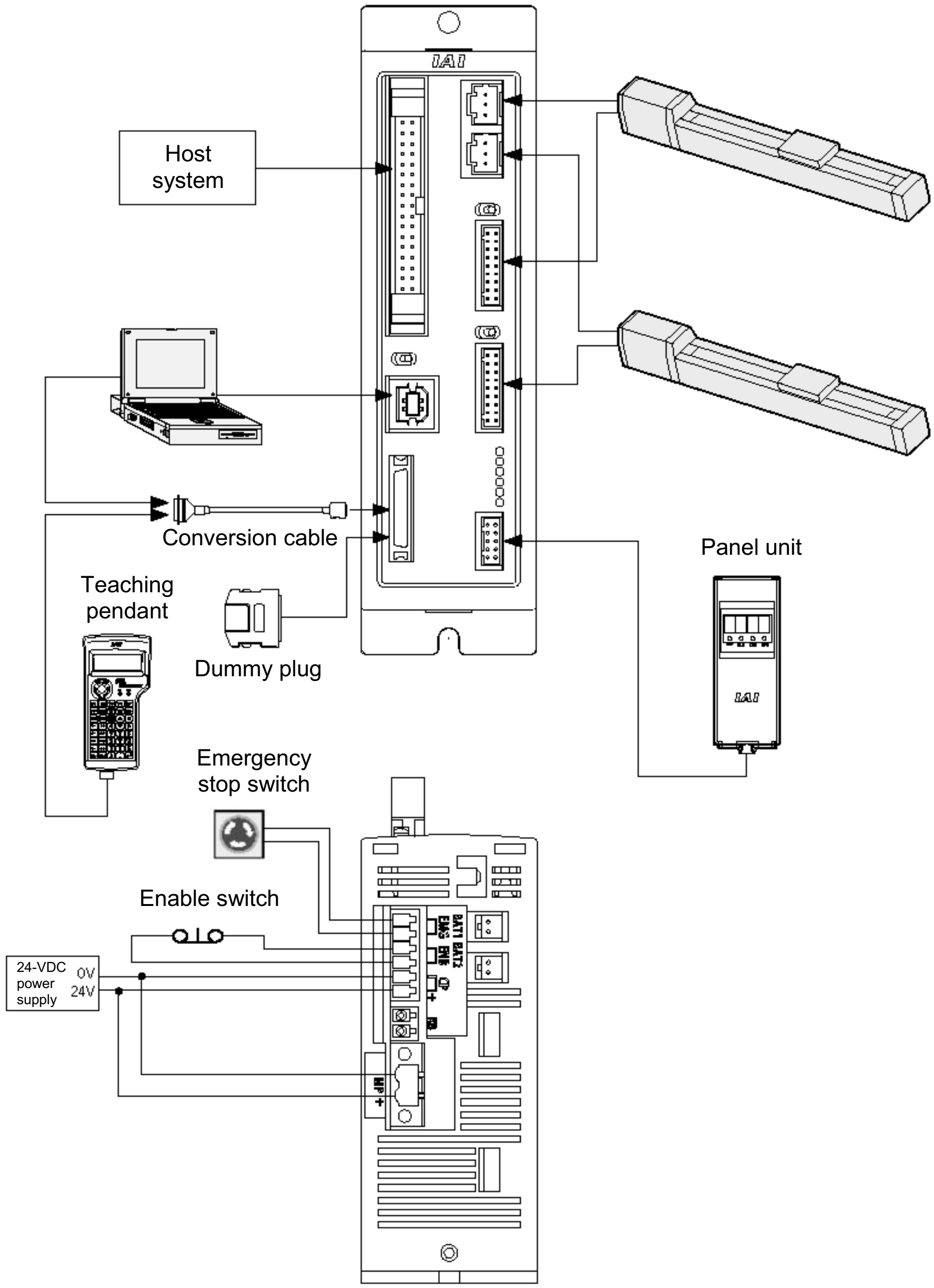
- After turning off the control power, be sure to wait for at least 5 seconds before turning it back on.
- Do not insert or remove connectors while the controller power is on. Doing so may cause malfunction.
- Note on introducing a controller of absolute specification

The following steps must be taken to initialize the absolute-data backup battery circuit to prevent the battery from being consumed quickly. Perform the initialization by following these steps:
[1] Before connecting the encoder cable, disconnect the absolute-data backup battery connector.
[2] Connect the encoder cable.
[3] Turn on the main power.
[4] Connect the absolute-data backup battery.
The above steps must always be performed after the encoder cable has been disconnected for any reason, such as to move the controller.

Read the operation manual for each actuator. If you have purchased our optional PC software and/or teaching pendant, read the respective operation manuals, as well.

* Utmost effort has been made to ensure that the information contained in this manual is true and correct. However, should you find any error or if you have any comment regarding the content, please contact IAI.

4. System Setup



* Note on connecting the encoder cable to a controller of absolute specification
Follow the steps below when connecting the encoder cable to a controller of absolute specification. If the specified steps are not followed, the absolute-data backup battery may be consumed quickly.

[1] Before connecting the encoder cable, disconnect the absolute-data backup battery connector.
[2] Connect the encoder cable, and turn on the main power.
[3] Connect the absolute-data backup battery connector. Once the connector has been plugged in, the main power can be turned off.

For the installation of the absolute-data backup battery, refer to 6.8, "Installation Method for the Absolute-Data Backup Battery" in Chapter 3 of Part 1.
If you have disconnected the encoder cable for any reason, such as to move the controller, also follow the same steps to connect the absolute-data backup battery connector.

5. Warranty Period and Scope of Warranty

The ASEL Controller you have purchased passed our strict outgoing inspection. This unit is covered by the following warranty:

1. Warranty Period

The warranty period shall be either of the following periods, whichever ends first:

- 18 months after shipment from our factory
- 12 months after delivery to a specified location

2. Scope of Warranty

Should the product fail during the above period under a proper use condition due to a fault on the part of the manufacturer, IAI will repair the defect free of charge. However, the following cases are excluded from the scope of warranty:

- Discoloration of paint or other normal aging
- Wear of consumable parts due to use
- Subjective imperfection, such as noise not affecting mechanical function
- Defect caused by inappropriate handling or use by the user
- Defect caused by inappropriate or erroneous maintenance/inspection
- Defect caused by use of a part other than IAI's genuine part
- Defect caused by unauthorized modification, etc., not approved by IAI or its agent
- Defect due to an act of God, accident, fire, etc.

The warranty covers only the product as it is delivered. IAI shall not be liable for any loss arising in connection with the delivered product. The user must bring the defective product to our factory to receive a warranty repair.

3. Scope of Service

The price of the delivered product does not include costs incurred in association with program generation, dispatch of technician, etc. Therefore, a separate fee will be chargeable in the following cases even during the warranty period:

- Guidance on installation/adjustment and witnessing of test operation
- Maintenance/inspection
- Technical guidance and training on operation, wiring method, etc.
- Technical guidance and training regarding programs, such as program generation
- Other services and operations where IAI finds a need to charge a separate fee

Chapter 2 Specifications

1. Controller Specifications

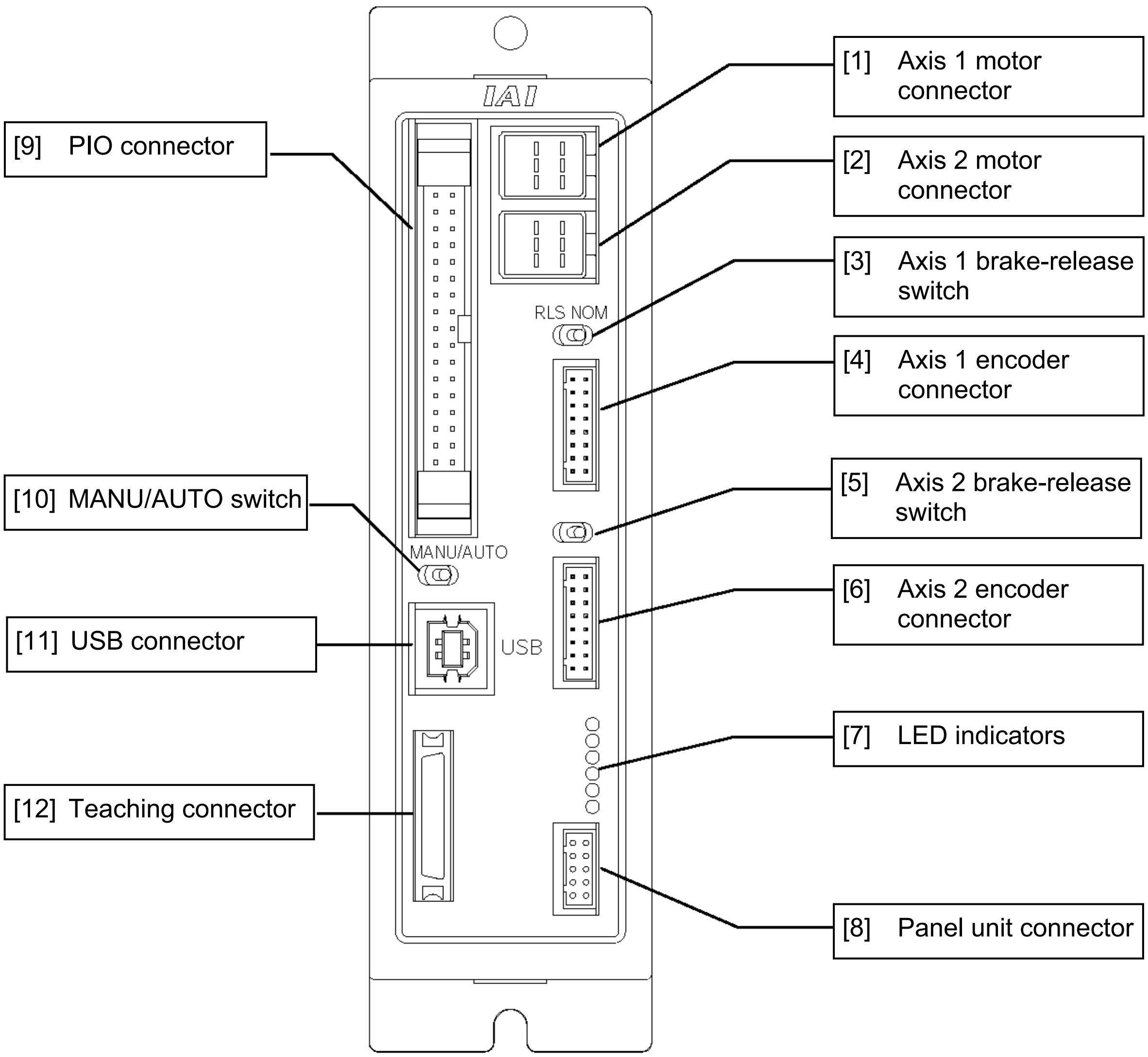
Base specifications of this product

Total output when maximum number of axes are connected	30 W x 2 axes
Control power input	24 VDC ± 10%
Motor power input	24 VDC ± 10%
Resistance against momentary power failure	Maximum 0.5 msec
Withstand voltage	1500 VAC for 1 minute (Measured between all power-supply terminals and FG)
Insulation resistance	500 VDC, 10 MΩ or more
Drive-source cutoff method	Internal relay
Emergency stop input	Contact B input (Internal power-supply type)
Emergency stop action	Deceleration stop + Regenerative brake by timer
Enable input	Contact B input (Internal power-supply type)
Control method	AC full digital servo
Position detection method	Incremental serial encoder Absolute serial encoder ABZ parallel encoder
Battery	Absolute-data backup battery/System-memory backup battery (Optional) Lithium battery: AB-5 by IAI, 3.6 V/2000 mAh
Programming language	Super SEL language
Number of program steps	2000 steps (total)
Number of positions	1500 positions (total)
Number of programs	64 programs
Multi-tasking capability	8 programs
Storage device	Flash ROM
Data input method	Teaching pendant or PC software
PIO power input	24 VDC ± 10%
Safety category	Category B (Built-in relay)
Regenerative resistor	Built-in, 100 Ω (2 W). An external resistor of 22 Ω (5 W) can be connected.
PIO inputs	24 points, NPN or PNP (Selectable as factory setting)
PIO outputs	8 points, NPN or PNP (Selectable as factory setting)
Air cooling method	Natural convection method
Weight	450 g
External dimensions	43 (W) x 159 (H) x 110 (D); mounting pitch 151 mm
Accessories	I/O flat cable Motor power connector Control power & system I/O connector RB connector (Not normally used)

2. Name and Function of Each Part

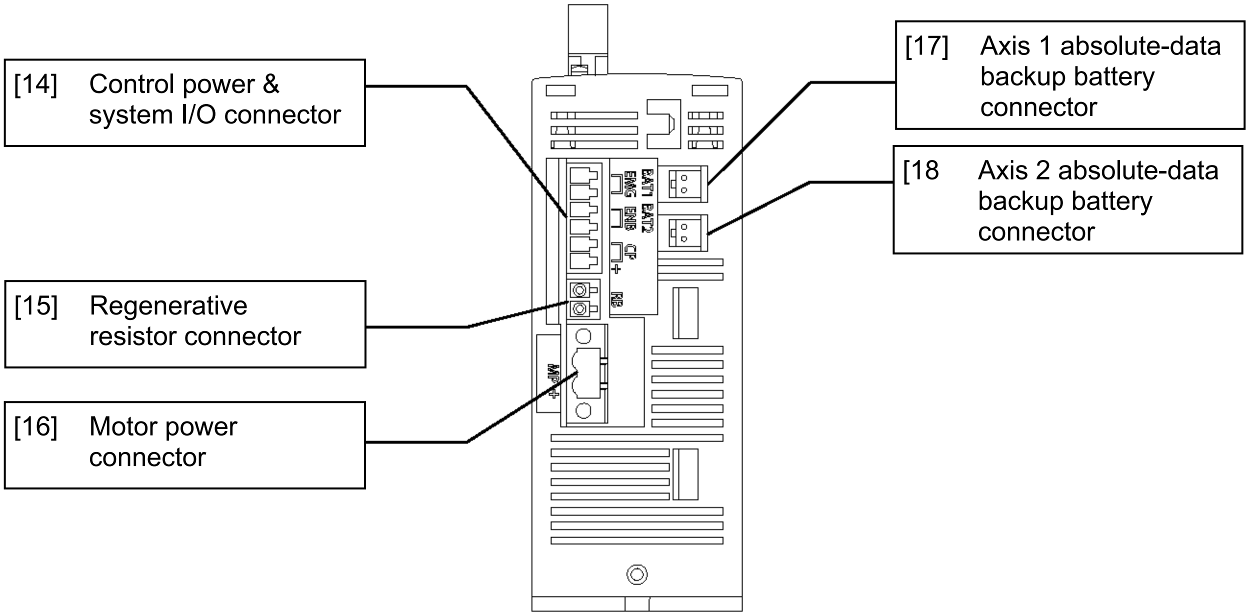
2.1 Name of Each Part

2.1.1 Front View

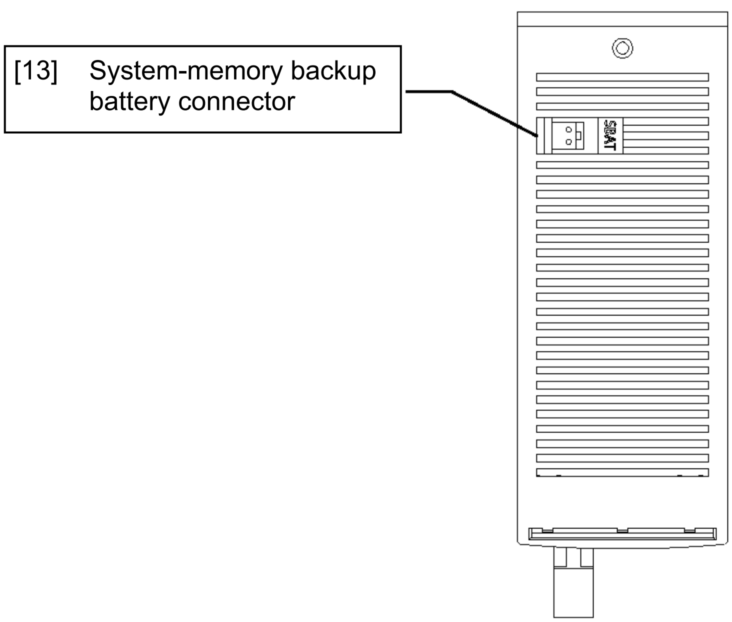


*1 For the 1-axis specification, [2], [5] and [6] are not installed and the front panel is masked.

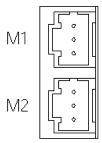
2.1.2 Down View



2.1.3 Top View



[1] Axis 1 motor connector (M1): This connector is used to connect the motor cable for axis 1.



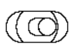
Motor Connector Specifications

Item	Specification	Remarks
Applicable connector	2.5-mm pitch connector, 3 pins	DF1E-3P-2.5DS (Hirose)
	Cable-end connector	DF1E-3S-2.5C (Hirose) Contact: DF1E-2022SC (Hirose)
Connector name	M1	
Maximum connection distance	20 m	
Connected cable	Motor cable	AWG22 X 3C

[2] Axis 2 motor connector (M2): This connector is used to connect the motor drive-source cable for axis 2. The specifications are the same as those of the axis 1 motor connector.

[3] Axis 1 brake-release switch (BK1): This switch is used to forcibly release the electromagnetic brake of the actuator constituting axis 1.

RLS (left) NOM (right)



Name	Description
RLS	Supply the power to the brake and forcibly release the brake.
NOM	Turn the brake ON/OFF using an internal sequence. Normally this switch is set to the “NOM” side.

[4] Axis 1 encoder/sensor connector (PG1): This connector is used to connect the encoder cable for axis 1. It connects the encoder cable of the actuator constituting axis 1.

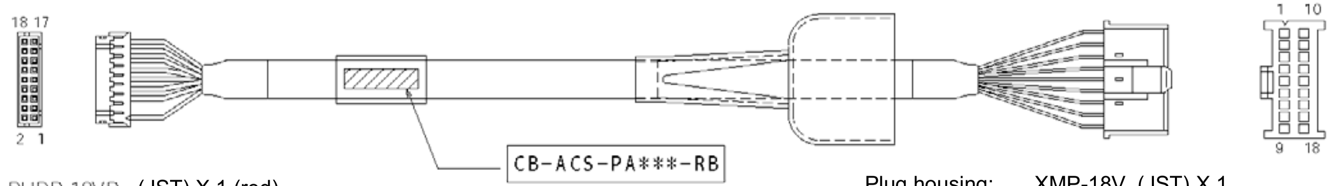
Encoder Connector Specifications

Item	Specification	Remarks
Applicable connector	2-mm pitch, double-row connector, 18 pins	S18B-PHDRS-B (JST)
	Cable-end connector	PHDR-18VR (JST) Contact: SPHD-001T-P0.5 (JST)
Connector name	PG1	
Maximum connection distance	20 m	
Connected cable	Motor cable	AWG26 X 7P Shielded

Encoder cable

Cable model: CB-ACS-PA＊＊＊-RB

Controller end



Actuator end

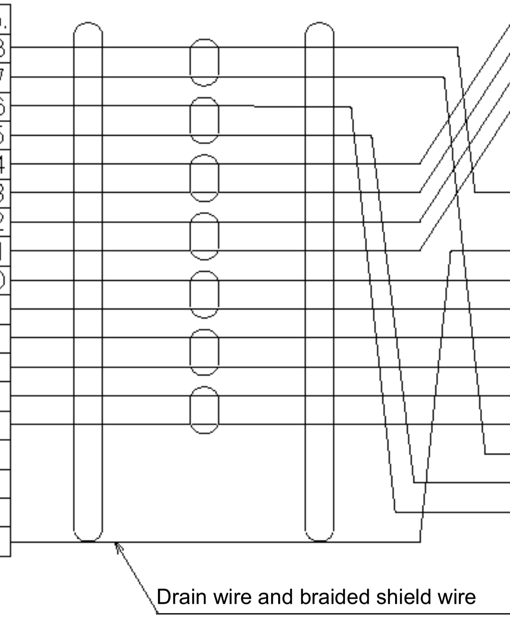
Housing: PHDR-18VR (JST) X 1 (red)
Contact: SPHD-001T-P0.5 (JST) X 15

Plug housing: XMP-18V (JST) X 1
Socket contact: BXA-001T-P0.6 (JST) X 15
Retainer: XMS-09V (JST) X 2

Wiring diagram

PHDR-18VR

Wire	Color	Signal	No.
AWG26 (pressure-welded)	White/Purple	LS+	18
	White/Gray	LS−	17
	Yellow	BK+	16
	Blue	BK−	15
	White/Blue	A+	14
	White/Yellow	A−	13
	White/Red	B+	12
	White/Black	B−	11
	Orange	SD/Z+	10
	Green	SD/Z−	9
	Purple	BAT+	8
	Gray	BAT−	7
	Red	VCC	6
	Black	GND	5
	—	—	4
	—	—	3
	—	—	2
	Drain	FG	1



XMP-18V

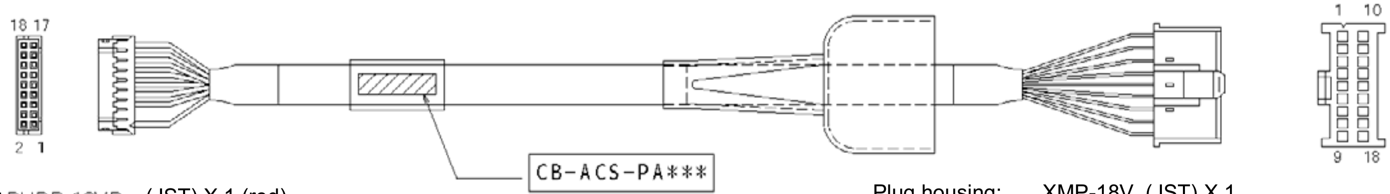
No.	Signal ABZ	Signal Serial	Color	Wire
1	A+	—	White/Blue	AWG26 (pressure-welded)
2	A−	—	White/Yellow	
3	B+	—	White/Red	
4	B−	—	White/Black	
5	—	—	—	
6	—	—	—	
7	LS+	LS+	White/Purple	
8	—	—	—	
9	FG	FG	Drain	
10	Z+	SD+	Orange	
11	Z−	SD−	Green	
12	—	BAT+	Purple	
13	/PS	BAT−	Gray	
14	VCC	VCC	Red	
15	GND	GND	Black	
16	LS−	LS−	White/Gray	
17	BK−	BK−	Blue	
18	BK+	BK+	Yellow	

(“White/blue” and other designations under “Color” indicate band color/insulator color.)

Encoder cable

Cable model: CB-ACS-PA＊＊＊

Controller end



Actuator end

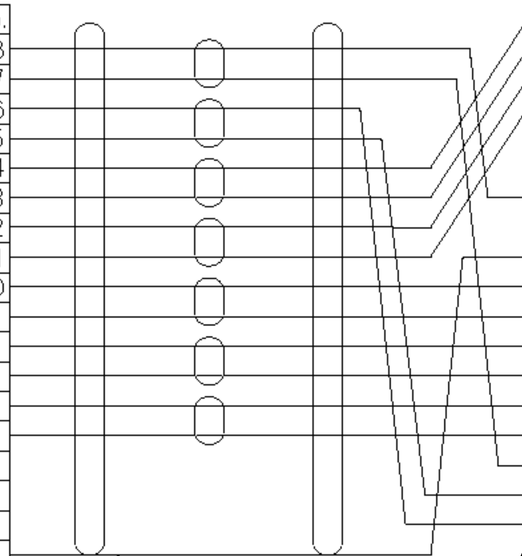
Housing: PHDR-18VR (JST) X 1 (red)
Contact: SPHD-001T-P0.5 (JST) X 15

Plug housing: XMP-18V (JST) X 1
Socket contact: BXA-001T-P0.6 (JST) X 15
Retainer: XMS-09V (JST) X 2

Wiring diagram

PHDR-18VR

Wire	Color	Signal	No.
AWG26 (pressure-welded)	Blue	LS+	18
	Orange	LS−	17
	Green	BK+	16
	Brown	BK−	15
	Gray	A+	14
	Red	A−	13
	Black	B+	12
	Yellow	B−	11
	Pink	SD/Z+	10
	Purple	SD/Z−	9
	White	BAT+	8
	Red/Blue	BAT−	7
	Orange/White	VCC	6
	Green/White	GND	5
	—	—	4
	—	—	3
	—	—	2
	Drain	FG	1



Drain wire and braided shield wire

XMP-18V

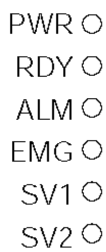
No.	Signal ABZ	Signal Serial	Color	Wire
1	A+	—	Gray	AWG26 (pressure-welded)
2	A−	—	Red	
3	B+	—	Black	
4	B−	—	Yellow	
5	—	—	—	
6	—	—	—	
7	LS+	LS+	Blue	
8	—	—	—	
9	FG	FG	Drain	
10	Z+	SD+	Pink	
11	Z−	SD−	Purple	
12	—	BAT+	White	
13	/PS	BAT−	Blue/Red	
14	VCC	VCC	Orange/White	
15	GND	GND	Green/White	
16	LS−	LS−	Orange	
17	BK−	BK−	Brown	
18	BK+	BK+	Green	

(“White/blue” and other designations under “Color” indicate band color/insulator color.)

[5] Axis 2 brake-release switch (BK2): This switch is used to forcibly release the electromagnetic brake of the actuator constituting axis 2. The specifications are the same as those of the axis 1 brake-release switch in [3].

[6] Axis 2 encoder/sensor connector (PG2): This connector is used to connect to the encoder cable for axis 2. The specifications are the same as those of the axis 1 encoder/sensor connector in [4].

[7] LED indicators: These indicators indicate the controller status.



Name	Color	Status when the LED is lit
PWR	Green	The controller has been started successfully and is receiving power.
RDY	Green	The controller is ready.
ALM	Orange	An alarm is present (an error of message level or higher has generated.)
EMG	Red	An emergency stop is being actuated.
SV1	Green	The servo for axis 1 is on.
SV2	Green	The servo for axis 2 is on.

[8] Panel unit connector: This connector is used to connect the optional panel unit.

[9] PIO connector: This 34-pin, flat DIO connector consists of 24 inputs and eight outputs.

Standard I/O Interface Specifications (key items)

Item	Description
Connector name	I/O
Applicable connector	Flat connector, 34 pins
Power supply	Power is supplied from connector pin Nos. 1 and 34.
Inputs	24 points (including general-purpose inputs and dedicated inputs)
Outputs	8 points (including general-purpose outputs and dedicated outputs)
Connected to	External PLC, sensor, etc.

I/O Interface List (Program mode)

Pin No.	Category	Port No.	Function	Cable color
1A		-	External power supply 24 V	1-Brown
1B	Input	016	Program specification (PRG No. 1)	1-Red
2A		017	Program specification (PRG No. 2)	1-Orange
2B		018	Program specification (PRG No. 4)	1-Yellow
3A		019	Program specification (PRG No. 8)	1-Green
3B		020	Program specification (PRG No. 10)	1-Blue
4A		021	Program specification (PRG No. 20)	1-Purple
4B		022	Program specification (PRG No. 40)	1-Gray
5A		023	Software reset (restart)	1-White
5B		000	Program start	1-Black
6A		001	General-purpose input	2-Brown
6B		002	General-purpose input	2-Red
7A		003	General-purpose input	2-Orange
7B		004	General-purpose input	2-Yellow
8A		005	General-purpose input	2-Green
8B		006	General-purpose input	2-Blue
9A		007	General-purpose input	2-Purple
9B		008	General-purpose input	2-Gray
10A		009	General-purpose input	2-White
10B		010	General-purpose input	2-Black
11A		011	General-purpose input	3-Brown
11B		012	General-purpose input	3-Red
12A		013	General-purpose input	3-Orange
12B		014	General-purpose input	3-Yellow
13A		015	General-purpose input	3-Green
13B	Output	300	Alarm output	3-Blue
14A		301	Ready output	3- Purple
14B		302	Emergency-stop output	3-Gray
15A		303	Emergency-stop output	3-White
15B		304	General-purpose output	3-Black
16A		305	General-purpose output	4-Brown
16B		306	General-purpose output	4-Red
17A		307	General-purpose output	4-Orange
17B	N		External power supply 0 V	4-Yellow

The above functions reflect the factory settings for the program mode.
These functions can be changed by changing the corresponding parameters.

[10] MANU/AUTO switch: This switch is used to specify the controller operation mode.

MANU (left)  AUTO (right)

	MANU	AUTO
Teaching pendant/PC software operation (When the teaching connector is used)	Possible	Not possible
PC software operation (when the USB connector is used)	Possible Note)	Not possible
Starting of an auto start program	Not possible	Possible

Note) When this switch is set to the “MANU” side and the USB connector is used, the servo cannot be turned on unless a dummy plug or teaching pendant is connected to the TP connector. When the USB connector is used, always keep a dummy plug or PC software cable connected to the TP plug while the controller is in use. (This is to cancel the disabled condition.)
If a dummy plug is used, always operate the controller in a condition where the emergency stop switch is within an easy reach.

[11] USB connector: This connector is used to connect the PC software and the controller via a USB cable.
Connector: USB connector B (XM7B-0442)
Connected to: USB cable
The maximum USB cable length is 5 m.

Note
- When the USB port is used, the USB driver contained in the “X-SEL PC Software IA-101-X-USB” CD-ROM must be installed by connecting all applicable controllers one by one. For the driver installation method, refer to the X-SEL PC Software Operation Manual.
- When the USB port is used, a dummy plug must be connected to the teaching connector [12].
  Dummy plug model: DP-3 (For ASEL-C)
  DP-4S (For ASEL-CS)

[12] Teaching connector (TP):

The teaching interface connects IAI's teaching pendant or a PC (PC software) to enable operation and setting of your equipment from the teaching pendant/PC.
Serial Communication (RS232C) is used for the interface.
connector. The signal level conforms to RS232C, and a desired baud rate (maximum 115.2 kbps) can be selected based on the program.
This connector can be used only when the mode switch is set to "MANU."

Interface Specifications of Teaching Serial Interface

Item	Description	Details
Connector	Cotroller side connector	ASEL-C Type: TX20A-26R-D2LT1-A1LHE (by JAE) ASEL-CS Type: HDR-EC26LFDT1-SLD+ (by HTK)
	Mating connector	ASEL-C Type: TX20A-26PH1-D2P1-D1E (by JAE) ASEL-CS Type: HDR-E26MSG1+ (by HTK)
Connector name	T.P.	Teaching connector
Baud rate	Up to 115.2 kbps	Half-duplex communication speeds of up to 115.2 kbps are supported.
Maximum wiring distance	10M	At 38.4 kbps
Interface standard	RS232C	
Connected unit	Dedicated teaching pendant	PC software or teaching pendant
Connection cable		Dedicated cable
Power supply	5 VDC or 24 VDC	A multi-fuse (MF-R090) is installed to protect each line against short current (the fuse will trip with currents of between 1.1 A and 2.2 A).
Protocol	X-SEL teaching protocol	The connector supports the X-SEL teaching pendant interface protocol.
Emergency-stop control	Series emergency-stop relay drive (24 V)	An emergency-stop relay drive line is provided in the interface connector. This line is connected in series with other emergency-stop contact.
Enabling control	Enable switch line (24 V)	A line for connecting an enable switch is provided as an operator interlock..

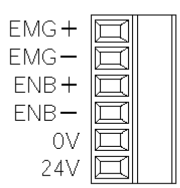
Teaching pendant & dedicated communication cable connector

Item	Specification			Remarks
Terminal assignments	Pin No.	I/O	Signal name	
	1		SG	Signal ground
	2	Out	EMGS	Emergency-stop status
	3	Out	VCC	Power output (Standard IA-T-X/XD power supply (5 V))
	4	In	DTR	Data terminal ready (Shorted to DSR)
	5		NC	Not connected
	6		NC	Not connected
	7		NC	Not connected
	8	Out	RSVVCC	Power output (ANSI compliant IA-T-XA power supply (24 V))
	9	In	EMGIN	Emergency-stop contact output, negative
	10	Out	RSVVCC	Power output (ANSI compliant IA-T-XA power supply (24 V))
	11		NC	Not connected
	12	Out	EMGOUT2	Emergency-stop contact output, positive
	13	Out	RTS	Request to send (Not used; fixed to 0 V)
	14	In	CTS	Clear to send (Not used / Used as the TP-connection detection terminal)
	15	Out	TXD	Transmitted data
	16	In	RXD	Received data
	17	Out	DSR	Data set ready (Shorted to DTR)
	18		NC	Not connected
	19		NC	Not connected
	20		NC	Not connected
	21		NC	Not connected
	22		NC	Not connected
	23	In	ENBTB	Enable input
	24	Out	ENBVCC	Enable drive power (24V)
	25		NC	Not connected (Reserved by ENBTBX2)
	26		SG	Signal ground

[13] System-memory backup battery connector: This connector is used to install the system-memory backup battery.

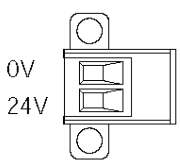
[14] Control power & system I/O connector: This connector is used to input the 24-VDC control power and connect the emergency stop switch and enable switch.
The power supply connected to this connector is used for the controller internal power, brake power, and so on, and not used as the motor drive source.
The 0-V input is connected to the ground for the controller's internal power supply and is not insulated.

Item	Specification		Remarks
Applicable connector	3.5 mm, 2-piece COMBICON, 6 pins		MC1.5/6-G-3.5 by Phoenix Contact
	Cable-end connector		MC1.5/6-ST-3.5 by Phoenix Contact
	Applicable wire size		AWG20 ~ 16 (0.5 ~ 1.25 sq)
	Recommended stripped-wire length		7 mm
Connector name	CP EMG ENB		
Input voltage	24 VDC + 10%/-10%		
Maximum input current	1.2 A		
Terminal assignments	No.	Name	Function
	1	EMG+	Emergency stop switch +
	2	EMG-	Emergency stop switch -
	3	ENB+	Enable switch +
	4	ENB-	Enable switch -
	5	0V	Control power input ground (Connected to the internal ground)
	6	24V	Control power input +24 V



[15] Regenerative resistor connector: This connector is used to connect a regenerative resistor when the built-in regenerative resistor alone cannot provide enough capacity in high-acceleration/high-load operation, etc. This connector is not normally used with the ASEL controller.

[16] Motor power connector: This connector is used to input the 24-VDC motor power.
The power supply connected to this connector is used as the dedicated motor drive source.
Since the controller has a built-in drive-source cutoff relay, the power supply to the motor will be cut off internally if an emergency stop is actuated or other abnormality occurs.
Although the motor power and control power are input independently, the 0-V terminals of both are connected inside the controller. They are also connected to the ground for the controller's internal power supply and are not insulated.



Item	Specification		Remarks
Applicable connector	5.08 mm, 2-piece COMBICON, 2 pins		MSTB2.5/2-GF-5.08 by Phoenix Contact
	Cable-end connector		MSTB2.5/2-STF-5.08 by Phoenix Contact
	Applicable wire size		AWG20 ~ 14 (0.5 ~ 2.0 sq)
	Recommended stripped-wire length		7 mm
Connector name	MP		
Input voltage	24 VDC ± 10%		
Maximum input current	10.2 A		5.1 A per axis
Terminal assignments	No.	Name	Function
	1	0V	Motor power input ground (Connected to the internal ground)
	2	24V	Motor power input +24 V

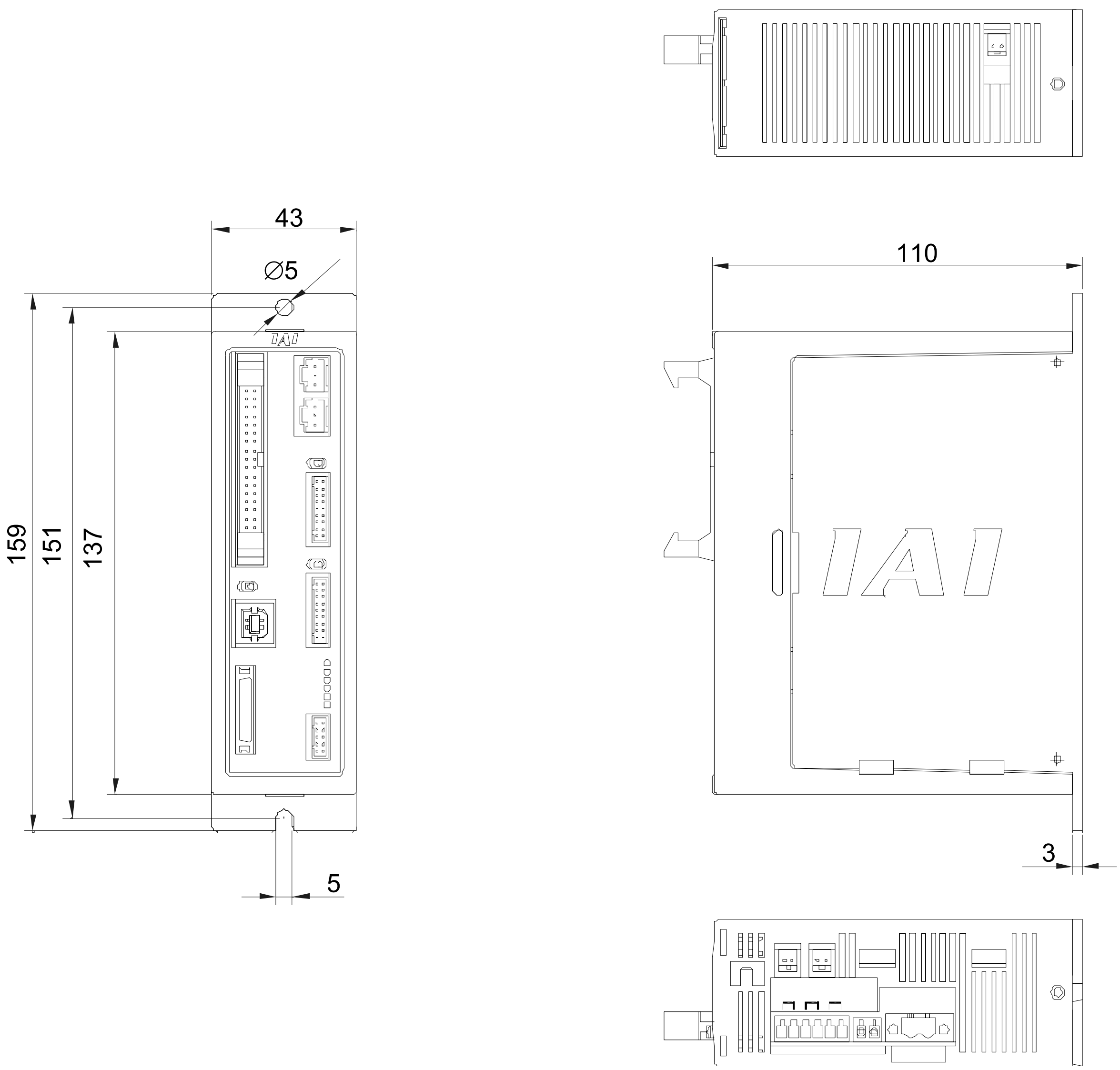
[17] Axis 1 absolute-data backup battery connector: This connector is used to connect the absolute-data backup battery for axis 1. (This connector is required only if your controller is of absolute-encoder specification.)

[18] Axis 2 absolute-data backup battery connector: This connector is used to connect the absolute-data backup battery for axis 2. (This connector is required only if your controller is of absolute-encoder specification.)

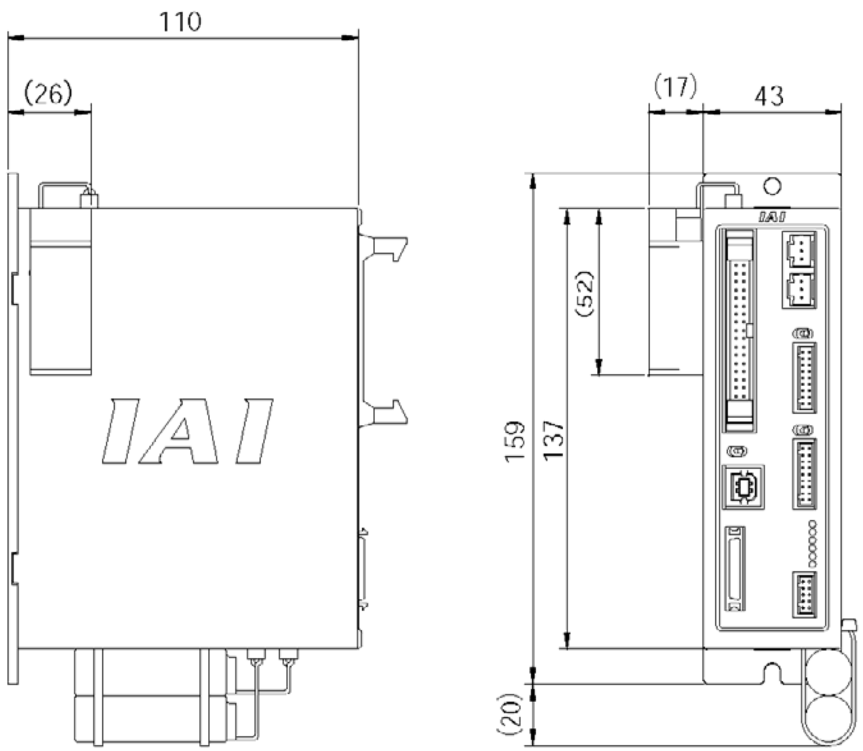
Chapter 3 Installation and Wiring

1. External Dimensions

(1) 2-axis specification
(The same external dimensions also apply to the 1-axis specification.)



(2) 2-axis specification with battery



As for the use environment, this product can be used in an environment of pollution degree 2[*1] or equivalent.

*1 Pollution degree 2: Normally only non-conductive pollutants exist, which are expected to be temporarily conductive due to condensation. (EN60947-5-1)

2. Installation Environment

(1) When installing and wiring the controller, do not block the ventilation holes provided for cooling. (Insufficient ventilation will not only prevent the product from functioning fully, but it may also result in failure.)
(2) Prevent foreign matter from entering the controller through the ventilation holes. Since the controller is not designed as dustproof or waterproof (oilproof), avoid using it in a dusty place or place subject to oil mist or splashed cutting fluid.
(3) Do not expose the controller to direct sunlight or radiant heat from a high heat source such as a heat-treating furnace.
(4) Use the controller in a non-condensing environment free from corrosive or inflammable gases.
(5) Use the controller in an environment where it will not receive external vibration or impact.
(6) Prevent electrical noise from entering the controller or its cables.

Environmental Condition of Controller

Item	Specification and description
Operating temperature range	0 ~ 40°C
Operating humidity range	10% ~ 95% (Non-condensing; conforming to JIS C3502 RH-2)
Storage temperature range	-25°C ~ 70°C (Excluding the battery)
Maximum operating altitude	2000 m
Protection class	IP20
Vibration	$10 \leq f < 57$: 0.035 mm (continuous), 0.075 mm (intermittent) $57 \leq f \leq 150$: 4.9 m/s^2 (continuous), 9.8 m/s^2 (intermittent) X, Y and Z directions
Impact	147 mm/s^2, 11 ms, half-sine pulse, 3 times each in X, Y and Z directions

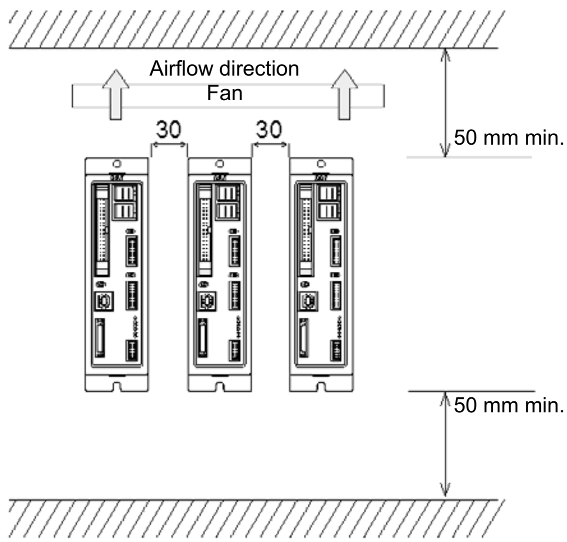
3. Heat Radiation and Installation

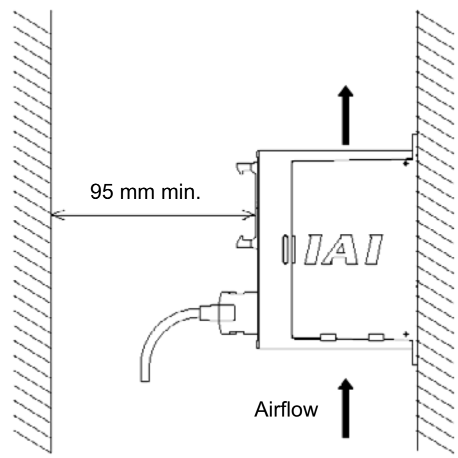
Design the control panel size, controller layout and cooling method so that the ambient temperature around the controller will be kept at or below 40°C.

Install the controller vertically on a wall, as illustrated below. The controller will be cooled by natural convection. Be sure to install the controller in the aforementioned direction and provide a minimum clearance of 50 mm above and below the controller.

If multiple controllers are to be installed side by side, providing additional suction fans on top of the controllers will help maintain a uniform ambient temperature.

Provide a minimum clearance of 95 mm between the front side of the controller and a wall (enclosure).



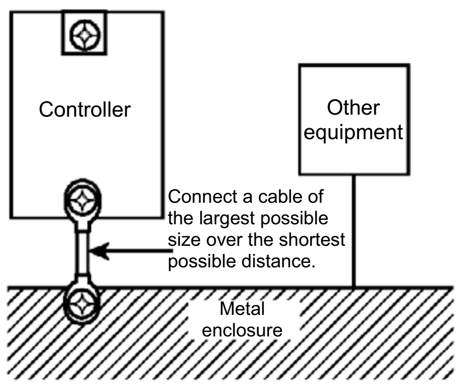


If multiple controllers are to be connected on top of one another, prevent the controller above from taking in the exhaust air from the controller below.

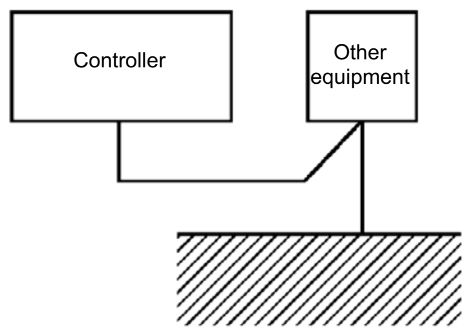
4. Noise Control Measures and Grounding

The ASEL controller has no dedicated terminal to connect the FG to ground. Accordingly, provide grounding using the controller mounting screw.

[1] Provide dedicated Class D grounding. The grounding wire should have a size of 2.0 to 5.5 mm^2 or larger.



Class D grounding Proper grounding



Avoid using this method.

[2] Notes on wiring method

Use twisted wires for the 24-VDC external power supply.

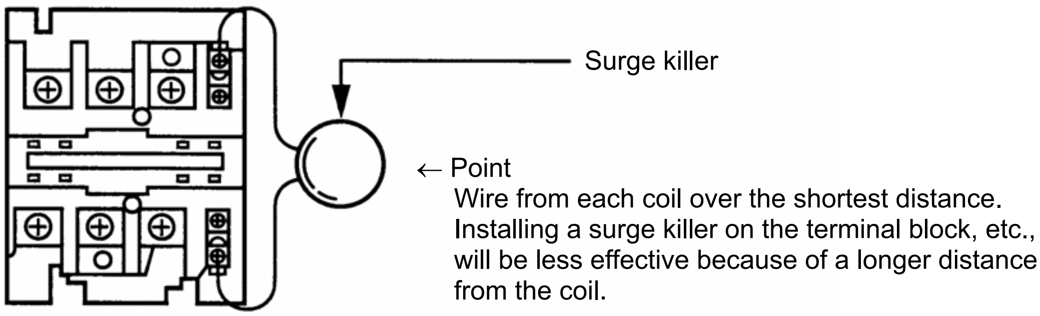
Wire the controller cables separately from lines creating a strong electric field such as power circuit lines (by not bundling them together or placing in the same cable duct).

If you wish to extend the motor cable or encoder cable beyond the length of each supplied cable, please contact IAI's Technical Service Section or Sales Engineering Section.

(3) Noise sources and noise elimination
There are many noise sources, but solenoid valves, magnet switches and relays are of particular concern when building a system. Noise from these parts can be eliminated using the measures specified below:

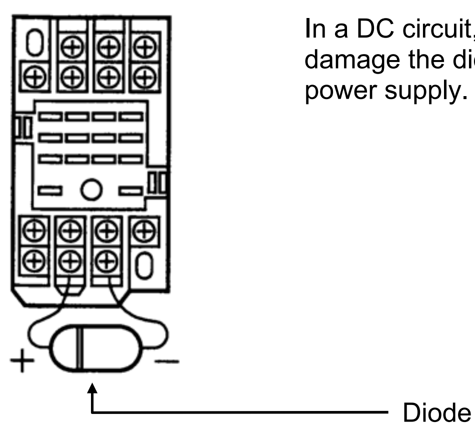
[1] AC solenoid valve, magnet switch, relay

Measure --- Install a surge killer in parallel with the coil.



[2] DC solenoid valve, magnet switch, relay

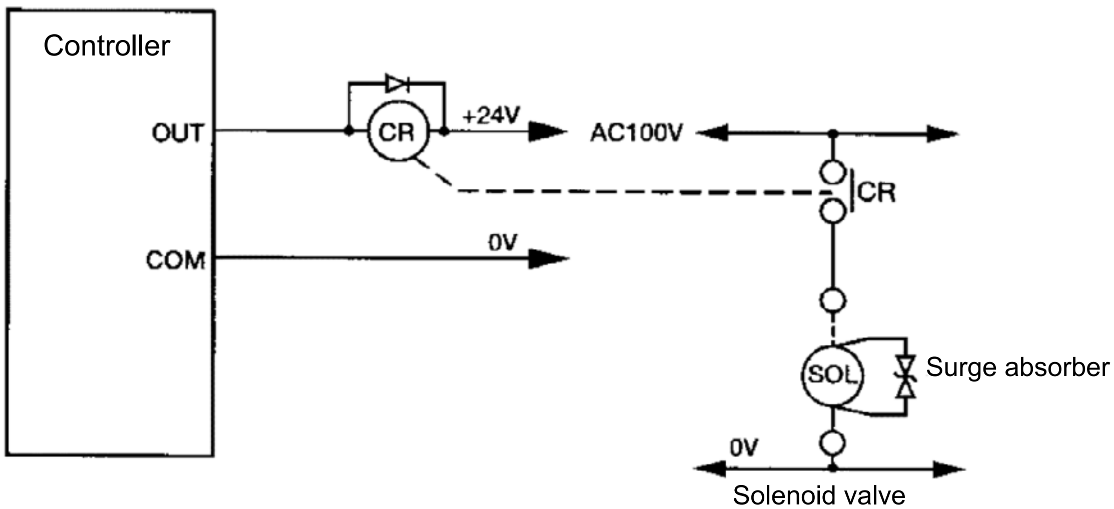
Measure --- Install a diode in parallel with the coil. Determine the diode capacity in accordance with the load capacity.



In a DC circuit, connecting a diode in reversed polarity will damage the diode, internal parts of the controller and DC power supply. Exercise due caution.

The above noise elimination measures are particularly important when a 24-VDC relay is driven directly by a controller output and there is also a 100-VAC solenoid valve, etc.

Reference Circuit Diagram



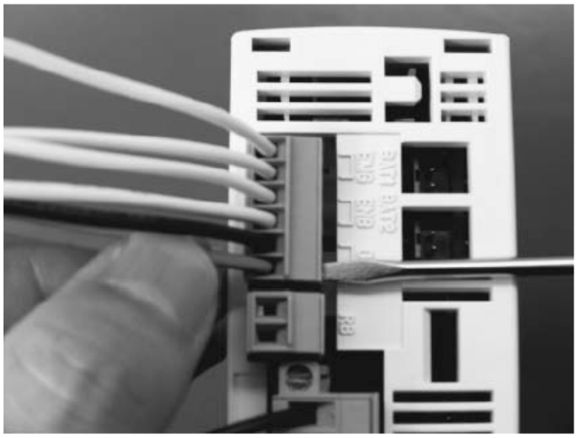
5. Supply Voltage

The supply voltage to the controller is 24 VDC ± 10%.
The power-supply current varies depending on the number of axes, as shown below.

	1-axis specification	2-axis specification
[1] Control power-supply current	1.2 A	
[2] Rated motor power-input current	1.7 A	3.4 A
[3] Maximum motor power-input current	5.1 A	10.2 A
[4] Rated current ([1] + [2])	2.9 A	4.6 A
[5] Maximum current ([1] + [3])	6.3 A	11.4 A

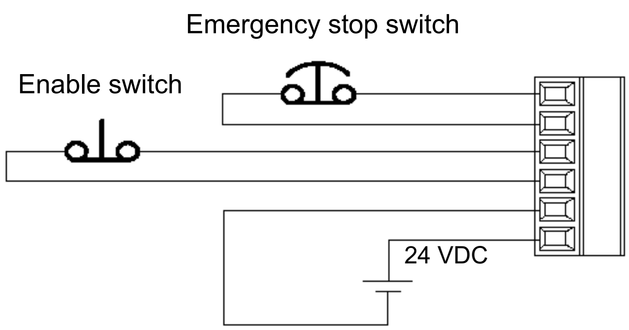
6. Wiring

6.1 Wiring the Control Power Supply, Emergency Stop Switch and Enable Switch



As shown to the left, insert the stripped end of each cable into the control power & system I/O connector, and tighten the screws with a screwdriver.

Recommended cable size: 0.75 mm^2 (AWG18)
Recommended stripped-wire length: 7 mm

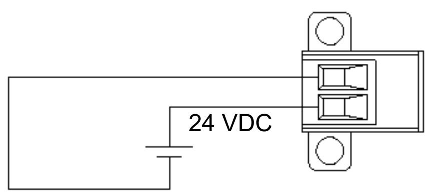


6.2 Wiring the Motor Power Cables

As shown to the left, insert the stripped end of each cable into the motor power connector, and tighten the screws with a screwdriver.

Recommended cable size: 2 mm^2 (AWG14)
Recommended stripped-wire length: 7 mm

As shown to the left, tighten the screws to affix the connector.

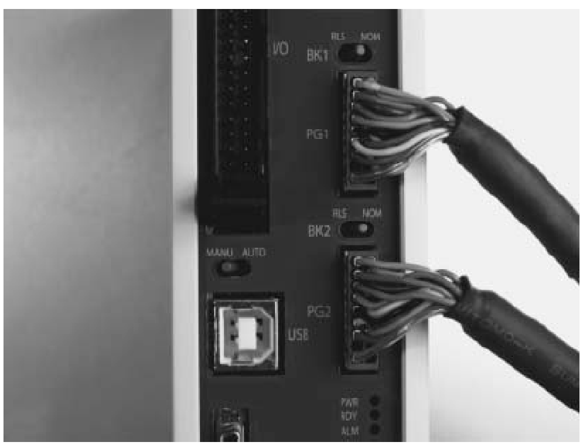


6.3 Connecting the Actuator

6.3.1 Connecting the Motor Cable (M1/M2)

Connect the motor cable from the actuator to the applicable motor connector on the front face of the controller.

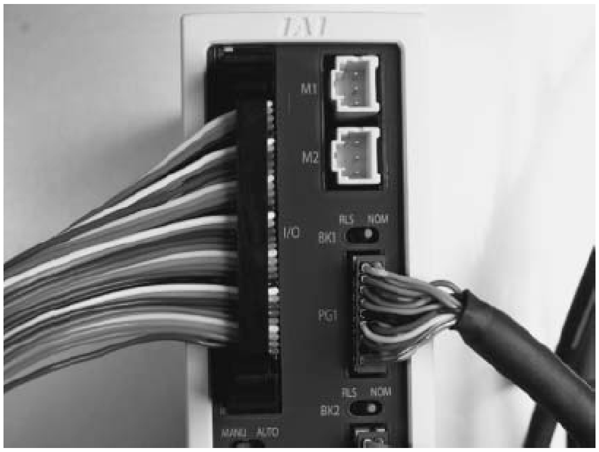
6.3.2 Connecting the Encoder Cable (PG1/PG2)



Connect the encoder cable from the actuator to the applicable encoder connector on the front face of the controller.

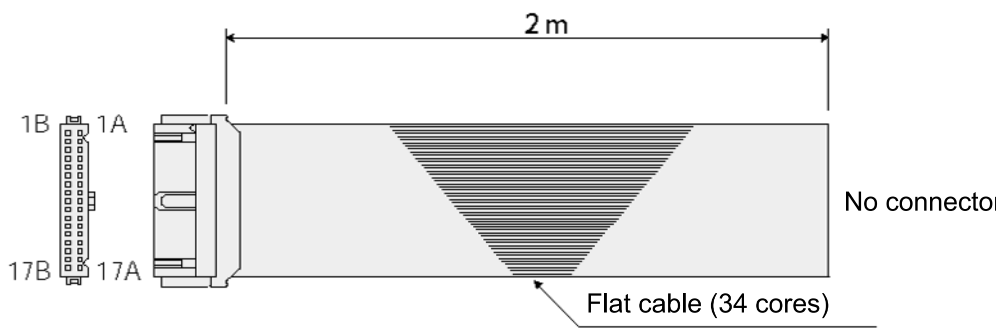
Caution:	With the absolute specification, disconnect the absolute-data backup battery connector before connecting the encoder cable. Connect the absolute-data backup battery connector after turning on the main power.

6.4 Connecting the PIO Cable (I/O)



Connect the supplied flat cable. Connect the opposite end (open end without connector) of the cable to a desired peripheral (host PLC, etc.).

I/O flat cable (supplied): Model CB-DS-P10020



No.	Color	Wire	No.	Color	Wire
1A	Brown 1	Flat cable, pressure-welded	9B	Gray 2	Flat cable, pressure-welded
1B	Red 1		10A	White 2	
2A	Orange 1		10B	Black 2	
2B	Yellow 1		11A	Brown-3	
3A	Green 1		11B	Red 3	
3B	Blue 1		12A	Orange 3	
4A	Purple 1		12B	Yellow 3	
4B	Gray 1		13A	Green 3	
5A	White 1		13B	Blue 3	
5B	Black 1		14A	Purple 3	
6A	Brown-2		14B	Gray 3	
6B	Red 2		15A	White 3	
7A	Orange 2		15B	Black 3	
7B	Yellow 2		16A	Brown-4	
8A	Green 2		16B	Red 4	
8B	Blue 2		17A	Orange 4	
9A	Purple 2		17B	Yellow 4	

6.4.1 I/O Connection Diagram

(1) NPN specification (Program mode)

0V

+24V

Pin No.	Category	Port No.	Function	Cable color
1A	P24		External power supply 24 V	1 - Brown
1B	Input	016	Program specification (PRG No. 1)	1 - Red
2A		017	Program specification (PRG No. 2)	1 - Orange
2B		018	Program specification (PRG No. 4)	1 - Yellow
3A		019	Program specification (PRG No. 8)	1 - Green
3B		020	Program specification (PRG No. 10)	1 - Blue
4A		021	Program specification (PRG No. 20)	1 - Purple
4B		022	Program specification (PRG No. 40)	1 - Gray
5A		023	Software reset (restart)	1 - White
5B		000	Program start	1 - Black
6A		001	General-purpose input	2 - Brown
6B		002	General-purpose input	2 - Red
7A		003	General-purpose input	2 - Orange
7B		004	General-purpose input	2 - Yellow
8A		005	General-purpose input	2 - Green
8B		006	General-purpose input	2 - Blue
9A		007	General-purpose input	2 - Purple
9B		008	General-purpose input	2 - Gray
10A		009	General-purpose input	2 - White
10B		010	General-purpose input	2 - Black
11A		011	General-purpose input	3 - Brown
11B		012	General-purpose input	3 - Red
12A		013	General-purpose input	3 - Orange
12B		014	General-purpose input	3 - Yellow
13A		015	General-purpose input	3 - Green

Pin No.	Category	Port No.	Function	Cable color
13B	Output	300	Alarm output	3 - Blue
14A		301	Ready output	3 - Purple
14B		302	General-purpose output	3 - Gray
15A		303	General-purpose output	3 - White
15B		304	General-purpose output	3 - Black
16A		305	General-purpose output	4 - Brown
16B		306	General-purpose output	4 - Red
17A		307	General-purpose output	4 - Orange
17B	N		External power supply 0 V	4 - Yellow

The above functions reflect the factory settings.

(2) PNP specification (Program mode)

+24V

0V

Pin No.	Category	Port No.	Function	Cable color
1A	P24		External power supply 24 V	1 - Brown
1B	Input	016	Program specification (PRG No. 1)	1 - Red
2A		017	Program specification (PRG No. 2)	1 - Orange
2B		018	Program specification (PRG No. 4)	1 - Yellow
3A		019	Program specification (PRG No. 8)	1 - Green
3B		020	Program specification (PRG No. 10)	1 - Blue
4A		021	Program specification (PRG No. 20)	1 - Purple
4B		022	Program specification (PRG No. 40)	1 - Gray
5A		023	Software reset (restart)	1 - White
5B		000	Program start	1 - Black
6A		001	General-purpose input	2 - Brown
6B		002	General-purpose input	2 - Red
7A		003	General-purpose input	2 - Orange
7B		004	General-purpose input	2 - Yellow
8A		005	General-purpose input	2 - Green
8B		006	General-purpose input	2 - Blue
9A		007	General-purpose input	2 - Purple
9B		008	General-purpose input	2 - Gray
10A		009	General-purpose input	2 - White
10B		010	General-purpose input	2 - Black
11A		011	General-purpose input	3 - Brown
11B		012	General-purpose input	3 - Red
12A		013	General-purpose input	3 - Orange
12B		014	General-purpose input	3 - Yellow
13A		015	General-purpose input	3 - Green

Pin No.	Category	Port No.	Function	Cable color
13B	Output	300	Alarm output	3 - Blue
14A		301	Ready output	3 - Purple
14B		302	General-purpose output	3 - Gray
15A		303	General-purpose output	3 - White
15B		304	General-purpose output	3 - Black
16A		305	General-purpose output	4 - Brown
16B		306	General-purpose output	4 - Red
17A		307	General-purpose output	4 - Orange
17B	N		External power supply 0 V	4 - Yellow

The above functions reflect the factory settings.

(3) NPN specification (Positioner mode)

0V +24V

Pin No.	Category	Port No.	Positioner mode: Standard mode	Positioner mode: Product switching mode	Positioner mode: 2-axis independent mode	Positioner mode: Teaching mode	Positioner mode: DC-S-C1 compatible mode	Cable color
1A	P24		24-V input					1 - Brown
1B	Input	16	Position input 10	Input 10	Position input 7	Axis 1 jog-	Position No. 1000 input	1 - Red
2A	Input	17	Position input 11	Input 11	Position input 8	Axis 2 jog+	—	1 - Orange
2B	Input	18	Position input 12	Input 12	Position input 9	Axis 2 jog-	—	1 - Yellow
3A	Input	19	Position input 13	Input 13	Position input 10	Inching (0.01 mm)	—	1 - Green
3B	Input	20	—	Input 14	Position input 11	Inching (0.1 mm)	—	1 - Blue
4A	Input	21	—	Input 15	Position input 12	Inching (0.5 mm)	—	1 - Purple
4B	Input	22	—	Input 16	Position input 13	Inching (1 mm)	—	1 - Gray
5A	Input	23	Error reset	Error reset	Error reset	Error reset	CPU reset	1 - White
5B	Input	0	Start	Start	Axis 1 start	Start	Start	1 - Black
6A	Input	1	Home return	Home return	Home return	Sarvo ON	Pause	2 - Brown
6B	Input	2	Servo ON	Servo ON	Axis 1 servo ON	*Pause	Cancellation	2 - Red
7A	Input	3	Push motion	Push motion	*Axis 1 pause	Position input 1	Interpolation setting	2 - Orange
7B	Input	4	*Pause	*Pause	*Axis 1 cancellation	Position input 2	Position No. 1 input	2 - Yellow
8A	Input	5	*Cancellation	*Cancellation	Axis 2 start	Position input 3	Position No. 2 input	2 - Green
8B	Input	6	Interpolation	Interpolation	Axis 2 home return	Position input 4	Position No. 4 input	2 - Blue
9A	Input	7	Position input 1	Input 1	Axis 2 servo ON	Position input 5	Position No. 8 input	2 - Purple
9B	Input	8	Position input 2	Input 2	*Axis 2 pause	Position input 6	Position No. 10 input	2 - Gray
10A	Input	9	Position input 3	Input 3	*Axis 2 cancellation	Position input 7	Position No. 20 input	2 - White
10B	Input	10	Position input 4	Input 4	Position input 1	Position input 8	Position No. 40 input	2 - Black
11A	Input	11	Position input 5	Input 5	Position input 2	Position input 9	Position No. 80 input	3 - Brown
11B	Input	12	Position input 6	Input 6	Position input 3	Position input 10	Position No. 100 input	3 - Red
12A	Input	13	Position input 7	Input 7	Position input 4	Position input 11	Position No. 200 input	3 - Orange
12B	Input	14	Position input 8	Input 8	Position input 5	Teaching mode specification	Position No. 400 input	3 - Yellow
13A	Input	15	Position input 9	Input 9	Position input 6	Axis 1 jog+	Position No. 800 input	3 - Green

Pin No.	Category	Port No.	Positioner mode: Standard mode	Positioner mode: Product switching mode	Positioner mode: 2-axis independent mode	Positioner mode: Teaching mode	Positioner mode: DC-S-C1 compatible mode	Cable color
13B	Output	300	*Alarm	*Alarm	*Alarm	*Alarm	*Alarm	3 - Blue
14A	Output	301	Ready	Ready	Ready	Ready	Ready	3 - Purple
14B	Output	302	Positioning complete	Positioning complete	Axis 1 positioning complete	Positioning complete	Positioning complete	3 - Gray
15A	Output	303	Home return complete	Home return complete	Axis 1 home return complete	Home return complete	—	3 - White
15B	Output	304	Servo ON output	Servo ON output	Axis 1 servo ON	Servo ON output	—	3 - Black
16A	Output	305	Push motion complete	Push motion complete	Axis 2 positioning complete		—	4 - Brown
16B	Output	306	System battery error	System battery error	Axis 2 home return complete	System battery error	System battery error	4 - Red
17A	Output	307	Absolute battery error	Absolute battery error	Axis 2 servo ON	Absolute battery error	Absolute battery error	4 - Orange
17B	N		0-V input					4 - Yellow

*: Contact B (Always ON)

(4) PNP specification (Positioner mode)

+24V

0V

Pin No.	Category	Port No.	Positioner mode					Cable color
			Standard mode	Product switching mode	2-axis independent mode	Teaching mode	DC-S-C1 compatible mode	
1A	P24		24-V input					1 - Brown
1B	Input	16	Position input 10	Input 10	Position input 7	Axis 1 jog-	Position No. 1000 input	1 - Red
2A		17	Position input 11	Input 11	Position input 8	Axis 2 jog+	—	1 - Orange
2B		18	Position input 12	Input 12	Position input 9	Axis 2 jog-	—	1 - Yellow
3A		19	Position input 13	Input 13	Position input 10	Inching (0.01 mm)	—	1 - Green
3B		20	—	Input 14	Position input 11	Inching (0.1 mm)	—	1 - Blue
4A		21	—	Input 15	Position input 12	Inching (0.5 mm)	—	1 - Purple
4B		22	—	Input 16	Position input 13	Inching (1 mm)	—	1 - Gray
5A		23	Error reset	Error reset	Error reset	Error reset	CPU reset	1 - White
5B		0	Start	Start	Axis 1 start	Start	Start	1 - Black
6A		1	Home return	Home return	Home return	Sarvo ON	Pause	2 - Brown
6B		2	Servo ON	Servo ON	Axis 1 servo ON	*Pause	Cancellation	2 - Red
7A		3	Push motion	Push motion	*Axis 1 pause	Position input 1	Interpolation setting	2 - Orange
7B		4	*Pause	*Pause	*Axis 1 cancellation	Position input 2	Position No. 1 input	2 - Yellow
8A		5	*Cancellation	*Cancellation	Axis 2 start	Position input 3	Position No. 2 input	2 - Green
8B		6	Interpolation	Interpolation	Axis 2 home return	Position input 4	Position No. 4 input	2 - Blue
9A		7	Position input 1	Input 1	Axis 2 servo ON	Position input 5	Position No. 8 input	2 - Purple
9B		8	Position input 2	Input 2	*Axis 2 pause	Position input 6	Position No. 10 input	2 - Gray
10A		9	Position input 3	Input 3	*Axis 2 cancellation	Position input 7	Position No. 20 input	2 - White
10B		10	Position input 4	Input 4	Position input 1	Position input 8	Position No. 40 input	2 - Black
11A		11	Position input 5	Input 5	Position input 2	Position input 9	Position No. 80 input	3 - Brown
11B		12	Position input 6	Input 6	Position input 3	Position input 10	Position No. 100 input	3 - Red
12A		13	Position input 7	Input 7	Position input 4	Position input 11	Position No. 200 input	3 - Orange
12B		14	Position input 8	Input 8	Position input 5	Teaching mode specification	Position No. 400 input	3 - Yellow
13A		15	Position input 9	Input 9	Position input 6	Axis 1 jog+	Position No. 800 input	3 - Green

Pin No.	Category	Port No.	Positioner mode					Cable color
			Standard mode	Product switching mode	2-axis independent mode	Teaching mode	DC-S-C1 compatible mode	
13B	Output	300	*Alarm	*Alarm	*Alarm	*Alarm	*Alarm	3 - Blue
14A		301	Ready	Ready	Ready	Ready	Ready	3 - Purple
14B		302	Positioning complete	Positioning complete	Axis 1 positioning complete	Positioning complete	Positioning complete	3 - Gray
15A		303	Home return complete	Home return complete	Axis 1 home return complete	Home return complete	—	3 - White
15B		304	Servo ON output	Servo ON output	Axis 1 servo ON	Servo ON output	—	3 - Black
16A		305	Push motion complete	Push motion complete	Axis 2 positioning complete		—	4 - Brown
16B		306	System battery error	System battery error	Axis 2 home return complete	System battery error	System battery error	4 - Red
17A		307	Absolute battery error	Absolute battery error	Axis 2 servo ON	Absolute battery error	Absolute battery error	4 - Orange
17B	N		0-V input					4 - Yellow

*: Contact B (Always ON)

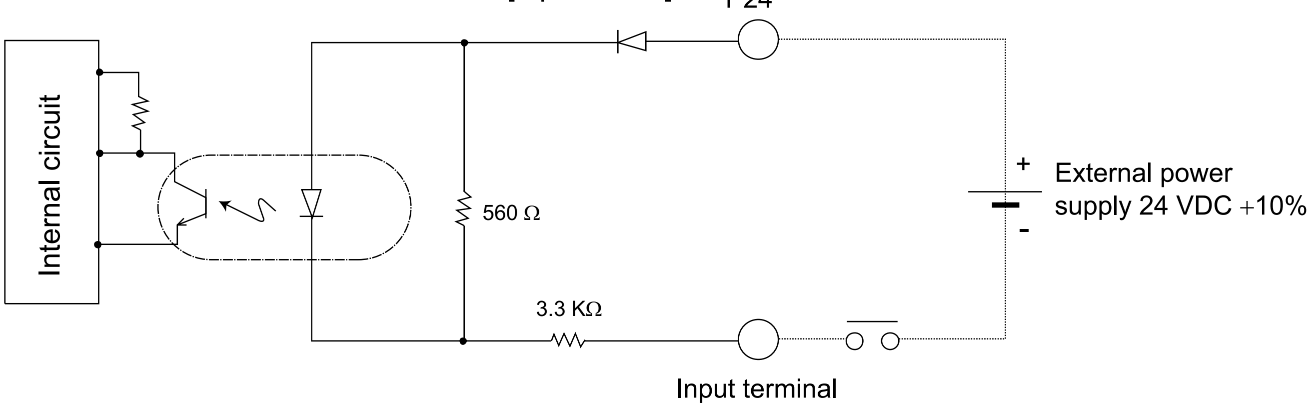
6.5 External I/O Specifications

6.5.1 NPN Specification

(1) Input part

External Input Specifications (NPN Specification)

Item	Specification
Input voltage	24 VDC ±10%
Input current	7 mA per circuit
ON/OFF voltage	ON voltage --- 16.0 VDC min. OFF voltage --- 5.0 VDC max.
Insulation method	Photocoupler insulation
External devices	[1] No-voltage contact (minimum load of approx. 5 VDC/1 mA) [2] Photoelectric/proximity sensor (NPN type) [3] Sequencer transistor output (open-collector type) [4] Sequencer contact output (minimum load of approx. 5 VDC/1 mA)

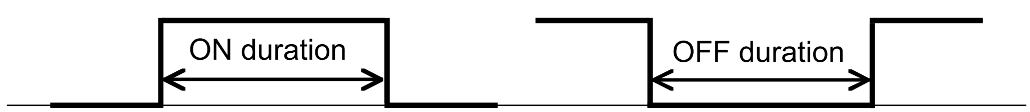


* P24: I/O interface pin No. 1

Caution

If a non-contact circuit is connected externally, malfunction may result from leakage current. Use a circuit in which leakage current in a switch-off state does not exceed 1 mA.

◎ ASEL controller's input signal



At the default settings, the system recognizes the ON/OFF durations of input signals if they are approx. 4 msec or longer. The ON/OFF duration settings can also be changed using I/O parameter No. 20 (input filtering frequency).

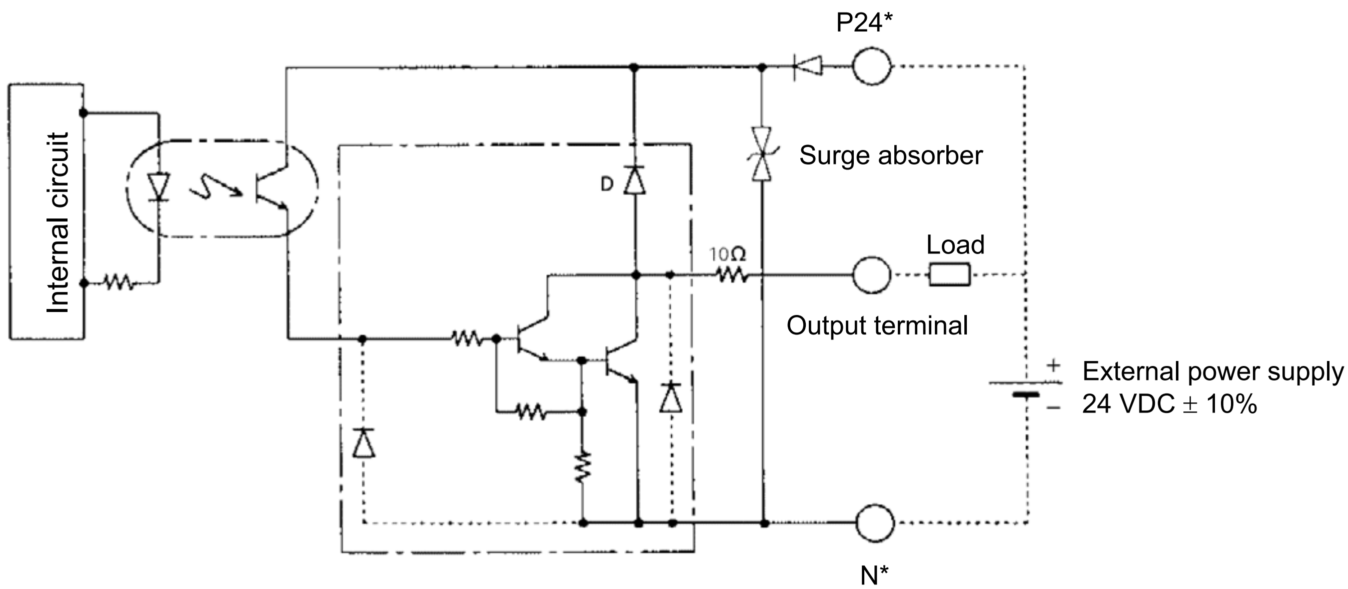
(2) Output part

External Output Specifications (NPN Specification)

Item	Specification	
Load voltage	24 VDC	TD62084 (or equivalent)
Maximum load current	100 mA per point, 400 mA per 8 ports Note)	
Leakage current	0.1 mA max. per point	
Insulation method	Photocoupler insulation	
External devices	[1] Miniature relay [2] Sequencer input unit	

Note) 400 mA is the maximum total load current of output port Nos. 300 to 307.

[Output circuit]



* P24: I/O interface pin No. 1A
* N: I/O interface pin No. 17B

Caution

In the event that the load is short-circuited or current exceeding the maximum load current is input, the overcurrent protection circuit will be actuated to cut off the circuit. However, give due consideration to the circuit connection layout to prevent short-circuit or overcurrent.

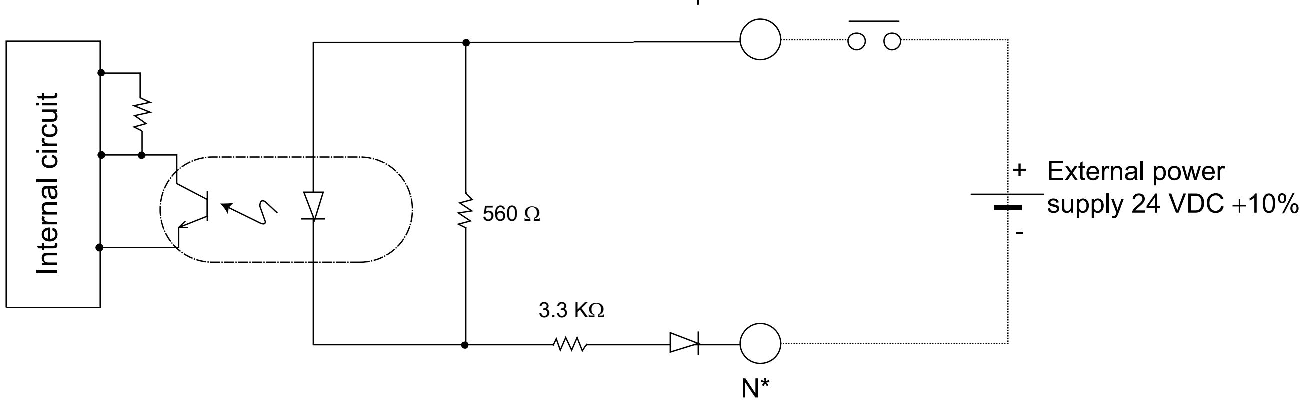
6.5.2 PNP Specification

(1) Input part

External Input Specifications (PNP Specification)

Item	Specification
Input voltage	24 VDC ±10%
Input current	7 mA per circuit
ON/OFF voltage	ON voltage --- 8 VDC max. OFF voltage --- 19 VDC min.
Insulation method	Photocoupler insulation
External devices	[1] No-voltage contact (minimum load of approx. 5 VDC/1 mA) [2] Photoelectric/proximity sensor (PNP type) [3] Sequencer transistor output (open-collector type) [4] Sequencer contact output (minimum load of approx. 5 VDC/1 mA)

[Input circuit]

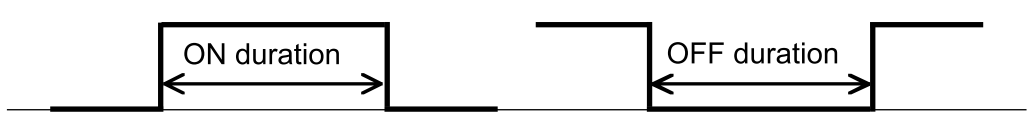


* N: I/O interface pin No. 17B

Caution

If a non-contact circuit is connected externally, malfunction may result from leakage current. Use a circuit in which leakage current in a switch-off state does not exceed 1 mA.

◎ ASEL controller's input signal



At the default settings, the system recognizes the ON/OFF durations of input signals if they are approx. 4 msec or longer. The ON/OFF duration settings can also be changed using I/O parameter No. 20 (input filtering frequency).

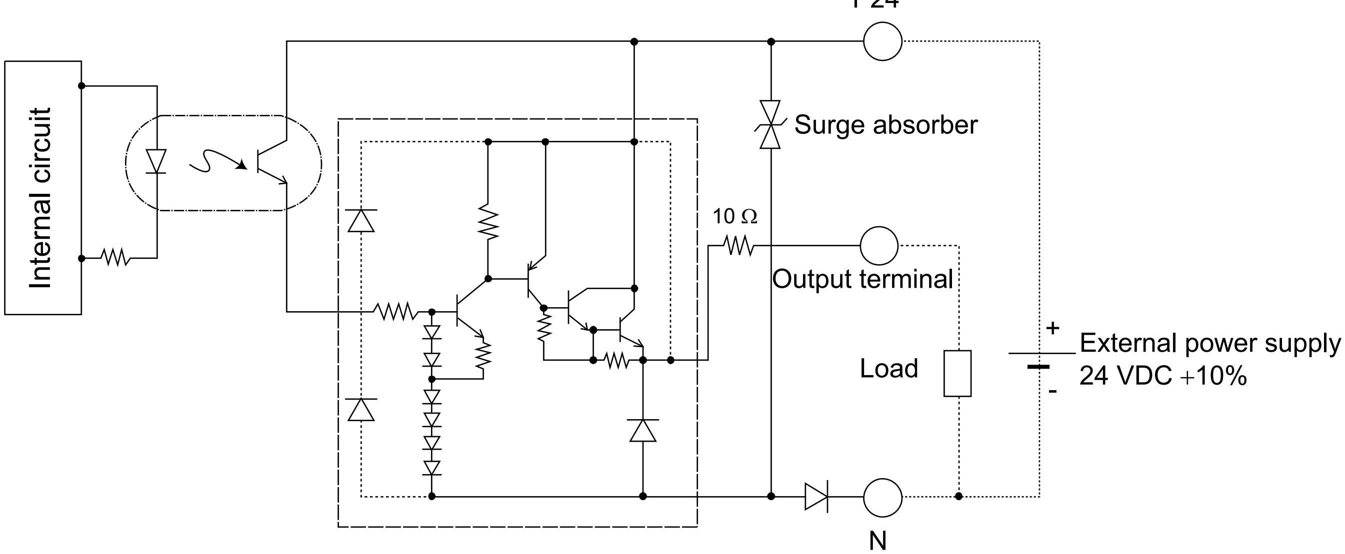
(2) Output part

External Output Specifications (PNP specification)

Item	Specification	
Load voltage	24 VDC	TD62784 (or equivalent)
Maximum load current	100 mA per point, 400 mA per 8 ports Note)	
Leakage current	0.1 mA max. per point	
Insulation method	Photocoupler insulation	
External devices	[1] Miniature relay [2] Sequencer input unit	

Note) 400 mA is the maximum total load current of output port Nos. 300 to 307.

[Output circuit]



* P24: I/O interface pin No. 1A
* N: I/O interface pin No. 17B

Caution

In the event that the load is short-circuited or a current exceeding the maximum load current is input, the overcurrent protection circuit will be actuated to cut off the circuit. However, give due consideration to the circuit connection layout to prevent short-circuit or overcurrent.

6.6 Connecting the Teaching Pendant/PC (Software) (TP) (Optional)

The ASEL controller's teaching connector (TP) is a small, connector. If you are using a teaching pendant or PC software cable, connect the cable to a connector conversion cable, and then connect the conversion cable to the teaching connector on the controller.

Note ASEL-C Type : CB-SEL-SJ002
ASEL-CS Type : CB-SEL-SJS002

6.7 Connecting the Panel Unit (Optional)

When the optional panel unit is connected, the controller status (program number of each active program, error codes, etc.) can be monitored.

6.7.1 Explanation of Codes Displayed on the Panel Unit (Optional)

(1) Application

Display				Priority (*1)	Description
	d	C	F	1	Control power cut off
E	F	x	x	1	System-down level error
	P	r	d	2	Writing data to the flash ROM.
	E	r	G	3	Emergency stop is being actuated (except during the update mode).
	E	n	b	4	Enable switch (deadman switch/safety gate) OFF (except in the update mode)
E	E	x	x	5	Cold-start level error
E	d	x	x	5	Cold-start level error
E	C	x	x	5	Operation-cancellation level error
E	b	x	x	5	Operation-cancellation level error
–		r	P	6	Waiting for a drive-source cutoff reset input (except during the update mode).
–		r	S	6	Operation is in pause (waiting for restart) (except during the update mode).
–	I	L	C	7	All servo axes are interlocked (except during the update mode).
E	A	x	x	8	Message level error
E	9	x	x	8	Message level error
r	U	d	C	9	Core update mode
	U	d	C	9	Core update is in progress.
F	U	d	c	9	Core update has completed.
r	U	d	S	9	Slave update mode
	U	d	S	9	Slave update is in progress.
F	U	d	S	9	Slave update has completed.
	P	N	o.	9	Running a program (last started program); “No.” indicates program number.
I	n	x	x	9	Initialization sequence number
	d	b	G	9	Debug mode

(*1) The priority increases as the number decreases.

Display				Priority (*1)	Description
A	r	d	y	9	Ready status (auto mode) (Program mode)
	r	d	y	9	Ready status (manual mode) (Program mode)
P	n	N	o.	9	Operating in positioner mode; “No.” indicates positioner mode number.
A	P	r	y	9	Ready status (auto mode) (Positioner mode)
	P	r	y	9	Ready status (manual mode) (Positioner mode)

(*1) The priority increases as the number decreases.

(2) Core

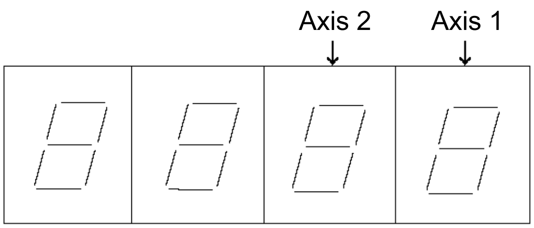
Display	Priority (*1)	Description
dCF	1	Control power cut off
EExx	1	Cold-start level error
Edxx	1	Cold-start level error
ECxx	1	Operation-cancellation level error
Ebxx	1	Operation-cancellation level error
EAxx	2	Message level error
E9xx	2	Message level error
rUdA	2	Application update mode
UdA	2	Application update is in progress.
FUdA	2	Application update has completed.
P___	2	Hardware test mode process
ErA	2	Clearing the application flash ROM.
FErA	2	Application flash ROM has been cleared.
JPA	2	Jump to the application
CHFC	2	Core flash-ROM check process
CHFA	2	Application flash-ROM check process
CHSd	2	SDRAM check process

(*1) The priority increases as the number decreases.

6.7.2 Current Monitor and Variable Monitor

By setting other parameter Nos. 49 and 50 appropriately, the optional panel unit can be used to monitor either current levels or variables.

(1) Current monitor
Currents of up to four axes having continuous axis numbers can be monitored.
Parameter settings
Other parameter No. 49 = 1
Other parameter No. 50 = Smallest axis number among the axes to be monitored
Example) If other parameter No. 49 is set to “1” and other parameter No. 50 to “1” for a 2-axis controller, the far-right segment digit will show the current for axis 1.



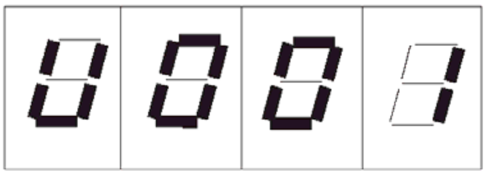
When data is written to the flash ROM or a software reset (restart) is executed after the parameter values have been input, the panel window will show the motor current to rating ratio (%) by a segment pattern, instead of “ready status” or “program run number.”
The segment display patterns and corresponding motor current to rating ratios (%) are shown below.

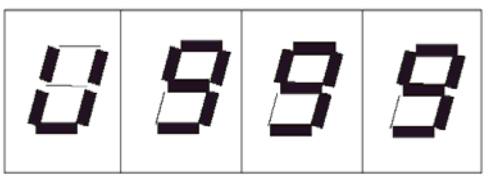
	0 < Motor current to rating ratio (%) ≤ 25		100 < Motor current to rating ratio (%) ≤ 150
	25 < Motor current to rating ratio (%) ≤ 50		150 < Motor current to rating ratio (%) ≤ 200
	50 < Motor current to rating ratio (%) ≤ 75		200 < Motor current to rating ratio (%)
	75 < Motor current to rating ratio (%) ≤ 100		

Thick lines indicate illuminated segments.

(2) Variable monitor
The contents of global integer variables can be displayed on the panel window.
Positive integers of 1 to 999 can be displayed.
Parameter settings
Other parameter No. 49 = 2
Other parameter No. 50 = Variable number of the global integer variable to be monitored

When data is written to the flash ROM or a software reset (restart) is executed after the parameter values have been input, the panel window will show the content of the global integer variable, instead of “ready status” or “program run number.” The far-left segment digit should read “U.”
Display example)

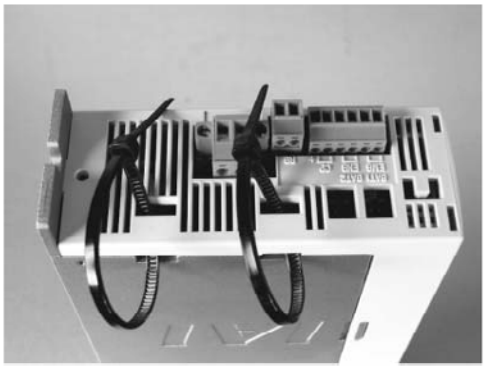




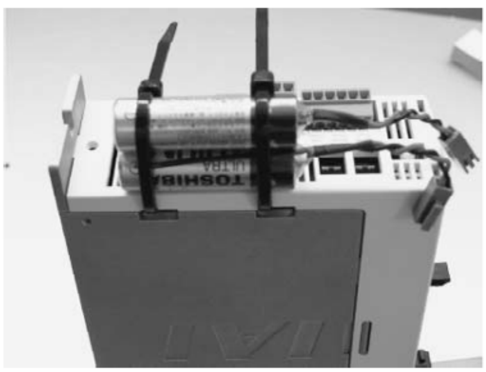
6.8 Installation Method for the Absolute-Data Backup Battery

The ASEL controller does not come with a holder or any other dedicated piece for installing the absolute-data backup battery. The user must affix the battery using tie-bands.

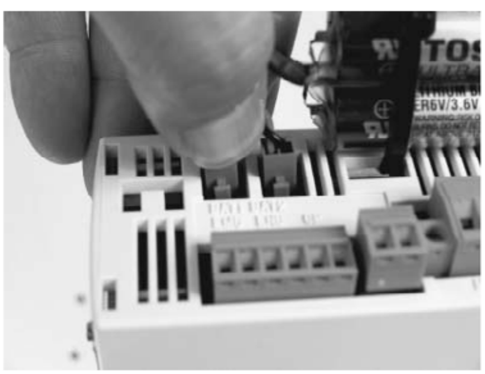
Example of installation



As shown to the left, guide tie-bands through the controller and tie the ends to make loose loops.

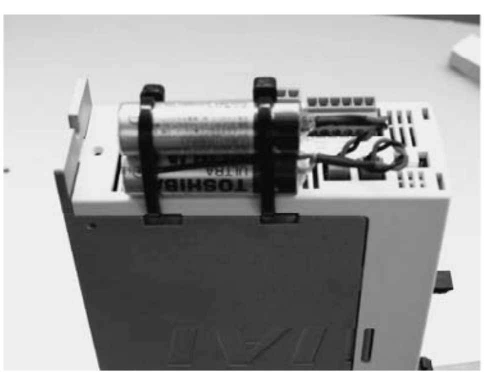


Guide the batteries into the tie-band loops.
Tighten the tie-bands and cut off any excess length at the end.

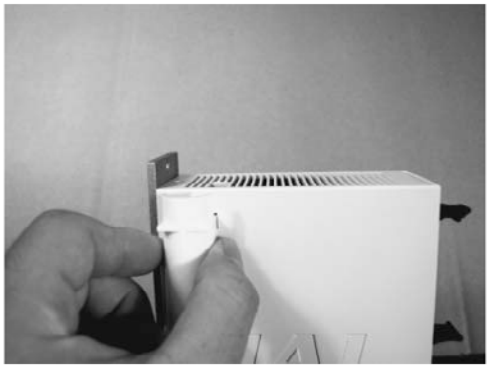


Connect each battery connector.
Pay attention to the connector orientation.
(The connector hook should face the left side when viewed from the front of the controller.)

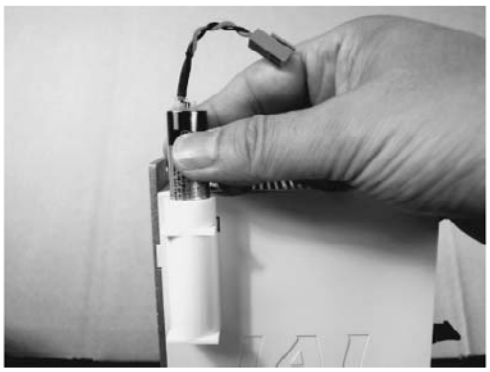
Caution:	If the main power cannot be turned on immediately after the encoder cable has been connected, do not connect the battery connector.



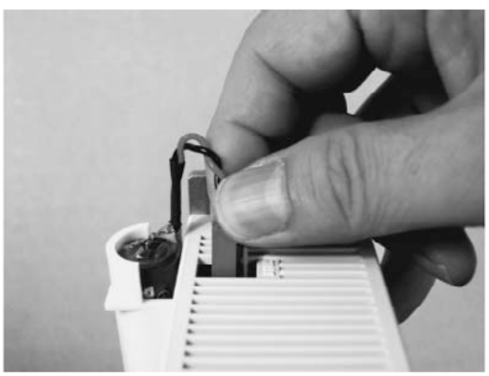
6.9 Installing the System-Memory Backup Battery (Optional)



As shown to the left, install the supplied battery holder on the left side face of the controller.



Insert the battery into the holder.



Connect the battery connector.
Pay attention to the connector orientation.
(The connector hook should face the right side.)

Chapter 4 Operation

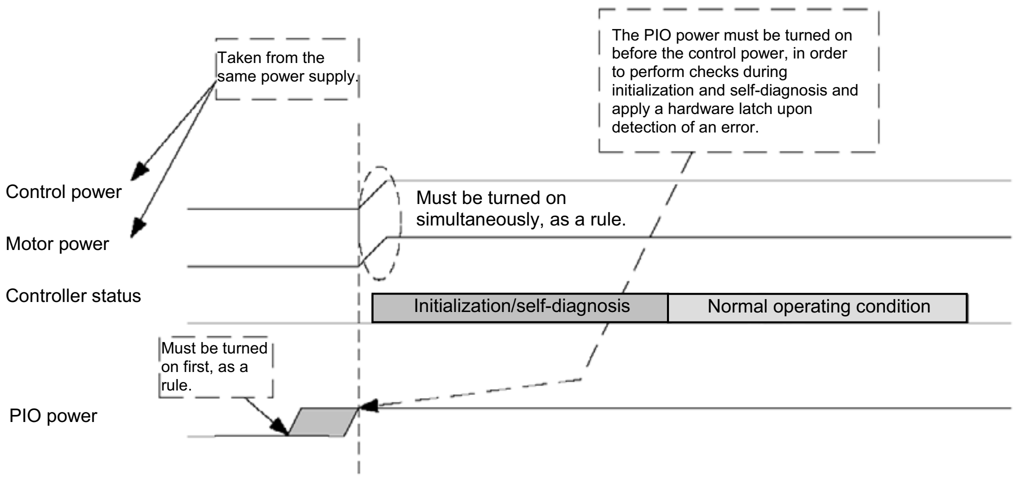
1. Startup

(1) Connect the motor cable and encoder cable to the controller.
(2) Connect the PIO connector to the host PLC using the supplied flat cable.
(3) Execute an emergency stop.
(4) Connect the PC or teaching pendant.
Set the AUTO/MANU switch to the “MANU” side.
(5) Supply the 24-V PIO power through the flat cable.
(6) Turn on the control power and motor power at the same time. (They should be taken from the same power supply.
(7) Reset the emergency stop.
★ The EMG lamp turns off.
★ If the ALM lamp is lit, an error is present. Check the error list to identify the problem.

If the 24-V PIO power is not supplied, an “E69” error will generate.
If your controller is of absolute specification, a “914” or “CA2” error may generate during the startup, indicating that an absolute reset must be performed. Refer to “How to Perform Absolute Reset.”
To check for errors, connect the teaching pendant, PC software or panel unit.

1.1 Power ON Sequence

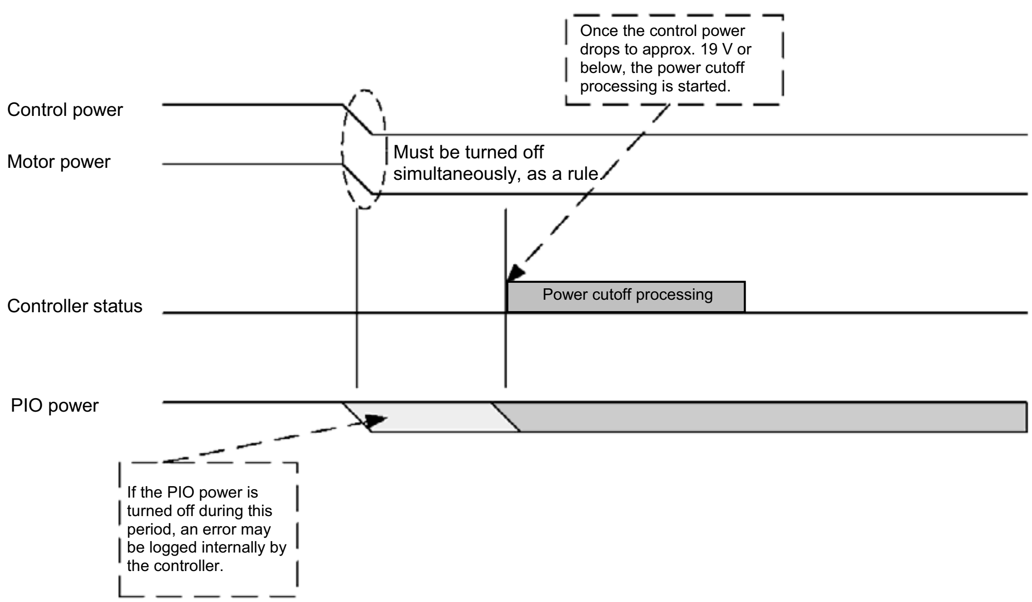
- Although separate inputs are provided for the control power and motor power, they should be supplied from the same power-supply terminal.
- Turn on the PIO power first. You can turn on the PIO power much earlier than the control power and motor power, as long as it is turned on before the control power/motor power.



* If the PIO power is not turned on before the control power is turned on, an error will be detected.

1.2 Power Cutoff Sequence

- If the PIO power is turned off before the control power and motor power (before the power cutoff processing is performed), a PIO power error may be logged internally by the controller.
- The PIO power can be turned off much later than the control power and motor power, as long as it is turned off after the control power/motor power.



2. How to Perform Absolute Reset (Absolute Specification)

If the ASEL controller experiences any abnormal absolute-encoder battery voltage or the battery or encoder cable is disconnected, an encoder battery error will generate. In this case, you must perform an absolute reset.
This chapter explains how to perform an absolute reset using the PC software. For the procedure to perform an absolute reset from the teaching pendant, refer to the operation manual for your teaching pendant.

2.1 Preparation

(1) PC
PC in which IAI's X-SEL PC software (X_SEL.exe) has been installed

(2) PC cable (supplied with the PC software)
RS232C cross cable (fitted with a female 9-pin connector on the PC end and a male 25-pin connector on the controller end)
+Connector conversion cable
Alternatively, use a USB cable and a dummy plug (optional).

(3) All adjustment items other than absolute reset must have been completed.

2.2 Procedure

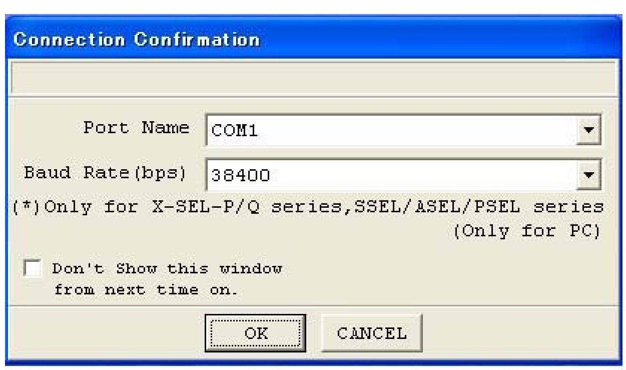
(1) Turn off the ASEL controller power. Turn on the PC power and wait for the OS to start.

(2) Connect the 9-pin D-sub connector on the PC cable to the communication port on the PC, and connect the 25-pin D-sub connector to the teaching connector on the controller.
Alternatively, connect the PC and controller using a USB cable. If the USB port is used, a dummy plug must be connected to the teaching connector.

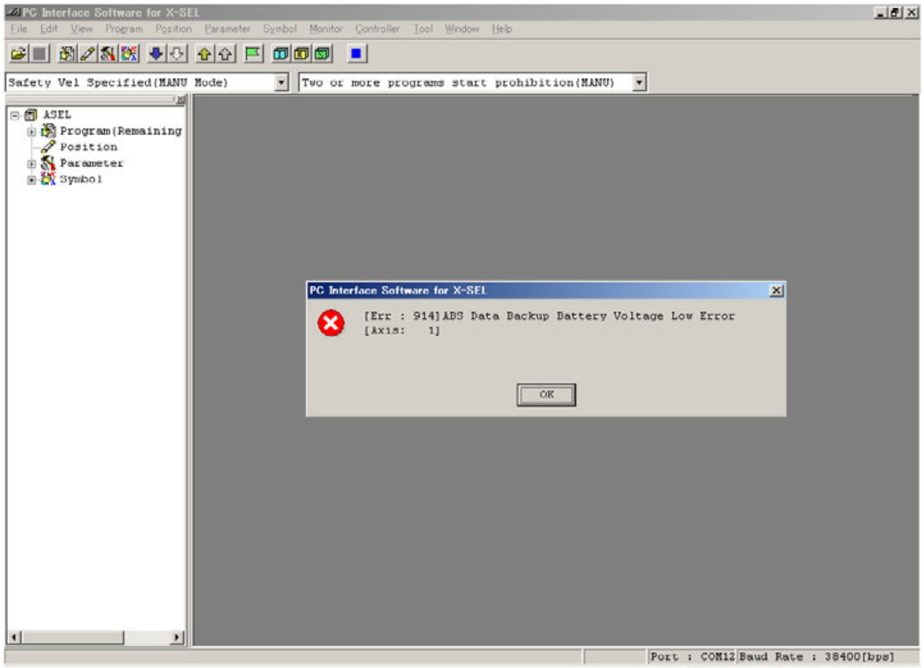
(3) Turn on the controller power. An encoder battery error will generate. If no other adjustment item is outstanding, "ECA2" or "E914" should be displayed on the 7-segment LED. This indicates that the controller has detected the encoder battery error.

(4) Launch the X-SEL PC software (X_SEL.exe) on the PC. The following steps explain the operating procedures in the X-SEL PC software.

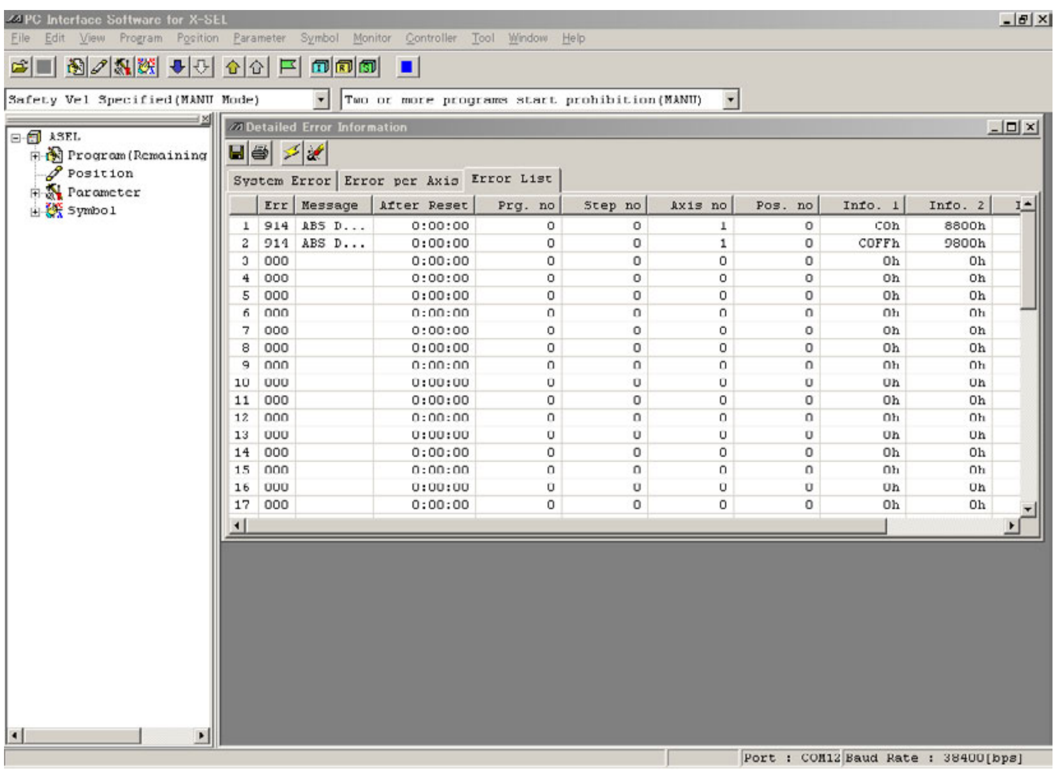
(5) When the Connection Check dialog box appears, set the communication port you are using on your PC. Click **OK**. (The baud rate need not be set. The software will automatically detect and set the baud rate.)



(6) The main window of the X-SEL PC software opens.
Click **OK** to close the error message.

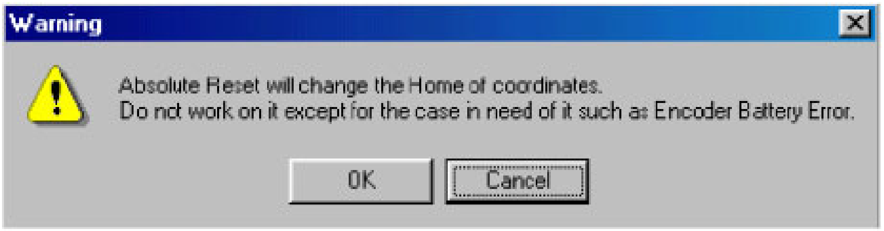


(7) From the **Monitor** menu, select **Error Detail** to check the condition of the present error.
If the controller is experiencing an encoder battery error, the displayed window should look like the one shown below (an absolute encoder is used for axis 2 in this example). After checking the error detail, close the Error Detail window.

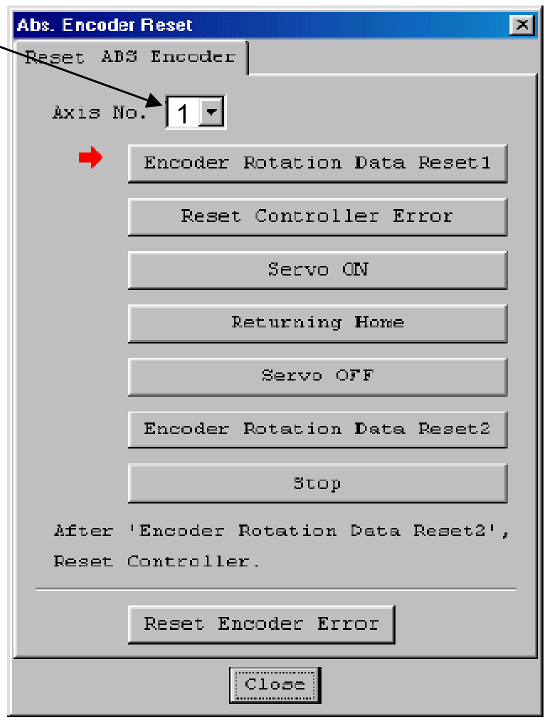


(8) From the **Controller** menu, select **Absolute Reset**.

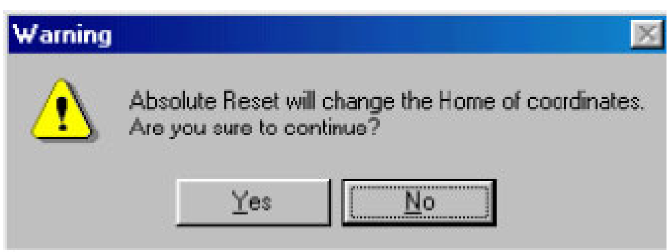
(9) When the Warning dialog box appears, click **OK**.



(10) The Absolute Reset dialog box appears.
Click here to select the axis you want to perform an absolute reset for.



(11) Click **Encoder Rotation Data Reset 1**. When the Warning dialog box appears, click **Yes**.



(12) Another Warning dialog box is displayed. Click **Yes** again.

(13) After the controller has finished processing encoder rotation data reset 1, the red arrow will move to the next item. Click the following processing buttons in this order (the arrow will move to the next one after each processing is completed):

1. Controller Error Reset
2. Servo ON
3. Home Return
4. Servo OFF
Encoder rotation data reset 2 is performed with the servo turned on. Accordingly, the Servo OFF step will be skipped.
5. Encoder Rotation Data Reset 2

After you have clicked **Encoder Rotation Data Reset 2** and the processing is finished, the red arrow will return to the position in shown in (10). To perform an absolute encoder reset for another axis, select the target axis and perform the steps from (10) again. To end the procedure, click **Close** to close the Absolute Reset dialog box.

(Note) If you have encountered a situation where an absolute encoder reset is required for two or more axes, be sure to repeat steps (10) to (13) for all applicable axes before performing the software reset in step (14) below.

(14) When the Confirmation dialog box appears, click **Yes** to restart the controller.

(Note) If you continue to operate the controller without resetting the software or reconnecting the power, the following errors may generate:
Error No. C70: ABS coordinate non-confirmation error
Error No. C6F: Home-return incomplete error

(15) If an optional panel unit is connected and no other error is present, “rdy” (when the controller is in the program mode) or “Pry” (in the positioner mode) should be displayed on the 7-seg LED.

(16) This completes the absolute reset.
To repeat the absolute reset, close the X-SEL PC software and perform the steps from the beginning.

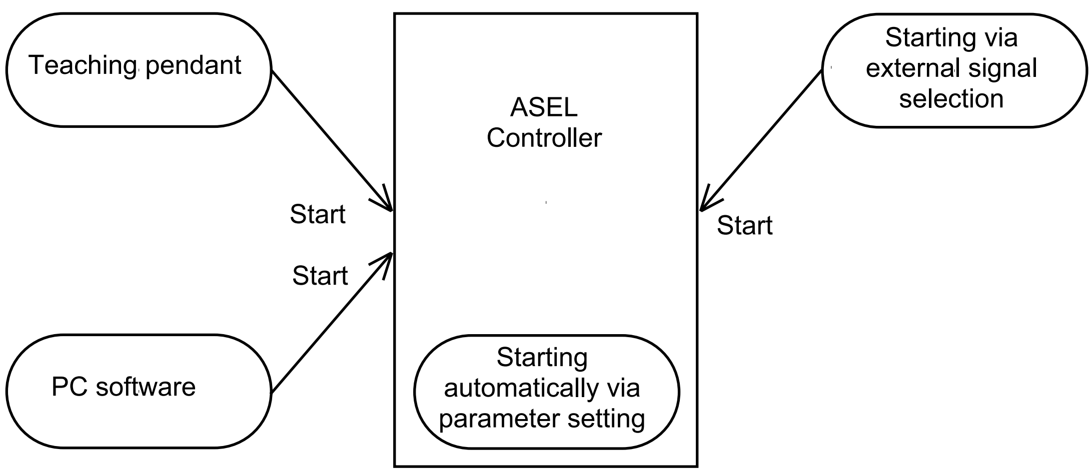
3. How to Start a Program

With the ASEL Controller, the stored programs can be started (run) using four methods. Of these methods, two are mainly used to debug programs or perform trial operations, while the remaining two are used in general applications on site.

The former two methods are “starting from the teaching pendant” and “starting from the PC software.”

These methods provide simple means of checking the operation. For details on “starting from the teaching pendant,” read the operation manual for the optional teaching pendant. For “starting from the PC software,” read the applicable explanation in the manual supplied with the PC software.

The latter two methods are “starting automatically via parameter setting” and “starting via external signal selection.” This chapter only explains the methods for “starting automatically via parameter setting” and “starting via external signal selection.”



3.1 Starting a Program by Auto-Start via Parameter Setting

Other parameter No. 7 (Auto program start setting) = 1 (Standard factory setting)

This parameter is set using the teaching pendant or PC software.

Set an auto-start program number

Set the number of the program you wish to start automatically in other parameter No. 1 (auto-start program number).
Set the controller mode to AUTO.

⇩

Reset the controller

Reconnect the power or execute a software reset, and the controller will be reset.

⇩

Automatically starting the program

Once the controller is reset in the above step, the program of the set number will start automatically.
*

Caution

[Note on starting a program by auto-start]
The automatic operation will begin immediately after the controller is reset, so the user may be surprised by unexpected movements of the equipment, particularly those caused by a sudden activation of the servo actuator. To ensure safety, always provide an interlocking function, such as allowing the program execution to proceed only after receiving a confirmation signal at the beginning of the program.

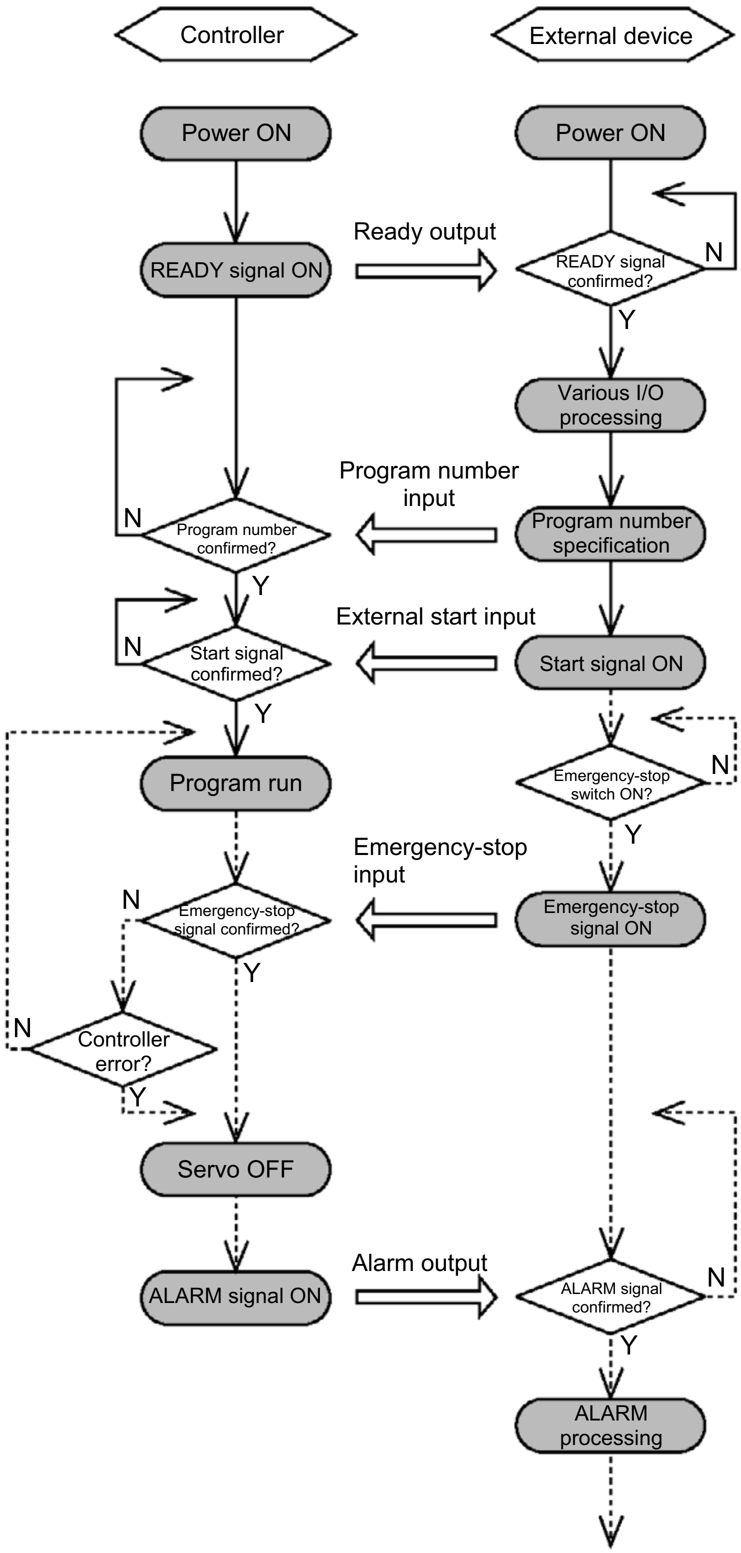
If you wish to start multiple programs at the same time, write multiple “EXPG” commands at the beginning of the main program to start the remaining programs. Provide safety measures for each program to be started.

* If the following setting is performed, the program of the selected program number will start automatically at the ON edge of the signal received by the selected input port. The program will be aborted at the OFF edge.
You can set a desired input port for receiving the auto program start signal (dedicated function).
Set the input function setting value “5” in the I/O parameter corresponding to the desired input port number (Nos. 30 through 45, 251 through 258).
(Refer to “I/O Function Lists” and “I/O Parameters.”)

3.2 Starting via External Signal Selection

Select a desired program number externally and then input a start signal.

(1) Flow chart



When the READY signal (Output port No. 301) turns ON, the RDY lamp (green) on the controller front panel will illuminate.

Input a desired program number as a BCD code from the external device (Input port Nos. 16 through 22).

Input a start signal (input port No. 0) from the external device.

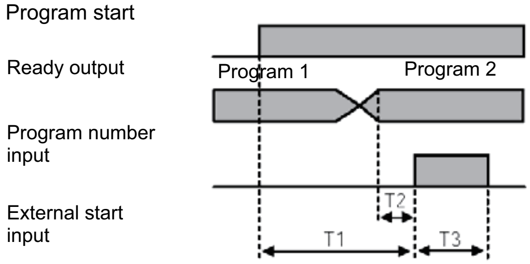
If the optional panel unit is connected, the CODE display area indicates the program number of each program that has been started.

If an emergency-stop signal was input from the external device or a controller error occurred, the controller will turn off the servo power. (The RDY lamp will turn off.)

Note) The assignments of dedicated input/output port functions (such as RDY output start signal) reflect the factory settings.

[2] Timing chart

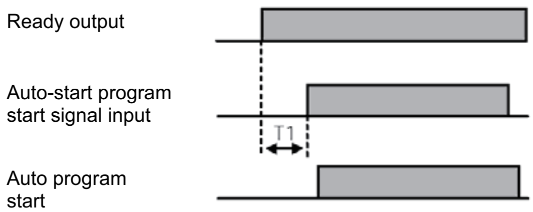
[1] Program start



T1: Duration after the ready output turns ON until input of external start signal is permitted
T1 = 10 msec min.
T2: Duration after the program number is input until input of external start signal is permitted
T2 = 50 msec min.
T3: Input duration of external start signal
T3 = 100 msec min.

[2] Auto program start

* Set input function specification value 5 (auto-start program start signal) for input port No. *.

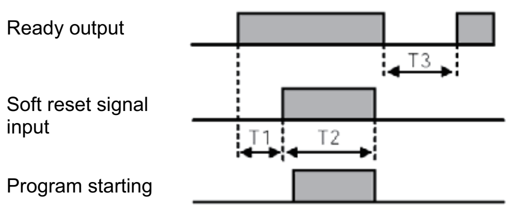


T1: Time after the ready output turns ON until the auto-start program start signal can be input to input port No. *
T1 = 10 msec min.

* Auto program start:
Set "0" in other parameter No. 7, "Auto program start setting."

[3] Soft reset signal

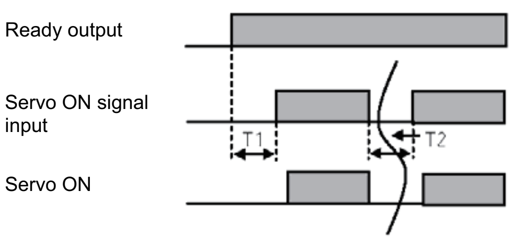
* Set input function specification value 3 (soft reset signal) for input port No. *.



T1: T1: Time after the ready output turns ON until input function specification value 3 (soft reset signal) can be input to input port No. *
T1 = 10 msec min.
T2: T2: Time until the soft reset signal becomes effective
T2 = 1 sec min.
T3: Time after the soft reset signal is cancelled until the ready signal is output

[4] Servo ON signal

* Set input function specification value 4 (servo ON signal) for input port No. *.

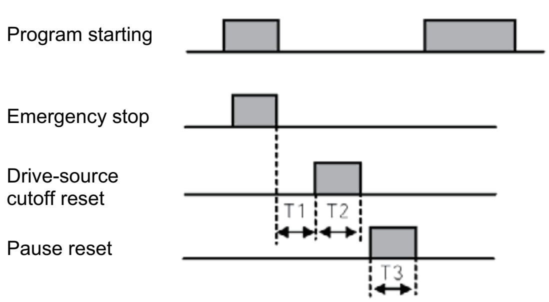


T1: Time after the ready output turns ON until input function specification value 4 (servo ON signal) can be input to input port No. *
T1 = 10 msec min.
T2: Interval after the servo is turned OFF until it is turned ON again
T2 = 1.5 sec min.

> ⚠ Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

[5] When the recovery type after emergency stop or enable operation is set to "Operation continued"

* Set other parameter No. 10 to "2," and set input function specification value 7 (operation-pause reset signal) for input port No. *. Set input function specification value 17 (drive-source cutoff reset input signal) for other input port No. *.



T1: Time after the emergency stop input is reset until the drive-source cutoff reset signal can be input.
T1 = 2 sec min.
T2: Time during which the drive-source cutoff reset signal is input
T1 = 10 msec min.
T3: Time during which the pause reset signal is input
T1 = 10 msec min.

4. Drive-Source Recovery Request and Operation-Pause Reset Request

(1) Drive-source recovery request

[1] Case where a drive-source request is required

A drive-source recovery request is required in the following case:

- Specify a desired input port for receiving the drive-source cutoff reset input signal (dedicated function).
  Occurrence of a drive-source cutoff factor → Recovery after the cutoff factor is removed.

[2] How to request a drive-source recovery

A drive-source recovery request can be issued using one of the following methods:

- Set the input function specification value "17" in the I/O parameter corresponding to the desired input port number (Nos. 30 through 45, 251 through 258). (Refer to "I/O Function Lists" and "I/O Parameters.")
  Input the ON edge to the input port of the specified number.
- Select [Drive-Source Recovery Request (P)] from the [Controller (C)] menu on the PC software screen.
- Select Ctl (controller operation) and RPwr (drive-source recovery request) on the mode selection screen of the teaching pendant.

(2) Operation-pause reset request

[1] Cases where an operation-pause reset request is required

An operation-pause reset request is required in any of the following cases:

- An emergency stop was actuated during automatic operation when other parameter No. 10 was set to "2" (Emergency-stop recovery type = Continued operation) (only during automatic operation) → Recovery (reset of operation pause) after the emergency stop is reset.
- The automatic operation was stopped using the deadman switch or enable switch when other parameter No. 11 was set to "2" (Deadman/enable switch recovery type = Continued operation) (only during automatic operation) → Recovery (reset of operation pause) after the stop is reset.
- Specify a desired input port for receiving the operation-pause input signal (dedicated function).
  Set the input function specification value "8" in the I/O parameter corresponding to the desired input port number (Nos. 30 through 45, 251 through 258). (Refer to "I/O Function Lists" and "I/O Parameters.")
  OFF level signal input is received by the import port of the specified number during auto operation (operations pause) → Recovery after detection of ON signal level by the input port (operation pause is reset).

[2] How to request an operation-pause reset

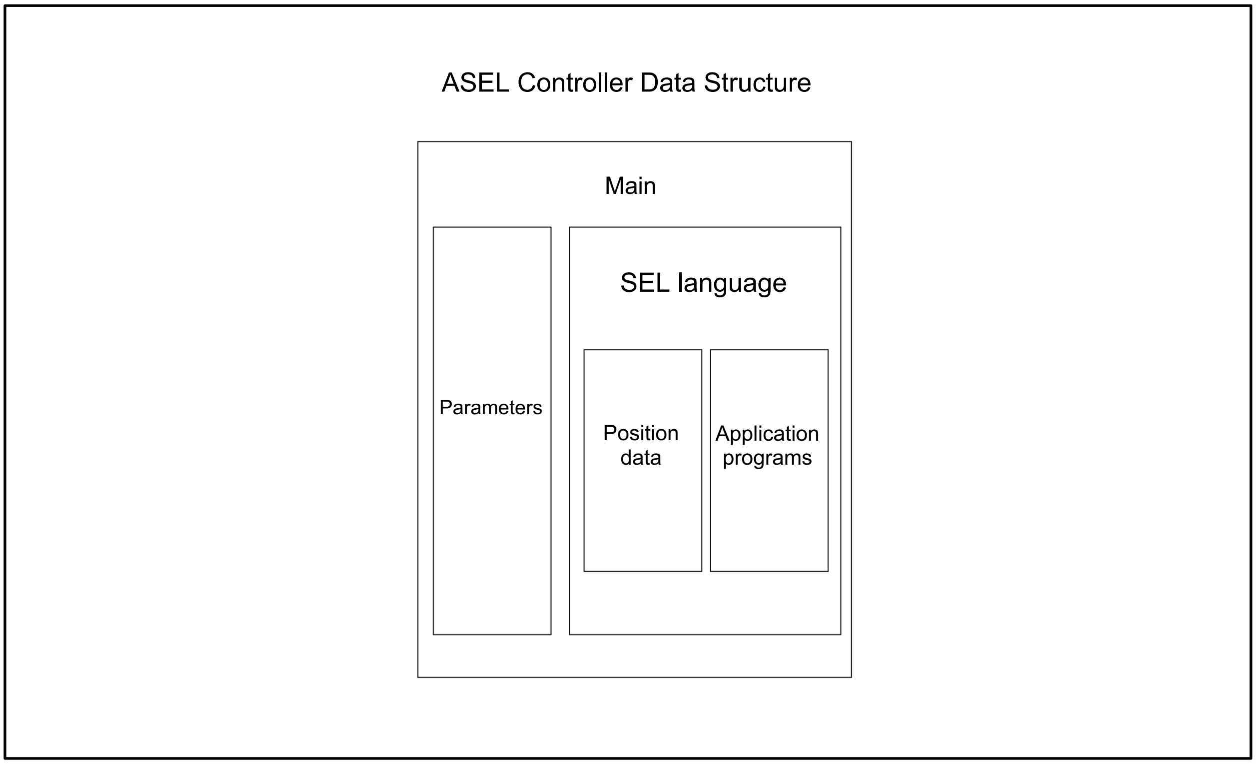
An operation-pause reset request can be issued using one of the following methods:

- Specify a desired input port for receiving the operation-pause input signal (dedicated function).
  Set the input function specification value "7" in the I/O parameter corresponding to the desired input port number (Nos. 30 through 45, 251 through 258). (Refer to "I/O Function Lists" and "I/O Parameters.")
  Input the ON edge to the input port of the specified number.
- Select [Operation-Pause Reset Request (L)] from the [Controller (C)] menu on the PC software screen.
- Select Ctl (controller operation) and RAct (operation-pause reset request) on the mode selection screen of the teaching pendant.

* If the case in [1] of (1) and any of the cases in [1] of (2) are present at the same time, a drive-source recovery request must be issued first, followed by an operation-pause reset request.

5. Controller Data Structure

The controller data consists of parameters as well as position data and application programs used to implement SEL language.



The user must create position data and application programs. The parameters are predefined, but their settings can be changed in accordance with the user's system.
Refer to Appendix, "List of Parameters," for details on the parameters.

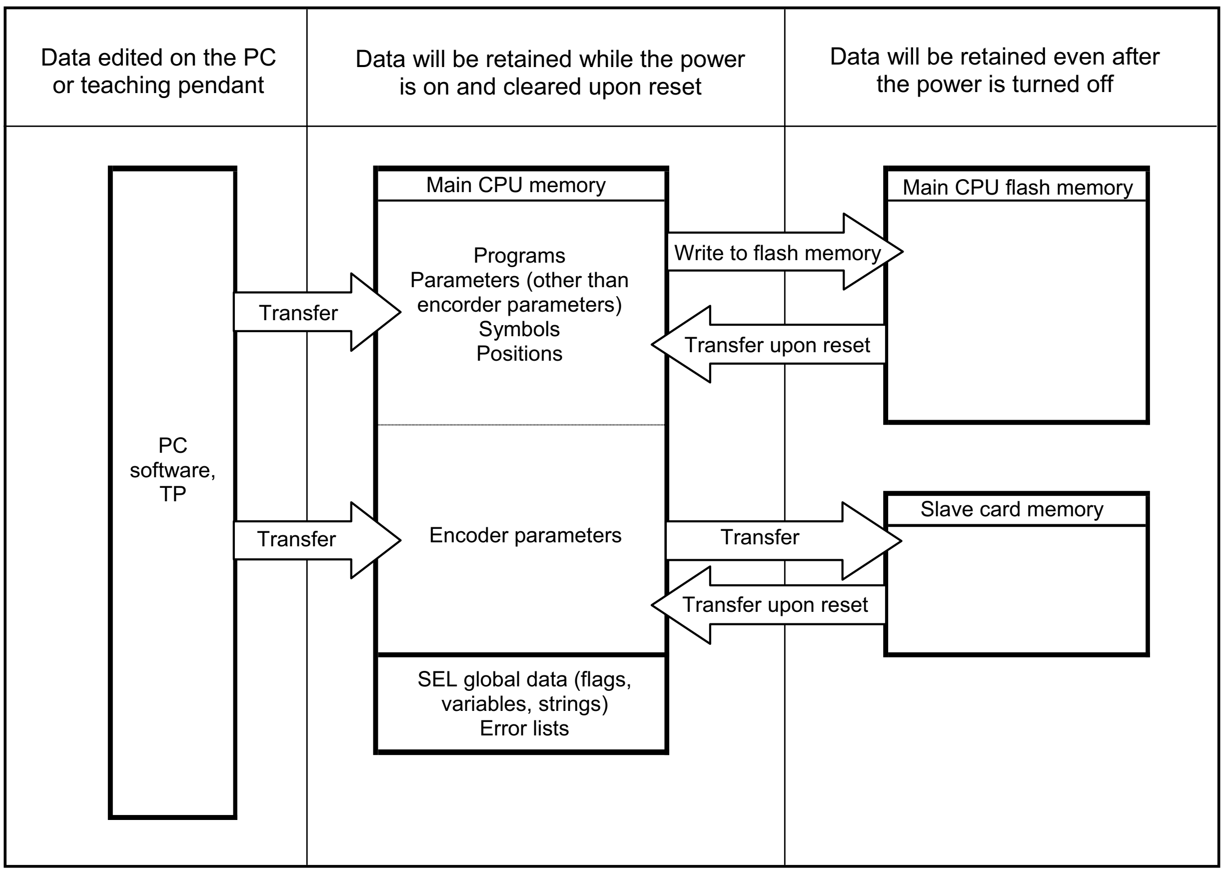
5.1 How to Save Data

The flow to save data in the ASEL controller is illustrated below.
When data is transferred from the PC software or teaching pendant to the controller, the data is only written to the main CPU memory as shown in the diagram below and will be erased once the controller is powered down or reset.

For important data, always write to the flash memory so that they will not be lost.

5.1.1 Factory Settings: When the System-Memory Backup Battery is Not Used

Other parameter No. 20 = 0 (System-memory backup battery not installed)



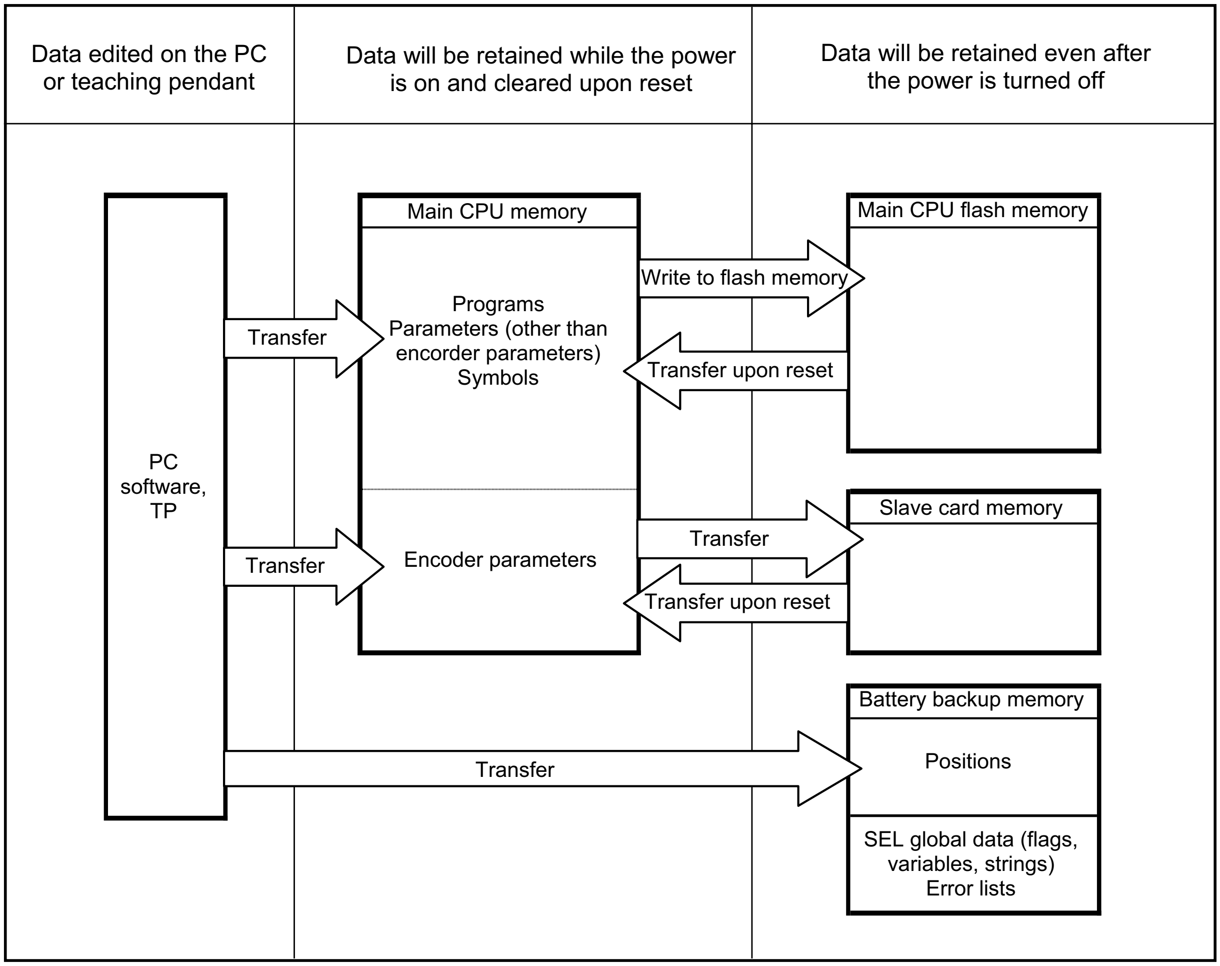
Since the programs, parameters and symbols are read from the flash memory at restart, the data in the temporary memory will remain the same as the original data before edit unless the edited data are written to the flash memory.

The controller always operates in accordance with the data in the main CPU memory (excluding the parameters).

Note: SEL global data cannot be retained if the backup battery is not installed.
SEL global data will be cleared once the control power is turned off or a software reset is executed.
The error list will be cleared once the control power is turned off.

5.1.2 When the System-Memory Backup Battery (Optional) is Used

Change the setting of other parameter No. 20 to 2 (System-memory backup battery installed).



Since the programs, parameters and symbols are read from the flash memory at restart, the data in the temporary memory will remain the same as the original data before edit unless the edited data are written to the flash memory.

The controller always operates in accordance with the data in the main CPU memory (excluding the parameters).

5.2 Points to Note

Point to note when transferring data and writing to the flash memory

Never turn off the main power while data is being transferred or written to the flash memory. The data will be lost and the controller operation may be disabled.

Point to note when saving parameters to a file

The encoder parameters are stored in the EEPROM of the actuator's encoder itself (unlike other parameters, they are not stored in the EEPROM of the controller). The encoder parameters will be read from the encoder's EEPROM to the controller when the power is turned on or upon software reset.

Therefore, if the parameters are saved to a file after turning on the controller (or restarting it via a software reset) without an actuator (encoder) connected, the encoder parameters saved to the file will become invalid.

Point to note when transferring a parameter file to the controller

When a parameter file is transferred to the controller, the encoder parameters will be transferred to the EEPROM of the encoder (excluding manufacturing/function information).

Therefore, if the parameter file transferred to the controller has been read from a controller that was started without an actuator connected, invalid encoder parameters will be written to the encoder's EEPROM (provided that an actuator is connected to the controller to which the file was transferred).

When saving the parameters to a file, do so with an actuator connected to the controller.

Chapter 5 Maintenance

- Routine maintenance and inspection are necessary so that the system will operate properly at all times. Be sure to turn off the power before performing maintenance or inspection.
- The standard inspection interval is six months to one year. If the environment warrants, however, the interval should be shortened.

1. Inspection points

- Check to see if the supply voltage to the controller is inside the specified range.
- Inspect the ventilation holes in the controller and remove dirt, dust and other foreign attachments, if any.
- Inspect the controller cables (controller → actuator) and check for any loose screws or cable disconnection.
- Check the controller mounting screws, etc., for looseness.
- Inspect each cable (axis link cable, general-purpose I/O cable, system I/O cable, power cable) for loose connection, disconnection, play, etc.

2. Spare consumable parts

Without spare parts, a failed controller cannot be repaired even when the problem is identified quickly. We recommend that you keep the following consumable parts as spares:

Consumable parts
- Cables
- System-memory backup battery (optional): AB-5 by IAI -- Must be replaced after approx. 5 years*
- Absolute-data backup battery (optional): AB-5 by IAI -- Must be replaced after approx. 2 years* (Absolute specification)
  *: The actual replacement timing will vary depending on the use condition. For details, refer to “◎ Battery Backup Function” in Appendix.

When the battery voltage drops, an applicable error code will be displayed on the panel window.

Error Codes Indicating Low Battery Voltage

System-memory backup battery	A01 or A02
Absolute-data backup battery	A23

3. Replacement Procedure for System-Memory Backup Battery (Optional)

Backing up the system memory

If the optional system-memory backup battery is installed in the ASEL controller and "Other parameter No. 20: Backup battery installation function type" is set to "2" (Installed), the following SRAM data will be retained even after the power is turned off:

- Position data
- SEL global data (flags, integer/real variables, string variables)
- Error list

Always follow the procedure below when replacing the system-memory backup battery:

Note:	If the system-memory backup battery is disconnected while other parameter No. 20, "Backup battery installation function type" is still set to "2" (Installed), the data stored in the SRAM will be lost. So that the position data can be restored after an accidental loss from the SRAM, save the position data to a file using the PC software before disconnecting the battery. For the method to save the position data to a file, refer to 6, "Position Data Edit Window" in the X-SEL PC Software Operation Manual.

(1) Turn on the controller power.

(2) Record (write down) the current setting of "Other parameter No. 20, Backup-battery installation function type." (This will be used when reverting the parameter to its original setting following the replacement of system-memory backup battery.)

(3) If the PC software is installed in your PC, save the position data to a file using the PC software. The data will be used as a backup in case the SRAM data saved to the flash ROM fails.

(4) Change "Other parameter No. 20, Backup-battery installation function type" to "1" and transfer the setting to the controller, and then perform a flash ROM write. (The point data will be saved to the flash ROM.)

* Confirm that the flash ROM writing process has completed.

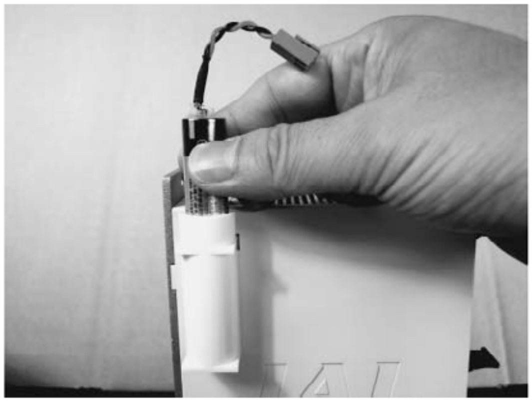
(5) Perform a software reset to restart the controller. (The SEL global data and error lists will be saved to the special area in the flash ROM.)

(6) When the controller has been restarted, turn off the power.

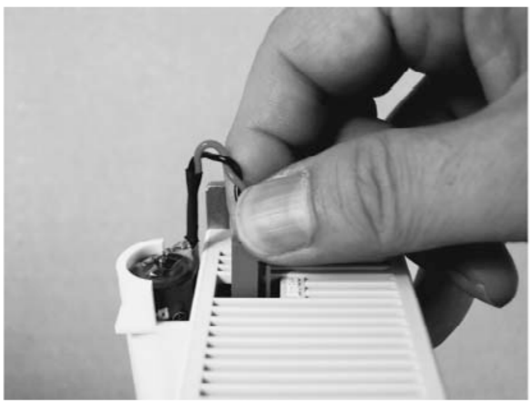
* Be sure to keep the power on from the start of controller restart until the RDY LED lamp on teh controller illuminates.

(7) Replace the system-memory backup battery. SRAM data will be lost if steps (1) through (6) are not performed properly.

Battery Replacement Procedure



[1] Remove the battery connector and pull out the battery.



[2] Insert a new battery into the holder and plug in the battery connector. The connector hook should face the right side.

(8) When the replacement of system-memory backup battery is complete, confirm that the battery is installed securely and then turn on the controller power.

(9) Revert “Other parameter No. 20, Backup-battery installation function type” to the value recorded in step (2), transfer the setting to the controller, and then perform a flash ROM write.

* Confirm that the flash ROM writing process has completed.

(10) Perform a software reset (restart the controller).
(Note) Commencing the operation without first executing a software reset or reconnecting the power may generate the following errors:
Error No. C70: ABS coordinate non-confirmation error
Error No. C6F: Home-return incomplete error

(11) When the controller has been restarted, confirm that the SRAM data have been restored.

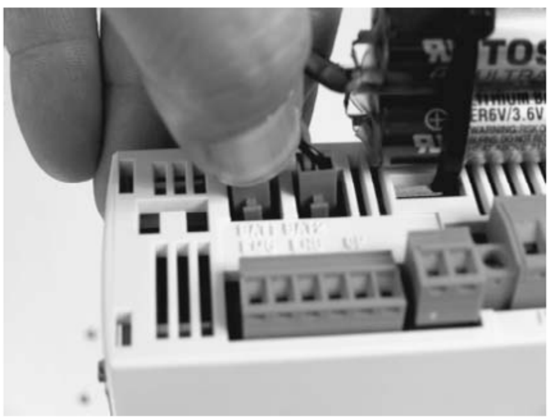
4. Replacement Procedure for Absolute-Data Backup Battery (Optional)

The replacement procedure is different depending on which error is present (No. A23, 914, CA2), or if no error is present at all, when the battery is replaced.

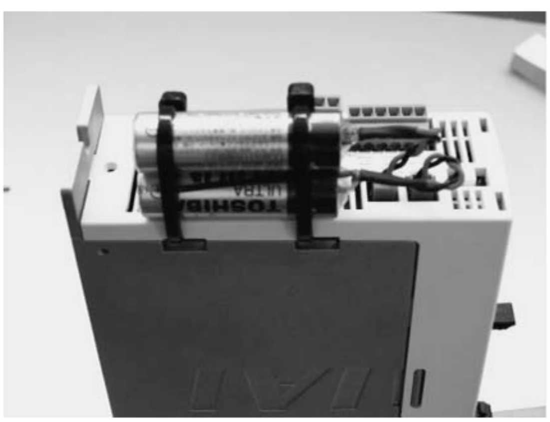
- If no error is present, perform steps (1) to (4).
- If an absolute-data backup battery voltage-low warning (Error No. A23) has been issued, perform steps (1) to (11).
- If an absolute-data backup battery voltage error (Error No. 914 or CA2) has been issued, perform steps (1) to (4) and then perform the procedure explained in Chapter 4-2 of Part 1 "How to Perform Absolute Reset."

Note: Among the steps explained below, complete (2) to (4) within 15 minutes.

(1) Turn off the controller power. (Turn off both the control power and drive power.)

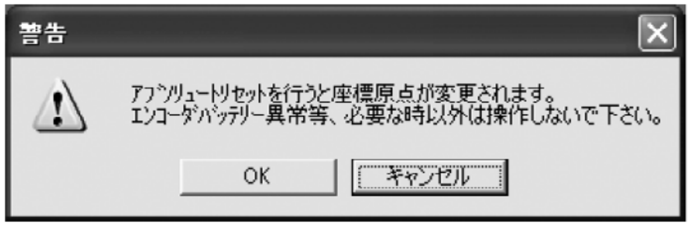


(2) Remove the battery connector and pull out the battery.



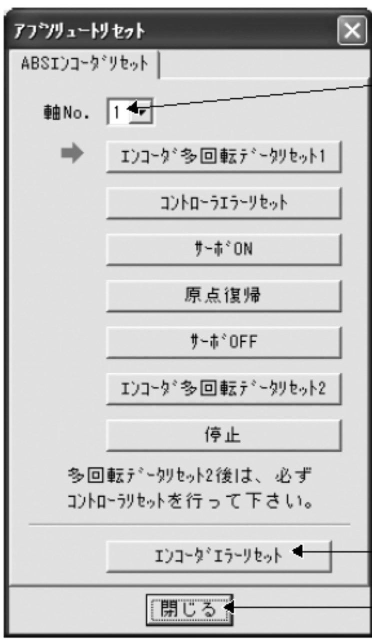
(3) Insert a new battery into the holder and plug in the battery connector. The connector hook should face the right side.

(4) Turn on the controller power.
(5) Start the PC software on a PC connected to the controller. From the **Controller** menu, select **Absolute Reset**.
(6) When the Warning dialog box appears, click **OK**.



Warning

(7) The Absolute Reset dialog box appears.



(8) Set the address number corresponding to the axis whose battery has just been replaced.
Note) Do not click **Encoder Rotation Data Reset 1**.

(9) Click **Encoder Error Reset**.

(10) Close the dialog box.

Absolute Reset

(11) In the PC software window, click the **Controller** menu and then select **Software Reset** to restart the controller.

Confirmation

(Note) If you continue to operate the controller without resetting the software or reconnecting the power, the following errors may generate:
Error No. C70: ABS coordinate non-confirmation error
Error No. C6F: Home-return incomplete error

This completes the procedure to reset a battery voltage low alarm/error.

Part 2 Programs

Chapter 1 SEL Language Data

1. Values and Symbols Used in SEL Language

1.1 List of Values and Symbols Used

The various functions required in a program are represented by values and symbols.

Function	Global range	Local range	Remarks
Input port	000 ~ 299 (300)		Varies depending on the function.
Output port	300 ~ 599 (300)		Varies depending on the function.
Flag	600 ~ 899 (300)	900 ~ 999 (100)	
Variable (integer)	200 ~ 299 (100) 1200 ~ 1299 (100)	1 ~ 99 (99) 1001 ~ 1099 (99)	99 is used for IN, INB, OUT, OUTB, etc.
Variable (real)	300 ~ 399 (100) 1300 ~ 1399 (100)	100 ~ 199 (100) 1100 ~ 1199 (100)	199 is used for PPUT, PGET, PARG, etc.
String	300 ~ 999 (700)	1 ~ 299 (299)	
Tag number		1 ~ 256 (256)	
Subroutine number		1 ~ 99 (99)	
Zone number	1 ~ 4 (4)		
Pallet number		1 ~ 10 (10)	
Axis number	1 ~ 2 (2)		Varies depending on the function.
Axis pattern	0 ~ 11		
Position number	1 ~ 1500 (1500)		
Program number	1 ~ 64 (64)		
Step number	1 ~ 2000 (2000)		
Task level	NORMAL/HIGH (2)		
SIO channel number	0 (1)		
Wait timer		1	
1-shot pulse timer		16 (Number of timers that can be operated simultaneously)	
Ladder timer		Local flag (100)	
Virtual input port (SEL system → SEL user program)	7000 ~ 7299 (300)		
Virtual output port (SEL user program → SEL system)	7300 ~ 7599 (300)		
Number of symbol definitions	500		
Number of times symbol can be used in commands	2500 (including literals)		
	Used in common from any program.	Referenced separately in each program. Cleared when the program is started.	

Caution

- Variables 99 and 199 are special variables this system uses in operations. Avoid using these two variables for general purposes.
- The values in the table represent ranges that can be processed by software. Items that require physical devices, such as I/O ports and functions relating to axis number and SIO, will be determined by possible combinations and models of commercial boards, etc., available for each device application.

- If the optional system-memory backup battery is installed, data of global variables and flags will be retained even after the controller power is turned off.
(Other parameter No. 20 must be set to "2." Refer to 5.1.2, "When the System Memory Backup Battery is Used" in Chapter 5 of Part 1.)

- The variables and flags in the local range will be cleared when the program is started.

- Ranges of values that can be used in SEL language
Integers and real numbers can be used. However, pay due attention to the following limitations:

(1) Numeric data
The ASEL Controller can handle values of maximum eight digits including a sign and a decimal point.
Integer: -9,999,999 to 99,999,999
Real number: Maximum eight digits including a sign and decimal point, regardless of the size of value
Example) 999999.9, 0.123456, -0.12345
If a floating point is used in operations, the number of valid digits will be limited to seven. Also note that operations using a floating point are subject to error.

(2) Position data
The input range of position data consists of four integer digits and three decimal digits.
–9999.999 to 9999.999
(The maximum value varies depending on the actuator model.)
If position data are used in internal operations as numeric data (repeated multiplications and divisions), the accuracy of the last digit may decrease.

Consider the above limitations fully when using values. Particularly when the CPEQ command is used in a comparison operation using real numbers, a match will rarely result. In this case, the CPLE or CPGE command that looks at the magnitude relationship of two terms must be used.

1.2 I/O Ports

(1) Input ports
Used as input ports for limit switches, sensor switches, etc.

Input number assignment
000 to 023 (standard)

(2) Output ports
Used as various output ports.

Output number assignment
300 to 307 (standard)

1.3 Virtual I/O Ports

(1) Virtual input ports

Port No.	Function
7000	Always OFF
7001	Always ON
7002	Voltage low warning for system-memory backup battery
7003	Abnormal voltage of system-memory backup battery
7004	(For future expansion = Use strictly prohibited)
7005	(For future expansion = Use strictly prohibited)
7006	Top-level system error = Message level error is present
7007	Top-level system error = Operation-cancellation level error is present
7008	Top-level system error = Cold-start level error is present
7009	(For future expansion = Use strictly prohibited)
7010	Drive-source cutoff factor is present (including when waiting for cutoff reset input)
7011	Latch signal indicating that all-operation-cancellation factor is present (latch signal for recognizing 1-shot cancellation factor; latch is cancelled by 7300-ON)
7012	All-operation-pause factor is present (including when waiting for restart switch signal) (Valid only during automatic operation recognition)
7013	All-servo-axis-interlock factor is present (all-operation-pause factor + interlock input-port factor)
7014	(For future expansion = Use strictly prohibited)
7015	Voltage low warning for axis-1 absolute-data backup battery
7016	Abnormal voltage of axis-1 absolute-data backup battery (latched until power-on reset or software reset)
7017	Voltage low warning for axis-2 absolute-data backup battery
7018	Abnormal voltage of axis-2 absolute-data backup battery (latched until power-on reset or software reset)
7019 ~ 7026	(For future expansion = Use strictly prohibited)
7027 ~ 7040	(For future expansion = Use strictly prohibited)
7041 ~ 7070	(For future expansion = Use strictly prohibited)
7071	In AUTO mode
7072	During automatic operation
7073 ~ 7100	(For future expansion = Use strictly prohibited)
7101	Running program No. 01 (including during pause)
~	~
7164	Running program No. 64 (including during pause)
7165 ~ 7299	(For future expansion = Use strictly prohibited)

(2) Virtual output ports

Port No.	Function
7300	Latch cancellation output for a latch signal indicating that all-operation-cancellation factor is present (7011) (latch is cancelled only when operation-cancellation factor is no longer present) (7300 will be turned OFF following an attempt to cancel latch.)
7301 ~ 7380	(For future expansion = Use strictly prohibited)
7381 ~ 7399	(For future expansion = Use strictly prohibited)
7400 ~ 7599	(For future expansion = Use strictly prohibited)

1.4 Flags

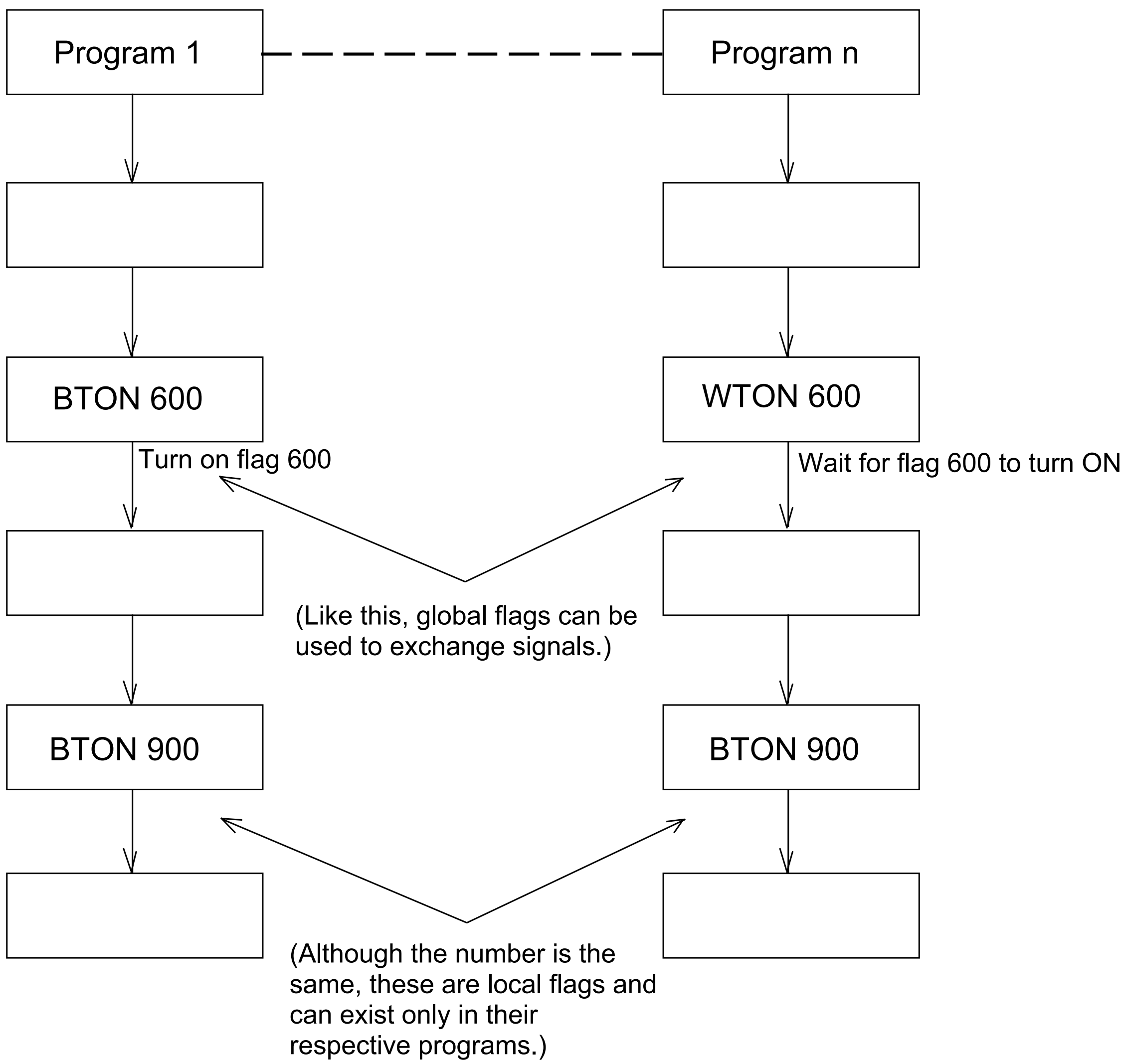
Contrary to its common meaning, the term "flag" as used in programming means "memory." Flags are used to set or reset data. They correspond to "auxiliary relays" in a sequencer.

Flags are divided into global flags (Nos. 600 to 899) that can be used in all programs, and local flags (Nos. 900 to 999) that can be used only in each program.

Global flags will be retained (backed up by battery) even after the power is turned off.

Local flags will be cleared when the power is turned off.

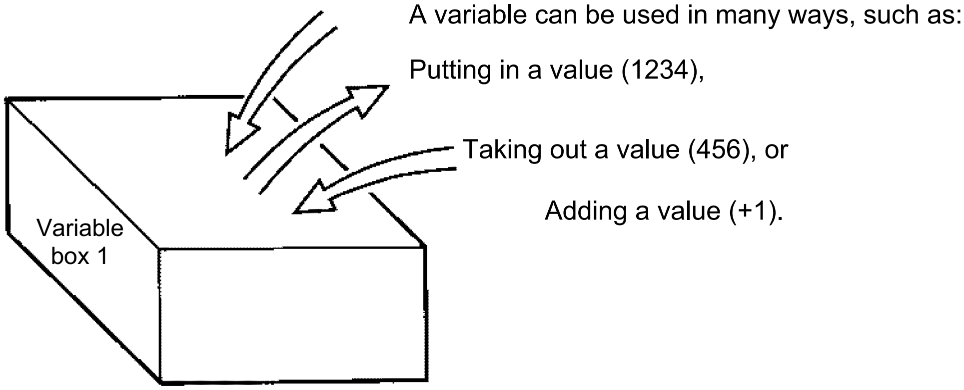
Flag number	600 ~ 899	Can be used in all programs	"Global flags"
Flag number	900 ~ 999	Used only in each program	"Local flags"



1.5 Variables

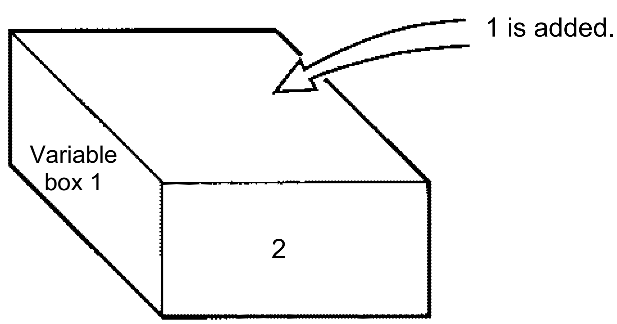
(1) Meaning of variable

"Variable" is a technical term used in software programming. Simply put, it means "a box in which a value is put." Variables can be used in many ways, such as putting in or taking out a value and performing addition or subtraction.



Command	Operand 1	Operand 2
ADD	1	1

If this command is applied to variable box 1, which already contains 2, then 1 will be added to the current value and 3 will result.

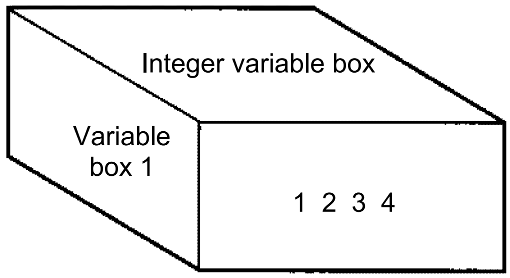


(Already contains 2)

(2) Types of variables

Variables are classified into two types, as follows:

[1] Integer variables
These variables cannot handle decimal places.
[Example] 1234

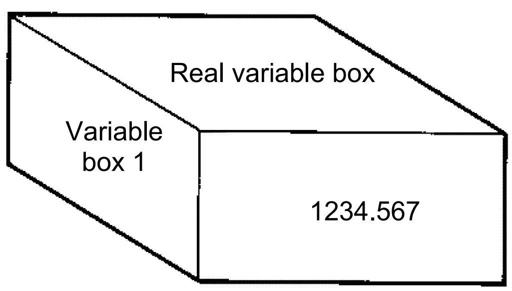


Integer variable number	200 ~ 299 1200 ~ 1299	Can be used in all programs	"Global integer variables"
Integer variable number	1 ~ 99 1001 ~ 1099	Used only in each program	"Local integer variables"

Caution

Integer 99 is a special register this system uses in integer operations. Any value in the range from –9,999,999 to 99,999,999 can be input in programs.

[2] Real variables
Actual values. These variables can handle decimal places.
[Example] 1234.567
(Decimal point)



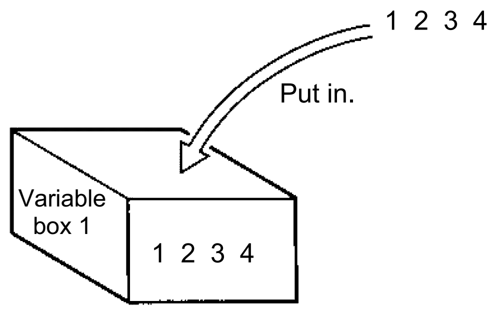
Real variable number	300 ~ 399 1300 ~ 1399	Can be used in all programs	"Global real variables"
Real variable number	100 ~ 199 1100 ~ 1199	Used only in each program	"Local real variables"

Caution

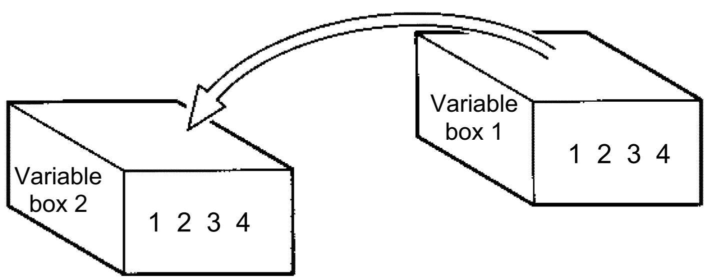
Real number 199 is a special register this system uses in real-number operations. Any value in the range from –99,999.9 to 999,999.9 (eight digits including a sign) can be input in programs.

[3] Variables with "*" (asterisk) (indirect specification)
An "*" (asterisk) is used to specify a variable.
In the following example, the content of variable box 1 will be put in variable box 2. If variable box 1 contains "1234," then "1234" will be put in variable box 2.

Command	Operand 1	Operand 2
LET	1	1234



Command	Operand 1	Operand 2
LET	2	*1



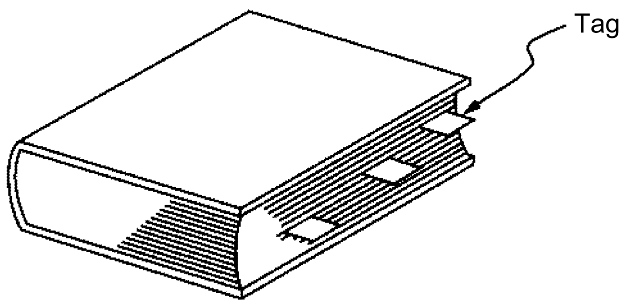
The above use of variables is called "indirect specification."

An "*" is also used when indirectly specifying a symbol variable (refer to 1.8, "Symbols").

Command	Operand 1	Operand 2	
LET	ABC	1	Put 1 in variable ABC.
LET	BCD	2	Put 2 in variable BCD.
ADD	ABC	*BCD	Add the content of variable BCD, or 2, to variable ABC. (The content of variable ABC becomes 3.)

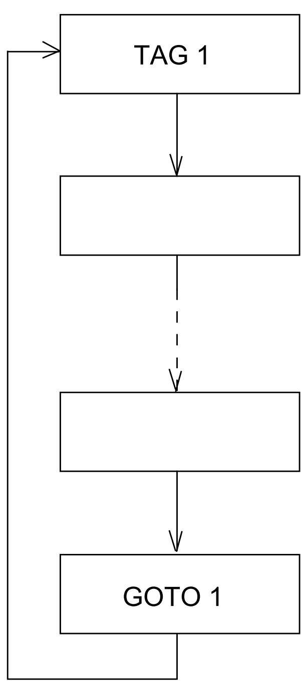
1.6 Tags

The term “tag” means “heading.”
Tags are used in the same way you attach labels to the pages in a book you want to reference frequently.
A tag is a destination specified in a jump command “GOTO.”



Command	Operand 1
TAG	Tag number (Integer between 1 and 256)

They are used only in each program.



1.7 Subroutines

By taking out the parts of a program that are used repeatedly and registering them as “subroutines,” the same processing can be performed with fewer steps. (A maximum of 15 nests are accommodated.)

They are used only in each program.

Command	Operand 1
EXSR	Subroutine number (Integer between 1 and 99; variable is also supported)

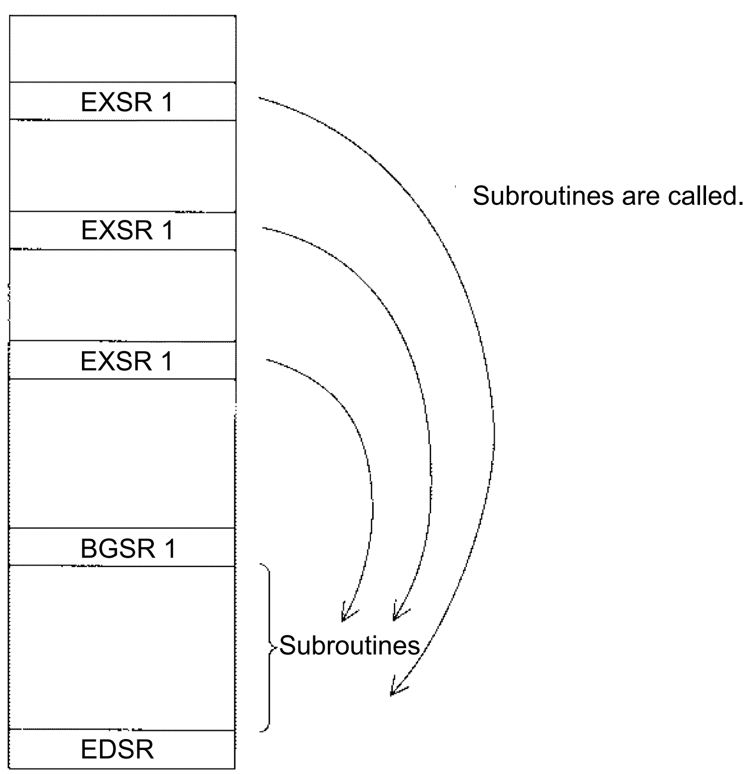
Subroutine execution command

Command	Operand 1
BGSR	Subroutine number (Integer between 1 and 99)

Subroutine start declaration

Command	Operand 1
EDSR	—

Subroutine end declaration



1.8 Symbols

In the ASEL Controller, values such as variable numbers and flag numbers can be handled as symbols. For the method to edit symbols, refer to "Editing Symbols" in the operation manual for PSEL teaching pendant or "Symbol Edit Window" in the operation manual for PSEL PC software.

(1) Supported symbols
The following items can be expressed using symbols:
Variable number, flag number, tag number, subroutine number, program number, position number, input port number, output port number, axis number, constant

(2) Description rules of symbols
[1] A maximum of nine single-byte alphanumeric characters or underscore starting with an alphabet (Note: The length of a character-string literal must not exceed eight single-byte characters.)
* Exercise caution that the same ASCII code may be expressed differently between the PC software and the teaching pendant because of the different fonts used by the two. (The same applies to character-string literals.)
5Ch --- PC software: Backslash \ (overseas specifications, etc.)
Teaching pendant: Yen mark ¥
7Eh --- PC software: ~
Teaching pendant: Right arrow →
[2] Symbols of the same name must not be defined within each function. (The same local symbol can be used in different programs.)
[3] Symbols of the same name must not be defined within the flag number, input-port number or output-port number group. (The same local symbol can be used in different programs.)
[4] Symbols of the same name must not be defined within the integer-variable number or real-variable number group. (The same local symbol can be used in different programs.)
[5] Symbols of the same name must not be defined within the integer constant or real constant group.

(3) Number of symbols that can be defined: Maximum 500

(4) Number of times symbols can be used in all SEL programs: Maximum 2500 times including character-string literals
* If symbol is used in all of the input condition, operand 1, operand 2 and output fields, it is deemed that symbol is used four times in one step.

1.9 Character-String Literals

Character-string literals are used in certain string-operation commands and consist of the portion enclosed by single quotation marks (' ') (maximum eight single-byte characters).
With the PC software, single-byte ASCII code characters from 20h to 7Eh (limited to those that can be input via keyboard) can be used inside the single quotation marks. With the teaching pendant, single-byte alphanumeric characters and single-byte underscores can be used.

1.10 Axis Specification

Axes can be specified based on axis number or axis pattern.

(1) Axis numbers and how axes are stated
Each of multiple axes is stated as follows:

Axis number	How axis is stated
1	Axis 1
2	Axis 2

The axis numbers stated above can also be expressed using symbols.

Use axis number if you wish to specify only one of multiple axes.

- Commands that use axis specification based on axis number
  BASE, PPUT, PGET, ACHZ, AXST, PASE, PARG, PRDQ, ECMD (1.5)

(2) Axis pattern
Whether or not each axis will be used is indicated by "1" or "0."

(Upper) (Lower)

Axis number	Axis 2	Axis 1
Used	1	1
Not used	0	0

[Example] When axes 1 and 2 are used

Axis 2
↓
11
↑
Axis 1

[Example] When axes 2 is used

Axis 2
↓
10 (In this case, the 0s are needed to indicate the position of axis 2.)

Indirect specification of axis pattern in a variable
The axis pattern is considered a binary value, and a converted decimal value is assigned to a variable.

[Example] To perform home return for axis 2 only, you can specify as follows based on axis pattern:

```
HOME     10
```

In indirect specification, 10 (binary) is expressed as 2 (decimal), so the same operation can be specified as follows:

```
LET      6   2
HOME     *6
```

If you must select and specify multiple axes at the same time, use axis pattern.

- Commands that use axis specification based on axis pattern
  OFST, GRP, SVON, SVOF, HOME, JFWN, JFWF, JBWN, JBWF, STOP, PTST, PRED
  CHVL, PBND, WZNA, WZNO, WZFA, WZFO, MOVD, MVDI, PTRQ

SEL language consists of a position part (position data = coordinates, etc.) and a command part (application program).

2. Position Part

As position data, coordinates, speeds, accelerations and decelerations are set and stored.

± 2000000.000 mm

*1, 2
1 ~ 2000/mmsec

*2
Standard
0.3 G

*2
Standard
0.3 G

Position No.	Axis 1	Axis 2	Speed	Acceleration	Deceleration
1					
2					
3					
|					
|					
|					
1498					
1499					
1500					

*1 Varies depending on the actuator model.
*2 If a speed, acceleration or deceleration is set in the position data, the applicable setting takes precedence over the corresponding data specified in the application program, as shown in the priority table below. Leave the position data fields empty if you wish to enable the corresponding data in the application program.

Priority	Speed	Acceleration (deceleration)
1	Setting corresponding to the position data specified by operand 1	Setting corresponding to the position data specified by operand 1
2	Setting by a VEL command	Setting by an ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

Values pertaining to a rotating axis are processed in degrees instead of millimeters.

If axis-specific parameter No. 1 (axis operation type) is set to “1” (rotational movement axis (angle control)) for a given axis, all millimeter values pertaining to that axis (including parameters, etc.) will be processed in degrees.

If the gear ratio parameters (axis-specific parameter Nos. 50 and 51) are set correctly, the angles (deg) will represent those of the body of rotation at the end.

Example)	Distance	1 mm → 1 deg
	Speed	1 mm/sec → 1 deg/sec
	Acceleration/deceleration	1 G = 9807 mm/sec^2 → 9807 deg/sec^2

3. Command Part

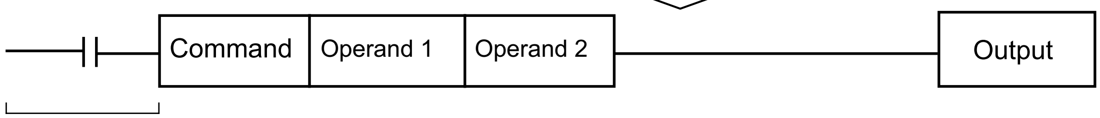
The primary feature of SEL language is its very simple command structure. Since the structure is simple, there is no need for a compiler (to translate into computer language) and high-speed operation is possible via an interpreter (the program runs as commands are translated).

3.1 SEL language Structure

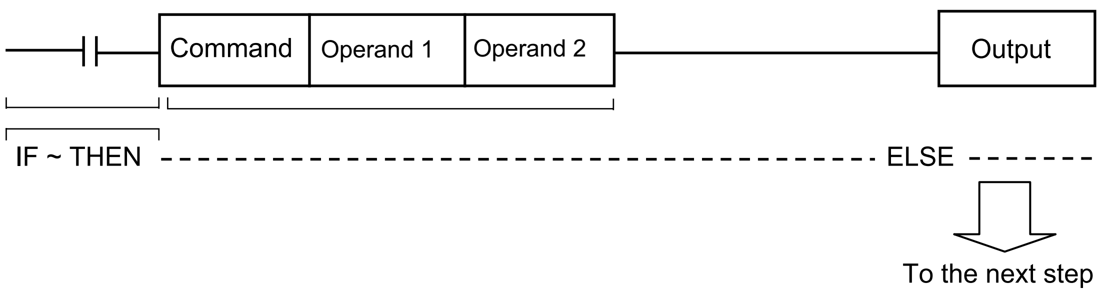
The table below shows the structure of one command step.

Extension condition (AND, OR)	Input condition (I/O, flag)	Command, declaration			Output (Output port, flag)
		Command, declaration	Operand 1	Operand 2	

Using a ladder diagram, this is expressed as follows:

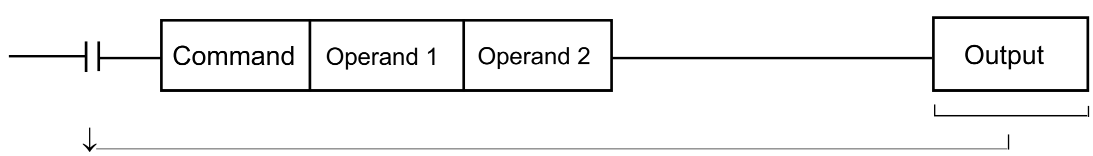


(1) The condition before the command is equivalent to "IF ~ THEN…" in BASIC.



1. If the input condition is satisfied, the command will be executed. If there is an output specification, the specified output port will be turned ON. If the input condition is not satisfied, the program will proceed to the next step regardless of the command that follows (e.g., WTON, WTOF). Obviously nothing will happen at the output port, but caution must be exercised.
2. If no condition is set, the command will be executed unconditionally.
3. To use the condition in reverse logic (so-called "contact b logic"), add "N" (NOT) to the condition.
4. The input condition supports input port, output port and flag.
5. The operand 1, operand 2 and output fields can be specified indirectly.

(2) The output field, which follows the command, operand 1 and operand 2 fields, will specify the following action:

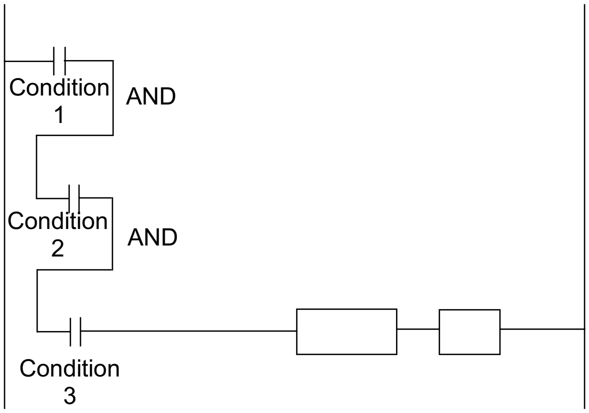


1. In the case of a control command relating to actuator operation, etc., the output will turn OFF the moment the execution of command is started, and turn ON when the execution is completed. In the case of a calculation operation command, etc., the output will turn ON if the result corresponds to a certain value, and turn OFF if not.
2. The output field supports output port and flag.

3.2 Extension Condition

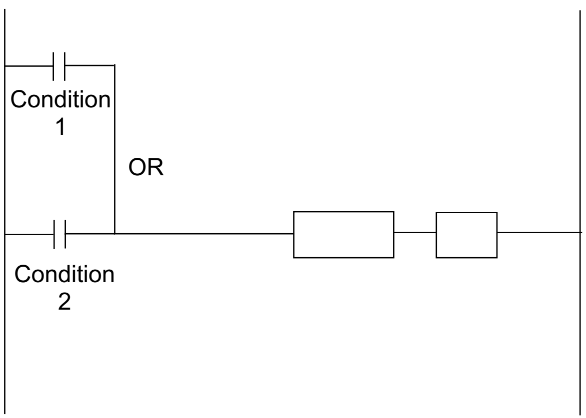
Conditions can be combined in a complex manner.

AND extension (Ladder diagram) (SEL language)



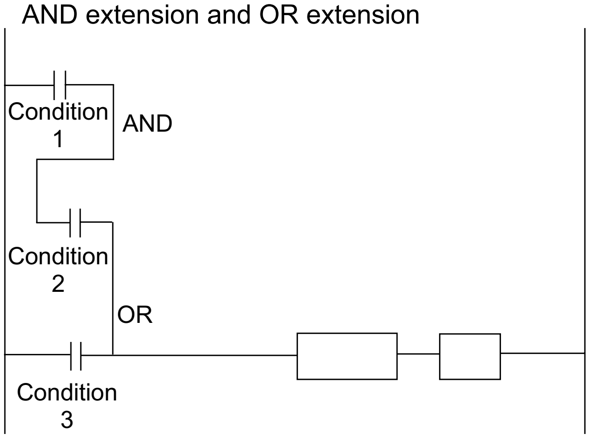
Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
A	Condition 2				
A	Condition 3	Command	Operand 1	Operand 2	

OR extension



Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
O	Condition 2	Command	Operand 1	Operand 2	

AND extension and OR extension



Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
A	Condition 2				
O	Condition 3	Command	Operand 1	Operand 2	

Chapter 2 List of SEL Language Command Codes

1. By Function

Variables can be specified indirectly in the operand 1, operand 2 and output fields.
Symbols can be input in the condition, operand 1, operand 2 and output fields.
The input items in ( ) under operand 1 and operand 2 are optional.

Once an "actuator control declaration" command is executed in a program, the command will remain valid as long as the program is running. To change the values (in operand 1, operand 2, etc.) already set by the "actuator control declaration" command, the necessary parts of the program must be set again. In other words, the values set by the last executed command will prevail.

The output field will be turned OFF when the command is executed. Once the execution is completed, the output field may be turned ON depending on the operation type condition in the output field. (The output field will remain OFF if the condition is not satisfied.)

Note: The output field of a comparison command CP☐☐ (CPEQ, CPNE, CPGT, CPGE, CPLT and CPLE) will not be turned OFF when the command is executed.

Operation type in the output field
CC: Command was executed successfully,
ZR: Operation result is zero, PE: Operation is complete,
CP: Command part has passed, TU: Time up

EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Variable assignment	Optional	LET	Assignment variable	Assigned value	ZR	Assign	95
	Optional	TRAN	Copy-destination variable	Copy-source variable	ZR	Copy	96
	Optional	CLR	Start-of-clear variable	End-of-clear variable	ZR	Clear variable	97
Arithmetic operation	Optional	ADD	Augend variable	Addend	ZR	Add	98
	Optional	SUB	Minuend variable	Subtrahend	ZR	Subtract	98
	Optional	MULT	Multiplicand variable	Multiplier	ZR	Multiply	99
	Optional	DIV	Dividend variable	Divisor	ZR	Divide	99
	Optional	MOD	Remainder assignment variable	Divisor	ZR	Calculate remainder	100
Function operation	Optional	SIN	Sine assignment variable	Operand [radian]	ZR	Sine	101
	Optional	COS	Cosine assignment variable	Operand [radian]	ZR	Cosine	101
	Optional	TAN	Tangent assignment variable	Operand [radian]	ZR	Tangent	102
	Optional	ATN	Inverse-tangent assignment operation	Operand	ZR	Inverse tangent	102
	Optional	SQR	Root assignment variable	Operand	ZR	Root	103
Logical operation	Optional	AND	AND operand variable	Operand	ZR	Logical AND	104
	Optional	OR	OR operand variable	Operand	ZR	Logical OR	105
	Optional	EOR	Exclusive-OR operand variable	Operand	ZR	Logical exclusive-OR	106
Comparison	Optional	CP☐☐	Comparison variable	Comparison value	EQ, NE, GT, GE, LT, LE	Compare	107
Timer	Optional	TIMW	Wait time (sec)	Prohibited	TU	Wait	108
	Optional	TIMC	Program number	Prohibited	CP	Cancel waiting	109
	Optional	GTTM	Time assignment variable	Prohibited	CP	Get time	110
I/O, flag operation	Optional	BT☐☐	Start output, flag	(End output, flag)	CP	Output, flag [ON, OF, NT]	111
	Optional	BTPN	Output port, flag	Timer setting	CP	Output ON pulse	112
	Optional	BTPF	Output port, flag	Timer setting	CP	Output OFF pulse	113
	Optional	WT☐☐	I/O, flag	(Wait time)	TU	Wait for I/O, flag [ON, OF]	114
	Optional	IN	Head I/O, flag	End I/O, flag	CC	Input binary (32 bits max.)	115
	Optional	INB	Head I/O, flag	Conversion digits	CC	Input BCD (8 digits max.)	116
	Optional	OUT	Head output, flag	End I/O, flag	CC	Output binary (32 bits max.)	117
	Optional	OUTB	Head output, flag	Conversion digits	CC	Output BCD (8 digits max.)	118
	Optional	FMIO	Format type	Prohibited	CP	Set IN (B)/OUT (B) command format	119

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Program control	Optional	GOTO	Jump-destination tag number	Prohibited	CP	Jump	122
	Prohibited	TAG	Declaration tag number	Prohibited	CP	Declare jump destination	122
	Optional	EXSR	Execution subroutine number	Prohibited	CP	Execute subroutine	123
	Prohibited	BGSR	Declaration subroutine number	Prohibited	CP	Start subroutine	123
	Prohibited	EDSR	Prohibited	Prohibited	CP	End subroutine	124
Task management	Optional	EXIT	Prohibited	Prohibited	CP	End program	125
	Optional	EXPG	Execution program number	(Execution program number)	CC	Start program	126
	Optional	ABPG	Stop program number	(Stop program number)	CC	Stop other program	127
	Optional	SSPG	Pause program number	(Pause program number)	CC	Pause program	128
	Optional	RSPG	Resumption program number	(Resumption program number)	CC	Resume program	129
Position operation	Optional	PGET	Axis number	Position number	CC	Assign position to variable 199	130
	Optional	PPUT	Axis number	Position number	CP	Assign value of variable 199	131
	Optional	PCLR	Start position number	End position number	CP	Clear position data	132
	Optional	PCPY	Copy-destination position number	Copy-source position number	CP	Copy position data	133
	Optional	PRED	Read axis pattern	Save-destination position number	CP	Read current axis position	134
	Optional	PRDQ	Axis number	Variable number	CP	Read current axis position (1 axis direct)	135
	Optional	PTST	Confirmation axis pattern	Confirmation position number	CC	Confirm position data	136
	Optional	PVEL	Speed [mm/sec]	Assignment-destination position number	CP	Assign position speed	137
	Optional	PACC	Acceleration [G]	Assignment-destination position number	CP	Assign position acceleration	138
	Optional	PDCL	Deceleration [G]	Assignment-destination position number	CP	Assign position deceleration	139
	Optional	PAXS	Axis-pattern assignment variable number	Position number	CP	Read axis pattern	140
	Optional	PSIZ	Size assignment variable number		CP	Confirm position size	141
	Optional	GVEL	Variable number	Position number	CP	Get speed data	142
	Optional	GACC	Variable number	Position number	CP	Get acceleration data	143
	Optional	GDCL	Variable number	Position number	CP	Get deceleration data	144
Actuator control declaration	Optional	VEL	Speed [mm/sec]	Prohibited	CP	Set speed	145
	Optional	OVRD	Speed ratio [%]	Prohibited	CP	Set speed coefficient	146
	Optional	ACC	Acceleration [G]	Prohibited	CP	Set acceleration	147
	Optional	DCL	Deceleration [G]	Prohibited	CP	Set deceleration	148
	Optional	SCRV	Ratio [%]	Prohibited	CP	Set sigmoid motion ratio	149
	Optional	OFST	Setting axis pattern	Offset value [mm]	CP	Set offset	150
	Optional	DEG	Division angle [deg]	Prohibited	CP	Set division angle	151
	Optional	BASE	Reference axis number	Prohibited	CP	Set reference axis	152
	Optional	GRP	Valid axis pattern	Prohibited	CP	Set group axes	153
	Optional	HOLD	(Input port to pause)	(HOLD type)	CP	Declare port to pause	154
	Optional	CANC	(Input port to abort)	(CANC type)	CP	Declare port to abort	155
	Optional	VLMX	Prohibited	Prohibited	CP	Specify VLMX speed	156
	Optional	DIS	Distance	Prohibited	CP	Set spline division distance	157
	Optional	POTP	0 or 1	Prohibited	CP	Set PATH output type	158
	Optional	PAPR	Distance	Speed	CP	Set PUSH command distance, speed	159
	Optional	QRTN	0 or 1	Prohibited	CP	Set quick-return mode	160

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Actuator control command	Optional	SV☐☐	Operation axis pattern	Prohibited	PE	Servo [ON, OF]	161
	Optional	HOME	Home-return axis pattern	Prohibited	PE	Return to home	162
	Optional	MOVP	Destination position number	Prohibited	PE	Move to specified position	163
	Optional	MOVL	Destination position number	Prohibited	PE	Move to specified position via interpolation	164
	Optional	MVPI	Travel position number	Prohibited	PE	Move to relative position	165
	Optional	MVLI	Travel position number	Prohibited	PE	Move to relative position via interpolation	166
	Optional	MOVD	Target position	(Axis pattern)	PE	Move via direct value specification	167
	Optional	MVDI	Travel distance	(Axis pattern)	PE	Move relatively via direct value specification	168
	Optional	PATH	Start position number	End position number	PE	Move along path	169
	Optional	J☐W☐	Axis operation pattern	Start I/O, flag	PE	Jog [FN, FF, BN, BF]	170
	Optional	STOP	Axis stop pattern	Prohibited	CP	Decelerate and stop axis	171
	Optional	PSPL	Start position number	End position number	PE	Move along spline	172
	Optional	PUSH	Target position number	Prohibited	PE	Move by push motion	173
	Optional	PTRQ	Axis pattern	Ratio [%]	CC	Change push torque limit parameter	175
	Optional	CIR2	Passing position 1 number	Passing position 2 number	PE	Move along circle 2 (arc interpolation)	176
	Optional	ARC2	Passing position number	End position number	PE	Move along arc 2 (arc interpolation)	177
	Optional	CHVL	Axis pattern	Speed	CP	Change speed	178
	Optional	ARCD	End position number	Center angle [deg]	PE	Move along arc via specification of end position and center angle	179
	Optional	ARCC	Center position number	Center angle [deg]	PE	Move along arc via specification of center position and center angle	180
	Optional	PBND	Axis pattern	Distance	CP	Set positioning band	181
	Optional	CIR	Passing position 1 number	Passing position 2 number	PE	Move along circle (CIR2 is recommended)	182
	Optional	ARC	Passing position number	End position number	PE	Move along arc (ARC2 is recommended)	183
	Refer to the page on palletizing for commands relating to arch motion.						
	Optional	ARCH	Position number	Position number	PE	Arch motion	216
	Optional	ACHZ	Axis number	Prohibited	CP	Declare arch-motion Z-axis	218
	Optional	ATRG	Position number	Position number	CP	Set arch trigger	219
	Optional	OFAZ	Offset value	Prohibited	CP	Set arch-motion Z-axis offset	220
Structural IF	Optional	IF☐☐	Comparison variable	Comparison value	CP	Compare [EQ, NE, GT, GE, LT, LE]	184
	Optional	IS☐☐	Column number	Column number, character literal	CP	Compare strings	185
	Prohibited	ELSE	Prohibited	Prohibited	CP	Declare execution destination when IF command condition is not satisfied	186
	Prohibited	EDIF	Prohibited	Prohibited	CP	Declare end of IF	186
Structural DO	Optional	DW☐☐	Comparison variable	Comparison value	CP	Loop [EQ, NE, GT, GE, LT, LE]	187
	Optional	LEAV	Prohibited	Prohibited	CP	Pull out from DO	187
	Optional	ITER	Prohibited	Prohibited	CP	Repeat DO	188
	Prohibited	EDDO	Prohibited	Prohibited	CP	Declare end of DO	188
Multi-branching	Optional	SLCT	Prohibited	Prohibited	CP	Declare start of multi-branching	189
	Prohibited	WH☐☐	Comparison variable	Comparison value	CP	Branch value [EQ, NE, GT, GE, LT, LE]	190
	Prohibited	WS☐☐	Column number	Column number, character literal	CP	Branch character string [EQ, NE]	191
	Prohibited	OTHE	Prohibited	Prohibited	CP	Declare branching destination when condition is not satisfied	192
	Prohibited	EDSL	Prohibited	Prohibited	CP	Declare end of SLCT	192

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
System information acquisition	Optional	AXST	Variable number	Axis number	CP	Get axis status	193
	Optional	PGST	Variable number	Program number	CP	Get program status	194
	Optional	SYST	Variable number	Prohibited	CP	Get system status	195
Zone	Optional	WZNA	Zone number	Axis pattern	CP	Wait for zone ON, with AND	196
	Optional	WZNO	Zone number	Axis pattern	CP	Wait for zone ON, with OR	197
	Optional	WZFA	Zone number	Axis pattern	CP	Wait for zone OFF, with AND	198
	Optional	WZFO	Zone number	Axis pattern	CP	Wait for zone OFF, with OR	199
Communication	Optional	OPEN	Channel number	Prohibited	CP	Open channel	200
	Optional	CLOS	Channel number	Prohibited	CP	Close channel	200
	Optional	READ	Channel number	Column number	CC	Read from channel	201
	Optional	TMRW	Read timer setting	(Write timer setting)	CP	Set READ timeout value	203
	Optional	WRIT	Channel number	Column number	CC	Output to channel	205
	Optional	SCHA	Character code	Prohibited	CP	Set end character	206
String operation	Optional	SCPY	Column number	Column number, character literal	CC	Copy character string	207
	Optional	SCMP	Column number	Column number, character literal	EQ	Compare character strings	208
	Optional	SGET	Variable number	Column number, character literal	CP	Get character	209
	Optional	SPUT	Column number	Data	CP	Set character	210
	Optional	STR	Column number	Data	CC	Convert character string; decimal	211
	Optional	STRH	Column number	Data	CC	Convert character string; hexadecimal	212
	Optional	VAL	Variable number	Column number, character literal	CC	Convert character string data; decimal	213
	Optional	VALH	Variable number	Column number, character literal	CC	Convert character string data; hexadecimal	214
	Optional	SLEN	Character string length	Prohibited	CP	Set length	215

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Palletizing-related	Optional	ARCH	Position number	Position number	PE	Arch motion	216
	Optional	ACHZ	Axis number	Prohibited	CP	Declare arch-motion Z-axis	218
	Optional	ATRG	Position number	Position number	CP	Set arch triggers	219
	Optional	OFAZ	Offset amount	Prohibited	CP	Set arch-motion Z-axis offset	220
	Optional	BGPA	Palletizing number	Prohibited	CP	Declare start of palletizing setting	221
	Prohibited	EDPA	Prohibited	Prohibited	CP	Declare end of palletizing setting	221
	Optional	PAPI	Count	Count	CP	Set palletizing counts	222
	Optional	PAPN	Pattern number	Prohibited	CP	Set palletizing pattern	222
	Optional	PASE	Axis number	Axis number	CP	Set palletizing axes	223
	Optional	PAPT	Pitch	Pitch	CP	Set palletizing pitches	223
	Optional	PAST	(Position number)	Prohibited	CP	Set palletizing reference point	224
	Optional	PAPS	Position number	(Palletizing position setting type)	CP	Set 3 palletizing points for teaching	225
	Optional	PSLI	Offset amount	(Count)	CP	Set zigzag	227
	Optional	PTNG	Palletizing number	Variable number	CP	Get palletizing position number	228
	Optional	PINC	Palletizing number	Prohibited	CC	Increment palletizing position number by 1	228
	Optional	PDEC	Palletizing number	Prohibited	CC	Decrement palletizing position number by 1	229
	Optional	PSET	Palletizing number	Data	CC	Set palletizing position number directly	229
	Optional	PARG	Palletizing number	Axis number	CP	Get palletizing angle	230
	Optional	PAPG	Palletizing number	Position number	CP	Get palletizing calculation data	230
	Optional	PMVP	Palletizing number	(Position number)	PE	Move to palletizing points via PTP	231
	Optional	PMVL	Palletizing number	(Position number)	PE	Move to palletizing points via interpolation	232
Building of pseudo-ladder task	Extension conditions LD (LOAD), A (AND), O (OR), AB (AND BLOCK) and OB (OR BLOCK) are supported.						
	Optional	CHPR	0 or 1	Prohibited	CP	Change task level	233
	Prohibited	TPCD	0 or 1	Prohibited	CP	Specify processing to be performed when input condition is not specified	233
	Prohibited	TSLP	Time	Prohibited	CP	Task sleep	234
	Optional	OUTR	Output, flag number	Prohibited	CP	Output relay for ladder	See 251
	Optional	TIMR	Local flag number	Timer setting	CP	Timer relay for ladder	See 251
Extended command	Optional	ECMD	1	Axis number	CC	Get motor current value	235
	Optional	ECMD	5	Axis number	CC	Get axis operation status	236
	Optional	ECMD	20	Variable number	CC	Get parameter value	237

2. Alphabetical Order

Operation type in the output field
CC: Command was executed successfully,
ZR: Operation result is zero, PE: Operation is complete,
CP: Command part has passed, TU: Time up

EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Command	Page	Condition	Operand 1	Operand 2	Output	Function
A						
ABPG	127	Optional	Stop program number	(Stop program number)	CC	Stop other program
ACC	147	Optional	Acceleration	Prohibited	CP	Set acceleration
ACHZ	216	Optional	Axis number	Prohibited	CP	Declare arch-motion Z-axis
ADD	98	Optional	Augend variable	Addend	ZR	Add
AND	104	Optional	AND operand variable	Operand	ZR	Logical AND
ARC	183	Optional	Passing position number	End position number	PE	Move along arc
ARC2	177	Optional	Passing position number	End position number	PE	Move along arc 2
ARCC	180	Optional	Center position number	Center angle	PE	Move along arc via specification of center position and center angle
ARCD	179	Optional	End position number	Center angle	PE	Move along arc via specification of end position and center angle
ARCH	216	Optional	Position number	Position number	PE	Arch motion
ATN	102	Optional	Inverse-tangent assignment operation	Operand	ZR	Inverse tangent
ATRG	219	Optional	Position number	Position number	CP	Set arch trigger
AXST	193	Optional	Variable number	Axis number	CP	Get axis status
B						
BASE	152	Optional	Reference axis number	Prohibited	CP	Set reference axis
BGPA	221	Optional	Palletizing number	Prohibited	CP	Declare start of palletizing setting
BGSR	123	Prohibited	Declaration subroutine number	Prohibited	CP	Start subroutine
BTPF	113	Optional	Output port, flag	Timer setting	CP	Output OFF pulse
BTPN	112	Optional	Output port, flag	Timer setting	CP	Output ON pulse
BT□□	111	Optional	Start output, flag	(End output, flag)	CP	Output, flag [ON, OF, NT]
C						
CANC	155	Optional	(Input port to abort)	(CANC type)	CP	Declare port to abort
CHPR	233	Optional	0 or 1	Prohibited	CP	Change task level
CHVL	178	Optional	Axis pattern	Speed	CP	Change speed
CIR	182	Optional	Passing position 1 number	Passing position 2 number	PE	Move along circle
CIR2	176	Optional	Passing position 1 number	Passing position 2 number	PE	Move along circle 2
CLOS	200	Optional	Channel number	Prohibited	CP	Close channel
CLR	97	Optional	Start-of-clear variable	End-of-clear variable	ZR	Clear variable
COS	101	Optional	Cosine assignment variable	Operand	ZR	Cosine
CP□□	107	Optional	Comparison variable	Comparison value	EQ NE GT GE LT LE	Compare
D						
DCL	148	Optional	Deceleration	Prohibited	CP	Set deceleration
DEG	151	Optional	Division angle	Prohibited	CP	Set division angle
DIS	157	Optional	Distance	Prohibited	CP	Set spline division distance
DIV	99	Optional	Dividend variable	Divisor	ZR	Divide
DW□□	187	Optional	Comparison variable	Comparison value	CP	Loop [EQ, NE, GT, GE, LT, LE]
E						
ECMD	235	Optional	1	Axis number	CC	Get motor current value
ECMD	236	Optional	5	Axis number	CC	Get axis operation status
ECMD	237	Optional	20	Variable number	CC	Get parameter value
EDDO	188	Prohibited	Prohibited	Prohibited	CP	Declare end of DO
EDIF	186	Prohibited	Prohibited	Prohibited	CP	Declare end of IF
EDPA	221	Prohibited	Prohibited	Prohibited	CP	Declare end of palletizing setting
EDSL	192	Prohibited	Prohibited	Prohibited	CP	Declare end of SLCT

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Command	Page	Condition	Operand 1	Operand 2	Output	Function
EDSR	124	Prohibited	Prohibited	Prohibited	CP	End subroutine
ELSE	186	Prohibited	Prohibited	Prohibited	CP	Declare execution destination when IF command condition is not satisfied
EOR	106	Optional	Exclusive-OR operand variable	Operand	ZR	Logical exclusive-OR
EXIT	125	Optional	Prohibited	Prohibited	CP	End program
EXPG	126	Optional	Execution program number	(Execution program number)	CC	Start program
EXSR	123	Optional	Execution subroutine number	Prohibited	CP	Execute subroutine
F						
FMIO	119	Optional	Format type	Prohibited	CP	Set IN (B)/OUT (B) command format
G						
GACC	143	Optional	Variable number	Position number	CP	Get acceleration data
GDCL	144	Optional	Variable number	Position number	CP	Get deceleration data
GOTO	122	Optional	Jump-destination tag number	Prohibited	CP	Jump
GRP	153	Optional	Valid axis pattern	Prohibited	CP	Set group axes
GTTM	110	Optional	Time assignment variable	Prohibited	CP	Get time
GVEL	142	Optional	Variable number	Position number	CP	Get speed data
H						
HOLD	154	Optional	(Input port to pause)	(HOLD type)	CP	Declare port to pause
HOME	162	Optional	Home-return axis pattern	Prohibited	PE	Return to home
I						
IF□□	184	Optional	Comparison variable	Comparison value	CP	Compare [EQ, NE, GT, GE, LT, LE]
INB	116	Optional	Head I/O, flag	Conversion digits	CC	Input BCD (8 digits max.)
IN	115	Optional	Head I/O, flag	End I/O, flag	CC	Input binary (32 bits max.)
IS□□	185	Optional	Column number	Column number, character literal	CP	Compare strings
ITER	188	Optional	Prohibited	Prohibited	CP	Repeat DO
J						
J□W□	170	Optional	Axis operation pattern	Start I/O, flag	PE	Jog [FN, FF, BN, BF]
L						
LEAV	187	Optional	Prohibited	Prohibited	CP	Pull out from DO
LET	95	Optional	Assignment variable	Assigned value	ZR	Assign
M						
MOD	100	Optional	Remainder assignment variable	Divisor	ZR	Calculate remainder
MOVD	167	Optional	Target position	(Axis pattern)	PE	Move via direct value specification
MOVL	164	Optional	Destination position number	Prohibited	PE	Move to specified position via interpolation
MOVP	163	Optional	Destination position number	Prohibited	PE	Move to specified position
MULT	99	Optional	Multiplicand variable	Multiplier	ZR	Multiply
MVDI	168	Optional	Travel distance	(Axis pattern)	PE	Move relatively via direct value specification
MVLI	166	Optional	Travel position number	Prohibited	PE	Move to relative position via interpolation
MVPI	165	Optional	Travel position number	Prohibited	PE	Move to relative position

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Command	Page	Condition	Operand 1	Operand 2	Output	Function
O						
OFAZ	220	Optional	Offset amount	Prohibited	CP	Set arch-motion Z-axis offset
OFST	150	Optional	Setting axis pattern	Offset value	CP	Set offset
OPEN	200	Optional	Channel number	Prohibited	CP	Open channel
OR	105	Optional	OR operand variable	Operand	ZR	Logical OR
OTHE	192	Prohibited	Prohibited	Prohibited	CP	Declare branching destination when condition is not satisfied
OUT	117	Optional	Head output, flag	End I/O, flag	CC	Output binary (32 bits max.)
OUTB	118	Optional	Head output, flag	Conversion digits	CC	Output BCD (8 digits max.)
OUTR	251	Optional	Output, flag number	Prohibited	CP	Output relay for ladder
OVRD	146	Optional	Speed ratio	Prohibited	CP	Set speed ratio
P						
PACC	138	Optional	Acceleration	Assignment-destination position number	CP	Assign position acceleration
PAPG	230	Optional	Palletizing number	Position number	CP	Get palletizing calculation data
PAPI	222	Optional	Count	Count	CP	Set palletizing counts
PAPN	222	Optional	Pattern number	Prohibited	CP	Set palletizing pattern
PAPR	159	Optional	Distance	Speed	CP	Set PUSH command distance, speed
PAPS	225	Optional	Position number	(Palletizing position setting type)	CP	Set 3 palletizing points for teaching
PAPT	223	Optional	Pitch	Pitch	CP	Set palletizing pitches
PARG	230	Optional	Palletizing number	Axis number	CP	Get palletizing angle
PASE	223	Optional	Axis number	Axis number	CP	Set palletizing axes
PAST	224	Optional	(Position number)	Prohibited	CP	Set palletizing reference point
PATH	169	Optional	Start position number	End position number	PE	Move along path
PAXS	140	Optional	Axis-pattern assignment variable number	Position number	CP	Read axis pattern
PBND	181	Optional	Axis pattern	Distance	CP	Set positioning band
PCLR	132	Optional	Start position number	End position number	CP	Clear position data
PCPY	133	Optional	Copy-destination position number	Copy-source position number	CP	Copy position data
PDCL	139	Optional	Deceleration	Assignment-destination position number	CP	Assign position deceleration
PDEC	229	Optional	Palletizing number	Prohibited	CC	Decrement palletizing position number by 1
PGET	130	Optional	Axis number	Position number	CC	Assign position to variable 199
PGST	194	Optional	Variable number	Program number	CP	Get program status
PMVL	232	Optional	Palletizing number	Prohibited	PE	Move to palletizing points via interpolation
PMVP	231	Optional	Palletizing number	Prohibited	PE	Move to palletizing points via PTP
POTP	158	Optional	0 or 1	Prohibited	CP	Set PATH output type
PPUT	131	Optional	Axis number	Position number	CP	Assign value of variable 199
PRDQ	135	Optional	Axis number	Variable number	CP	Read current axis position (1 axis direct)
PRED	134	Optional	Read axis pattern	Save-destination position number	CP	Read current axis position
PSET	229	Optional	Palletizing number	Data	CC	Set palletizing position number directly
PSIZ	141	Optional	Size assignment variable number		CP	Confirm position size
PSLI	227	Optional	Offset amount	(Count)	CP	Set zigzag
PSPL	172	Optional	Start position number	End position number	PE	Move along spline
PTRQ	175	Optional	Axis pattern	Ratio	CC	Change push torque limit parameter
PTST	136	Optional	Confirmation axis pattern	Confirmation position number	CP	Confirm position data
PUSH	173	Optional	Target position number	Prohibited	PE	Move by push motion
PVEL	137	Optional	Speed	Assignment-destination position number	CP	Assign position speed

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Command	Page	Condition	Operand 1	Operand 2	Output	Function
Q						
QRTN	160	Optional	0 or 1	Prohibited	CP	Set quick-return mode
R						
READ	201	Optional	Channel number	Column number	CC	Read from channel
RSPG	129	Optional	Resumption program number	(Resumption program number)	CC	Resume program
S						
SCHA	206	Optional	Character code	Prohibited	CP	Set end character
SCMP	208	Optional	Column number	Column number, character literal	EQ	Compare character strings
SCPY	207	Optional	Column number	Column number, character literal	CC	Copy character string
SCRV	149	Optional	Ratio	Prohibited	CP	Set sigmoid motion ratio
SGET	209	Optional	Variable number	Column number, character literal	CP	Get character
SIN	101	Optional	Sine assignment variable	Operand	ZR	Sine
SLCT	189	Optional	Prohibited	Prohibited	CP	Declare start of multi-branching
SLEN	215	Optional	Character string length	Prohibited	CP	Set length
SPUT	210	Optional	Column number	Data	CP	Set character
SQR	103	Optional	Root assignment variable	Operand	ZR	Root
SSPG	128	Optional	Pause program number	(Pause program number)	CC	Pause program
STOP	171	Optional	Axis stop pattern	Prohibited	CP	Decelerate and stop axis
STR	211	Optional	Column number	Data	CC	Convert character string; decimal
STRH	212	Optional	Column number	Data	CC	Convert character string; hexadecimal
SUB	98	Optional	Minuend variable	Subtrahend	ZR	Subtract
SV□□	161	Optional	Operation axis pattern	Prohibited	PE	Servo [ON, OF]
SYST	195	Optional	Variable number	Prohibited	CP	Get system status

Operation type in the output field
CC: Command was executed successfully, ZR: Operation result is zero,
PE: Operation is complete, CP: Command part has passed, TU: Time up
EQ: Operand 1 = Operand 2, NE: Operand 1 ≠ Operand 2,
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2,
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Command	Page	Condition	Operand 1	Operand 2	Output	Function
T						
TAG	122	Prohibited	Declaration tag number	Prohibited	CP	Declare jump destination
TAN	102	Optional	Tangent assignment variable	Operand	ZR	Tangent
TIMC	109	Optional	Program number	Prohibited	CP	Cancel waiting
TIMR	251	Optional	Local flag number	Timer setting	CP	Timer relay for ladder
TIMW	108	Optional	Wait time	Prohibited	TU	Wait
TMRW	203	Optional	Read timer setting	(Write timer setting)	CP	Set READ timeout value
TPCD	233	Prohibited	0 or 1	Prohibited	CP	Specify processing to be performed when input condition is not specified
TRAN	96	Optional	Copy-destination variable	Copy-source variable	ZR	Copy
TSLP	234	Prohibited	Time	Prohibited	CP	Task sleep
V						
VAL	213	Optional	Variable number	Column number, character literal	CC	Convert character string data; decimal
VALH	214	Optional	Variable number	Column number, character literal	CC	Convert character string data; hexadecimal
VEL	145	Optional	Speed	Prohibited	CP	Set speed
VLMX	156	Optional	Prohibited	Prohibited	CP	Specify VLMX speed
W						
WH□□	190	Prohibited	Comparison variable	Comparison value	CP	Branch value [EQ, NE, GT, GE, LT, LE]
WRIT	205	Optional	Channel number	Column number	CC	Output to channel
WS□□	191	Prohibited	Column number	Column number, character literal	CP	Branch character string [EQ, NE]
WT□□	114	Optional	I/O, flag	(Wait time)	TU	Wait for I/O, flag [ON, OF]
WZFA	198	Optional	Zone number	Axis pattern	CP	Wait for zone OFF, with AND
WZFO	199	Optional	Zone number	Axis pattern	CP	Wait for zone OFF, with OR
WZNA	196	Optional	Zone number	Axis pattern	CP	Wait for zone ON, with AND
WZNO	197	Optional	Zone number	Axis pattern	CP	Wait for zone ON, with OR

Chapter 3 Explanation of Commands

1. Commands

1.1 Variable Assignment

- LET (Assign)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	LET	Variable number	Data	ZR

[Function] Assign the value specified in operand 2 to the variable specified in operand 1. The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]

LET	1	10	Assign 10 to variable 1.

[Example 2]

LET	3	10	Assign 10 to variable 3.
LET	1	*3	Assign the content of variable 3 (10) to variable 1.

(Note) When data in a real variable is assigned to an integer variable, all decimal fractions are rounded to the nearest integer.

LET	100	13.5	Assign 13.5 to real variable 100.
LET	1	*100	Assign 14, which is a rounded result of the content of real variable 100 (13.5), to integer variable 1.

- TRAN (Copy)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	TRAN	Variable number	Variable number	ZR

[Function] Assign the content of the variable specified in operand 2 to the variable specified in operand 1.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]	TRAN	1	2	Assign the content of variable 2 to variable 1.
	LET	1	*2	A LET command of the same effect as the above operation
[Example 2]	LET	3	4	Assign 4 to variable 3.
	LET	4	10	Assign 10 to variable 4.
	TRAN	1	*3	Assign the content of variable 3 (which is variable 4, or 10) to variable 1.

(Note) When data in a real variable is assigned to an integer variable, all decimal fractions are rounded to the nearest integer.

LET	100	13.5	Assign 13.5 to real variable 100.
TRAN	1	100	Assign 14, which is a rounded result of the content of real variable 100 (13.5), to integer variable 1.

- CLR (Clear variable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CLR	Variable number	Variable number	ZR

[Function] Clear the variables from the one specified in operand 1 through the other specified in operand 2.
The contents of the variables that have been cleared become 0.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]	CLR	1	5	Clear variables 1 through 5.
[Example 2]	LET	1	10	Assign 10 to variable 1.
	LET	2	20	Assign 20 to variable 2.
	CLR	*1	*2	Clear the variables from the content of variable 1 (variable 10) through the content of variable 2 (variable 20).

1.2 Arithmetic Operation

- ADD (Add)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ADD	Variable number	Data	ZR

[Function] Add the content of the variable specified in operand 1 and the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	3	Assign 3 to variable 1.
	ADD	1	2	Add 2 to the content of variable 1 (3). 5 (3+2=5) will be stored in variable 1.
[Example 2]	LET	1	2	Assign 2 to variable 1.
	LET	3	2	Assign 2 to variable 3.
	ADD	1	*3	Add the content of variable 3 (2) to the content of variable 1 (2). 4 (2+2=4) will be stored in variable 1.

- SUB (Subtract)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SUB	Variable number	Data	ZR

[Function] Subtract the value specified in operand 2 from the content of the variable specified in operand 1, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	3	Assign 3 to variable 1.
	SUB	1	2	Subtract 2 from the content of variable 1 (3). 1 (3–2=1) will be stored in variable 1.
[Example 2]	LET	1	3	Assign 3 to variable 1.
	LET	3	2	Assign 2 to variable 3.
	SUB	1	*3	Subtract the content of variable 3 (2) from the content of variable 1 (3). 1 (3–2=1) will be stored in variable 1.

- MULT (Multiply)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MULT	Variable number	Data	ZR

[Function] Multiply the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	3	Assign 3 to variable 1.
	MULT	1	2	Multiply the content of variable 1 (3) by 2. 6 (3x2=6) will be stored in variable 1.

[Example 2]	LET	1	4	Assign 4 to variable 1.
	LET	3	2	Assign 2 to variable 3.
	MULT	1	*3	Multiply the content of variable 1 (4) by the content of variable 3 (2). 8 (4x2=8) will be stored in variable 1.

- DIV (Divide)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	DIV	Variable number	Data	ZR

[Function] Divide the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

(Note) If the variable specified in operand 1 is an integer variable, any decimal places will be rounded off.

[Example 1]	LET	1	6	Assign 6 to variable 1.
	DIV	1	2	Divide the content of variable 1 (6) by 2. 3 (6÷2=3) will be stored in variable 1.

[Example 2]	LET	1	6	Assign 6 to variable 1.
	LET	3	2	Assign 2 to variable 3.
	DIV	1	*3	Divide the content of variable 1 (6) by the content of variable 3 (2). 3 (6÷2=3) will be stored in variable 1.

- MOD (Remainder)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MOD	Variable number	Data	ZR

[Function] Assign, to the variable specified in 1, the remainder obtained by dividing the content of the variable specified in operand 1 by the value specified in operand 2.
The output will turn ON when the operation result becomes 0.

(Note) A MOD command is used with integer variables.

[Example 1]

LET	1	7	Assign 7 to variable 1.
MOD	1	3	Obtain the remainder of dividing the content of variable 1 (7) by 3. 1 (7÷3=2 with a remainder of 1) will be assigned to variable 1.

[Example 2]

LET	1	7	Assign 2 to variable 1.
LET	3	3	Assign 3 to variable 3.
MOD	1	*3	Obtain the remainder of dividing the content of variable 1 (7) by the content of variable 3 (3). 1 (7÷3=2 with a remainder of 1) will be assigned to variable 1.

1.3 Function Operation

- SIN (Sine operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SIN	Variable number	Data	ZR

[Function] Assign the sine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle x π ÷ 180

[Example 1]

```
SIN    100    0.523599    Assign the sine of 0.523599 (0.5) to variable 100.
```

[Example 2]

```
LET    101    30          30 x π ÷ 180 (radian)
MULT   101    3.141592    (30° will be converted to radian and assigned to
DIV    101    180         variable 101.)
SIN    100    *101        Assign the sine of the content of variable 101 (0.5) to
                          variable 100.
```

- COS (Cosine operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	COS	Variable number	Data	ZR

[Function] Assign the cosine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle x π ÷ 180

[Example 1]

```
COS    100    1.047197    Assign the cosine of 1.047197 (0.5) to variable 100.
```

[Example 2]

```
LET    101    60          60 x π ÷ 180 (radian)
MULT   101    3.141592    (60° will be converted to radian and assigned to
DIV    101    180         variable 101.)
COS    100    *101        Assign the cosine of the content of variable 101 (0.5)
                          to variable 100.
```

- TAN (Tangent operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	TAN	Variable number	Data	ZR

[Function] Assign the tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle x π ÷ 180

[Example 1] TAN 100 0.785398 Assign the tangent of 0.785398 (1) to variable 100.

[Example 2]

LET	101	45	45 x π ÷ 180 (radian)
MULT	101	3.141592	(45° will be converted to radian and assigned to
DIV	101	180	variable 101.)
TAN	100	*101	Assign the tangent of the content of variable 101 (1) to variable 100.

- ATN (Inverse-tangent operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ATN	Variable number	Data	ZR

[Function] Assign the inverse tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of inverse tangent is radian.

(Note 1) Radian = Angle x π ÷ 180

[Example 1] ATN 100 1 Assign the inverse tangent of 1 (0.785398) to variable 100.

[Example 2]

LET	101	1	Assign 1 to variable 101.
ATN	100	*101	Assign the inverse tangent of the content of variable 101 (0.785398) to variable 100.

- SQR (Root operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SQR	Variable number	Data	ZR

[Function] Assign the root of the data specified in operand 2 to the variable specified in operand 1. The output will turn ON when the operation result becomes 0.

[Example 1]	SQR	1	4	Assign the root of 4 (2) to variable 1.
[Example 2]	LET	2	5	Assign 5 to variable 2.
	SQR	100	*2	Assign the root of the content of variable 2 (5) to variable 100.

1.4 Logical Operation

- AND (Logical AND)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	AND	Variable number	Data	ZR

[Function] Assign the logical AND operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	204	Assign 204 to variable 1.
	AND	1	170	Assign the logical AND operation result (136) of the content of variable 1 (204) and 170, to variable 1.
[Example 2]	LET	1	204	Assign 204 to variable 1.
	LET	3	170	Assign 170 to variable 3.
	AND	1	*3	Assign the logical AND operation result (136) of the content of variable 1 (204) and the content of variable 3 (170) to variable 1.

	Decimal		Binary
	204		11001100
AND	170	AND	10101010
	136		10001000

- OR (Logical OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OR	Variable number	Data	ZR

[Function] Assign the logical OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	204	Assign 204 to variable 1.
	OR	1	170	Assign the logical OR operation result (238) of the content of variable 1 (204) and 170, to variable 1.
[Example 2]	LET	1	204	Assign 204 to variable 1.
	LET	3	170	Assign 170 to variable 3.
	OR	1	*3	Assign the logical OR operation result (238) of the content of variable 1 (204) and the content of variable 3 (170) to variable 1.

Decimal	Binary
204	11001100
OR 170	OR 10101010
238	11101110

- EOR (Logical exclusive-OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	EOR	Variable number	Data	ZR

[Function] Assign the logical exclusive-OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1. The output will turn ON when the operation result becomes 0.

[Example 1]	LET	1	204	Assign 204 to variable 1.
	EOR	1	170	Assign the logical exclusive-OR operation result (102) of the content of variable 1 (204) and 170, to variable 1.
[Example 2]	LET	1	204	Assign 204 to variable 1.
	LET	3	170	Assign 170 to variable 3.
	EOR	1	*3	Assign the logical exclusive-OR operation result (102) of the content of variable 1 (204) and the content of variable 3 (170) to variable 1.

Decimal	Binary
204	11001100
EOR 170	EOR 10101010
102	01100110

1.5 Comparison Operation

- CP□□ (Compare)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CP□□	Variable number	Data	EQ NE GT GE LT LE

[Function] The output will be turned ON if the comparison result of the content of the variable specified in operand 1 and the value specified in operand 2 satisfies the condition.
The value in the variable does not change.
The output will be turned OFF if the condition is not satisfied.

(Note) The output will not be turned OFF when the command is executed.

CP□□

EQ	....	Operand 1 = Operand 2
NE	....	Operand 1 ≠ Operand 2
GT	....	Operand 1 > Operand 2
GE	....	Operand 1 ≥ Operand 2
LT	....	Operand 1 < Operand 2
LE	....	Operand 1 ≤ Operand 2

[Example 1]

	LET	1	10		Assign 10 to variable 1.
	CPEQ	1	10	600	Turn ON flag 600 if the content of variable 1 is 10.
600	ADD	2	1		Add 1 to variable 2 if flag 600 is ON.

[Example 2]

	LET	1	10		Assign 10 to variable 1.
	LET	3	10		Assign 10 to variable 3.
	CPEQ	1	*3	310	Turn ON output 310 if the content of variable 1 (10) is equal to the content of variable 3.

1.6 Timer

- TIMW (Timer)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	TIMW	Time	Prohibited	TU

[Function] Stop the program and wait for the time specified in operand 1.
The setting range is 0.01 to 99, and the unit is second.
The output will turn ON when the specified time has elapsed and the program proceeds to the next step.

[Example 1]	TIMW	1.5		Wait for 1.5 seconds.
[Example 2]	LET	1	10	Assign 10 to variable 1.
	TIMW	*1		Wait for the content of variable 1 (10 seconds).

- TIMC (Cancel timer)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	TIMC	Program number	Prohibited	CP

[Function] Cancel a timer in other program running in parallel.

(Note) Timers in TIMW, WTON, WTOF and READ commands can be cancelled. In the case of WTON, WTOF and READ commands, even if timeout is not specified it is assumed that an unlimited timer has been specified and the wait time will be cancelled.

[Example 1]	TIMC	10		Cancel the wait time in program 10.
[Example 2]	LET	1	10	Assign 10 to variable 1.
	TIMC	*1		Cancel the wait time in the content of variable 1 (program 10).

[Example 3]	Program 1		Program 10	
	:		:	
	:		WTON 8 20	Program 10 waits for input 8 for 20 seconds.
	:		(Wait for input 8)	
	TIMC	10	(Wait for input 8)	Cancel the wait time in program 10.
	:		:	

(Note) The steps shown in the above example represent those executed simultaneously in different programs.

- GTTM (Get time)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GTTM	Variable number	Prohibited	CP

[Function] Read system time to the variable specified in operand 1. The time is specified in units of 10 milliseconds.
The time obtained here has no base number. Therefore, this command is called twice and the difference will be used to calculate the elapsed time.

[Example 1]

```
GTTM   1          Read the reference time to variable 1.
ADD    1    500   Set the ending time to 5 seconds later.
GTTM   2          Read the current system time to variable 2.
DWLE   2    *1    Proceed to the step next to EDDO when 5 seconds elapsed.
  :               The above process will be repeated for 5 seconds.
  :
GTTM   2          Read the current system time to variable 2.
EDDO
```

[Example 2]

```
LET    1    5     Assign 5 to variable 1.
GTTM   *1         Store the current system time in the content of variable 1
                  (variable 5).
```

1.7 I/O, Flag Operation

- BT□□ (Output port, flag operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	BT□□	Output, flag	(Output, flag)	CP

[Function] Reverse the ON/OFF status of the output ports or flags from the one specified in operand 1 through the other specified in operand 2.

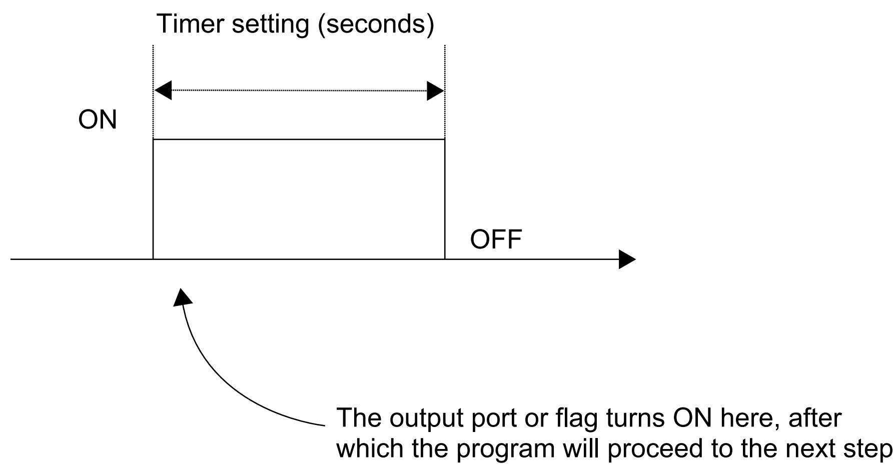
```
BT□□
   ┌ON ............ Switch the status to ON.
  ─┤OF ............ Switch the status to OFF.
   └NT ............ Reverse the status.
```

[Example 1]	BTON	300		Turn ON output port 300.
[Example 2]	BTOF	300	307	Turn OFF output ports 300 through 307.
[Example 3]	LET	1	600	Assign 600 to variable 1.
	BTNT	*1		Reverse the content of variable 1 (flag 600).
[Example 4]	LET	1	600	Assign 600 to variable 1.
	LET	2	607	Assign 607 to variable 2.
	BTON	*1	*2	Turn ON the flags from the content of variable 1 (flag 600) through the content of variable 2 (flag 607).

● BTPN (Output ON pulse)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	BTPN	Output port, flag	Timer setting	CP

[Function] Turn ON the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned ON and then the program will proceed to the next step. The output port or flag will be turned OFF automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00 seconds (including up to two decimal places).



(Note 1) If this command is executed with respect to an output port or flag already ON, the output port or flag will be turned OFF upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned OFF.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program. (There is no limitation as to how many times these timers can be used in a single program.)

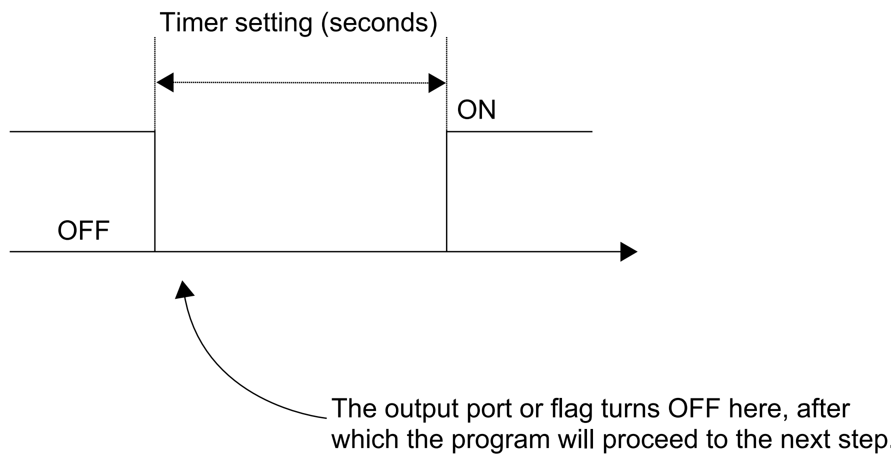
[Example]

BTPN	300	1	Turn ON output port 300 for 1 second.
BTPN	600	10	Turn ON flag 600 for 10 seconds.

- BTPF (Output OFF pulse)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	BTPF	Output port, flag	Timer setting	CP

[Function] Turn OFF the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned OFF and then the program will proceed to the next step. The output port or flag will be turned ON automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00 seconds (including up to two decimal places).



(Note 1) If this command is executed with respect to an output port or flag already OFF, the output port or flag will be turned ON upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned ON.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program. (There is no limitation as to how many times these timers can be used in a single program.)

[Example]

BTPF	300	1	Turn OFF output port 300 for 1 second.
BTPF	600	10	Turn OFF flag 600 for 10 seconds.

(Note 5) If a different task or interruption processing occurs after the port has turned ON, and before it turns OFF again, an error will occur in the pulse output time. In this case, BTPF can no longer be used as a constant-time pulse output command.

- WT□□ (Wait for I/O port, flag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WT□□	I/O, flag	(Time)	TU

[Function] Wait for the I/O port or flag specified in operand 1 to turn ON/OFF.
The program can be aborted after the specified time by setting the time in operand 2.
The setting range is 0.01 to 99 seconds.
The output will turn ON upon elapse of the specified time (only when operand 2 is specified).

Note) A local flag cannot be entered in operand 1.

WT□□
- ON .... Wait for the applicable I/O port or flag to turn ON.
- OF .... Wait for the applicable I/O port or flag to turn OFF.

[Example 1]	WTON	15		Wait for input port 15 to turn ON.
[Example 2]	WTOF	307	10	Wait for 10 seconds for output port 307 to turn OFF.
[Example 3]	LET	1	600	Assign 600 to variable 1.
	WTON	*1		Wait for the content of variable 1 (flag 600) to turn ON.
[Example 4]	LET	1	8	Assign 8 to variable 1.
	LET	2	5	Assign 5 to variable 2.
	WTOF	*1	*2	Wait for the content of variable 2 (5 seconds) for the content of variable 1 (input port 8) to turn OFF.

- IN (Read I/O, flag as binary)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	IN	I/O, flag	I/O, flag	CC

[Function] Read the I/O ports or flags from the one specified in operand 1 through the other specified in operand 2, to variable 99 as a binary.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.... Binary
15	14	13	12	11	10	9	8	.... Input port number
ON	OFF	OFF	OFF	OFF	ON	OFF	ON	

1	0	0	0	0	1	0	1	.... Binary

$2^7 + 0 + 0 + 0 + 0 + 2^2 + 0 + 2^0$

$128 + 0 + 0 + 0 + 0 + 4 + 0 + 1 = 133$

$\boxed{133}$ .... Variable 99

(Note 1) A maximum of 32 bits can be input.

(Note 2) When 32 bits have been input and the most significant bit is ON, the value read to variable 99 will be treated as a negative value.

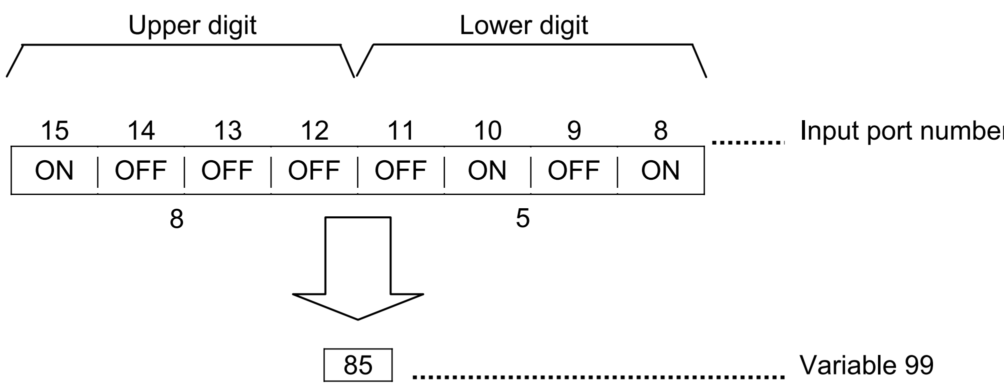
(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]	IN	8	15	Read input ports 8 through 15, to variable 99 as a binary.
[Example 2]	LET	1	8	Assign 8 to variable 1.
	LET	2	15	Assign 15 to variable 2.
	IN	*1	*2	Read the input ports from the content of variable 1 (input port 8) through the content of variable 2 (input port 15), to variable 99 as a binary.

● INB (Read I/O, flag as BCD)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	INB	I/O, flag	BCD digits	CC

[Function] Read the I/O ports or flags from the one specified in operand 1 for the number of digits specified in operand 2, to variable 99 as a BCD.



(Note 1) A maximum of eight digits (32 bits) can be input.

(Note 2) The number of I/O ports and flags that can be used is 4 x n (digits).

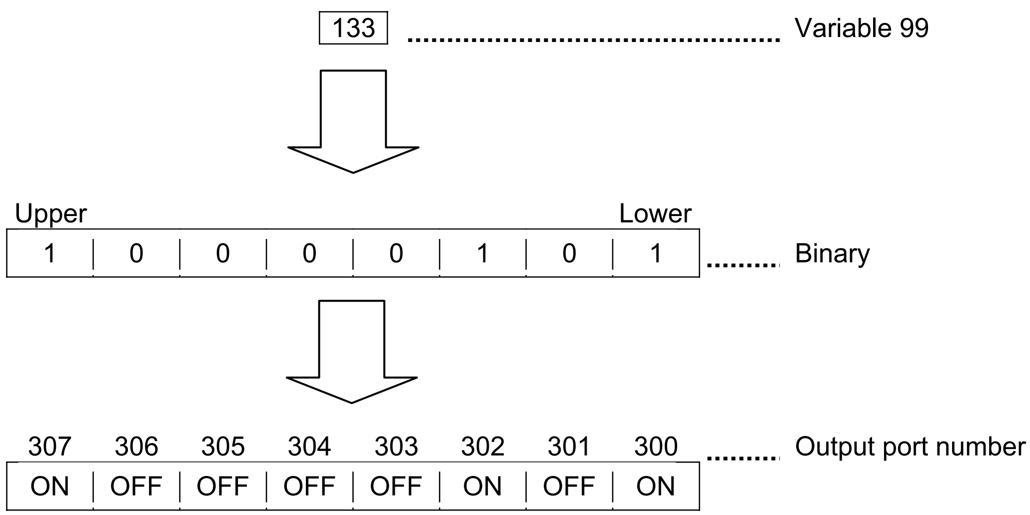
(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]	INB	8	2	Read input ports 8 through 15, to variable 99 as a BCD.
[Example 2]	LET	1	8	Assign 8 to variable 1.
	LET	2	2	Assign 2 to variable 2.
	INB	*1	*2	Read the input ports from the content of variable 1 (input port 8) for the content of variable 2 (two digits) (until input port 15), to variable 99 as a BCD.

- OUT (Write output, flag as binary)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OUT	Output, flag	Output, flag	CC

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 through the other specified in operand 2.



(Note 1) A maximum of 32 bits can be output.

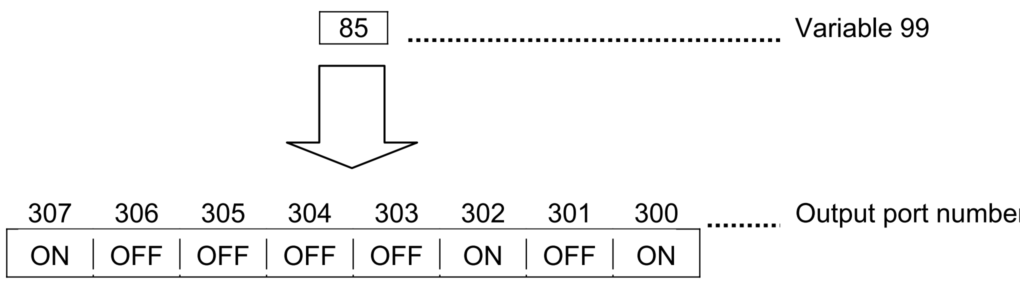
(Note 2) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]	OUT	300	307	Write the value in variable 99 to output ports 300 through 307 as a binary.
[Example 2]	LET	1	300	Assign 300 to variable 1.
	LET	2	307	Assign 307 to variable 2.
	OUT	*1	*2	Write the value in variable 99 to the output ports from the content of variable 1 (output port 300) through the content of variable 2 (output port 307) as a binary.

- OUTB (Write output, flag as BCD)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OUTB	Output, flag	BCD digits	CC

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 for the number of digits specified in operand 2 as a BCD.



(Note 1) A maximum of eight digits (32 bits) can be output.

(Note 2) The number of output ports and flags that can be used is 4 x n (digits).

(Note 3) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]	OUTB	300	2	Write the value in variable 99 to the output ports from 300 for two digits (until output port 307) as a BCD.
[Example 2]	LET	1	300	Assign 300 to variable 1.
	LET	2	2	Assign 2 to variable 2.
	OUTB	*1	*2	Write the value in variable 99 to the output ports from the content of variable 1 (output port 300) for the content of variable 2 (two digits) (until output port 307) as a BCD.

- FMIO (Set IN, INB, OUT, OUTB command format)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	FMIO	Format type	Prohibited	CP

[Function] Set the data format for reading or writing I/O ports and flags with an IN, INB, OUT or OUTB command.

(1) Operand 1 = 0 (Default status when a FMIO command has not been executed)
Data is read or written without being reversed.

Variable 99		Temporary data					I/O port, flag status (0 = OFF, 1 = ON)			
							(I/O, flag number upper)		(I/O, flag number lower)	
01234567h	⇔	01h	23h	45h	67h	⇔	0000 0001	0010 0011	0100 0101	0110 0111

OUT(B) command →

← IN(B) command

(2) Operand 1 = 1
Data is read or written after its upper eight bits and lower eight bits are reversed every 16 bits.

Variable 99		Temporary data					I/O port, flag status (0 = OFF, 1 = ON)			
							(I/O, flag number upper)		(I/O, flag number lower)	
01234567h	⇔	23h	01h	67h	45h	⇔	0010 0011	0000 0001	0110 0111	0100 0101

OUT(B) command →

← IN(B) command

(3) Operand 1 = 2
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits.

Variable 99		Temporary data					I/O port, flag status (0 = OFF, 1 = ON)			
							(I/O, flag number upper)		(I/O, flag number lower)	
01234567h	⇔	45h	67h	01h	23h	⇔	0100 0101	0110 0111	0000 0001	0010 0011

OUT(B) command →

← IN(B) command

(4) Operand 1 = 3
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits and its upper eight bits and lower eight bits are reversed every 16 bits.

								(I/O, flag number upper)		(I/O, flag number lower)	
01234567h	⇔	67h	45h	23h	01h	⇔	0110 0111	0100 0101	0010 0011	0000 0001
Variable 99							I/O port, flag status (0 = OFF, 1 = ON)			

Temporary data

OUT(B) command →

← IN(B) command

[Example 1] Variable 99 = 00123456h (Decimal: 1193046, BCD: 123456)

00123456h ← OUT(B) command / IN(B) command → Variable 99 1193046 (IN/OUT command)
123456 (INB/OUTB command)

OUT(B) command ↓ ↑ IN(B) command

						(I/O, flag number upper)		(I/O, flag number lower)	
FMIO = 0	00h	12h	34h	56h	⇔	0000 0000	0001 0010	0011 0100	0101 0110
FMIO = 1	12h	00h	56h	34h	⇔	0001 0010	0000 0000	0101 0110	0011 0100
FMIO = 2	34h	56h	00h	12h	⇔	0011 0100	0101 0110	0000 0000	0001 0010
FMIO = 3	56h	34h	12h	00h	⇔	0101 0110	0011 0100	0001 0010	0000 0000

Temporary data OUT(B) command → I/O port, flag status (0 = OFF, 1 = ON)

← IN(B) command

[Example 2] Variable 99 = 00001234h (Decimal: 4660, BCD: 1234)

00001234h ⇄ (OUT(B) command → / ← IN(B) command) Variable 99 4660 (IN/OUT command)
1234 (INB/OUTB command)

OUT(B) command ↓ ↑ IN(B) command

	Temporary data				⇔	(I/O, flag number upper)		(I/O, flag number lower)	
FMIO = 0	00h	00h	12h	34h	⇔	0000 0000	0000 0000	0001 0010	0011 0100
FMIO = 1	00h	00h	34h	12h	⇔	0000 0000	0000 0000	0011 0100	0001 0010
FMIO = 2	12h	34h	00h	00h	⇔	0001 0010	0011 0100	0000 0000	0000 0000
FMIO = 3	34h	12h	00h	00h	⇔	0011 0100	0001 0010	0000 0000	0000 0000

Temporary data — OUT(B) command → / ← IN(B) command — I/O port, flag status (0 = OFF, 1 = ON)

[Example 3] Variable 99 = 00000012h (Decimal: 18, BCD: 12)

00000012h ⇄ (OUT(B) command → / ← IN(B) command) Variable 99 18 (IN/OUT command)
12 (INB/OUTB command)

OUT(B) command ↓ ↑ IN(B) command

	Temporary data				⇔	(I/O, flag number upper)		(I/O, flag number lower)	
FMIO = 0	00h	00h	00h	12h	⇔	0000 0000	0000 0000	0000 0000	0001 0010
FMIO = 1	00h	00h	12h	00h	⇔	0000 0000	0000 0000	0001 0010	0000 0000
FMIO = 2	00h	12h	00h	00h	⇔	0000 0000	0001 0010	0000 0000	0000 0000
FMIO = 3	12h	00h	00h	00h	⇔	0001 0010	0000 0000	0000 0000	0000 0000

Temporary data — OUT(B) command → / ← IN(B) command — I/O port, flag status (0 = OFF, 1 = ON)

1.8 Program Control

- GOTO (Jump)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GOTO	Tag number	Prohibited	CP

[Function] Jump to the position of the tag number specified in operand 1.

(Note) A GOTO command is valid only within the same program.

[Example 1]

```
TAG     1       Set a tag.
 :
 :
 :
GOTO    1       Jump to tag 1.
```

Using a GOTO command to branch out of or into any of the syntaxes listed below is prohibited.

Since the maximum number of nests is defined for each conditional branching command or subroutine call, a nest will be infinitely repeated if an ED□□ is not passed, and a nest overflow error will generate. In the case of palletizing setting, an error will generate if the second BGPA is declared after the first BGPA declaration without passing an EDPA.

(1) IF□□ or IS□□ and EDIF syntax
(2) DW□□ and EDDO syntax
(3) SLCT and EDSL syntax
(4) BGSR and EDSR syntax
(5) BGPA and EDPA syntax

- TAG (Declare tag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	TAG	Tag number	Prohibited	CP

[Function] Set the tag number specified in operand 1.

[Example 1] Refer to the section on GOTO command.

- EXSR (Execute subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	EXSR	Subroutine number	Prohibited	CP

[Function] Execute the subroutine specified in operand 1.
A maximum of 15 nested subroutine calls are supported.

(Note) This command is valid only for subroutines within the same program.

[Example 1]

```
EXSR    1           Execute subroutine 1.
  :
  :
EXIT
BGSR    1           Start subroutine 1.
  :
  :
  :
EDSR                End subroutine 1.
```

[Example 2]

```
LET     1    10     Assign 10 to variable 1.
EXSR    *1          Execute the content of variable 1 (subroutine 10).
```

- BGSR (Start subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	BGSR	Subroutine number	Prohibited	CP

[Function] Declare the start of the subroutine specified in operand 1.

[Example 1] Refer to the section on EXSR command.

(Note) Using a GOTO command to branch out of or into a BGSR-EDSR syntax is prohibited.

- EDSR (End subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	EDSR	Prohibited	Prohibited	CP

[Function] Declare the end of a subroutine.
This command is always required at the end of a subroutine.
Thereafter, the program will proceed to the step next to the EXSR that has been called.

[Example 1] Refer to the section on EXSR command.

1.9 Task Management

- EXIT (End program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	EXIT	Prohibited	Prohibited	CP

[Function] End the program.
If the last step has been reached without encountering any EXIT command, the program will return to the beginning.

(Note) Status at program end

- Output ports .... Retained
- Local flags .... Cleared
- Local variables .... Cleared
- Current values .... Retained
- Global flags .... Retained
- Global variables .... Retained

[Example 1]

```
  :
  :
EXIT          End the program.
```

- EXPG (Start other program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	EXPG	Program number	(Program number)	CC

[Function] Start the programs from the one specified in operand 1 through the other specified in operand 2, and run them in parallel. Specification in operand 1 only is allowed.

[Example 1] EXPG 10 12 Start program Nos. 10, 11 and 12.

Error-generation/output-operation conditions
When one EXPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	A57 “Multiple program start error”	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 1 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple EXPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range	None of programs inside the specified range are running		
Error	A57 “Multiple program start error”	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 2 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.
* 3 --- In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

- ABPG (Abort other program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ABPG	Program number	(Program number)	CC

[Function] Forcibly end the programs from the one specified in operand 1 to the other specified in operand 2. Specification in operand 1 only is allowed.

(Note 1) If an ABPG command is issued while a movement command is being executed, the axes will immediately decelerate and stop.
(Note 2) Not only the operation but also the execution of the step itself will be terminated.

[Example 1] ABPG 10 12 End program Nos. 10, 11 and 12.

Error-generation/output-operation conditions
When one ABPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	None	C2C “Program number error”
Output operation	ON (OFF *2)	ON	ON	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 1 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.
* 2 --- If an own task (own program) is specified in an ABPG command, the own task will be terminated and then deleted. The output will turn OFF.

When multiple ABPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *3			Program number error *1
	Registered program exists inside the specified range *4		None of programs inside the specified range are registered	
	Running program exists inside the specified range	None of programs inside the specified range are running		
Error	None	None	None	C2C “Program number error”
Output operation	ON (OFF *5)	ON	ON	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 3 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.
* 4 --- In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.
* 5 --- If an own task (own program) is included in the specified range, the own task will be terminated, upon which the processing of the ABPG command will end. Since the own task will be deleted, the result of ending the processing of specified programs will become indeterminable. Exercise caution. The output will always turn OFF regardless of the result.

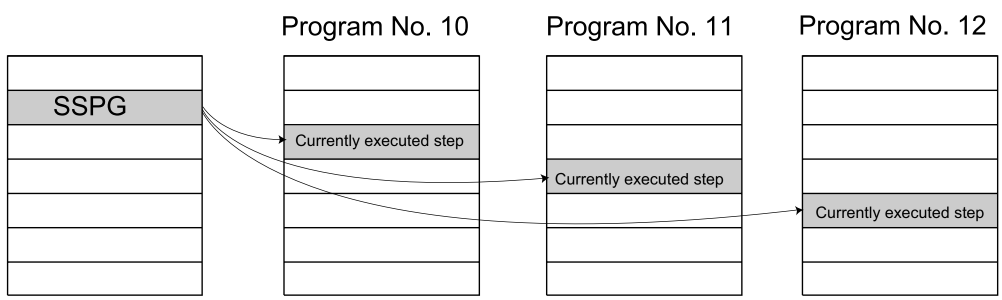
- SSPG (Pause program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SSPG	Program number	(Program number)	CC

[Function] Pause the program from the one specified in operand 1 through the other specified in operand 2, at the current step. Specification in operand 1 only is allowed.

(Note 1) Pausing a program will also pause the operation the program has been executing.
(Note 2) Not only the operation but also the execution of the step itself will be paused.

[Example 1] SSPG 10 12 Pause program Nos. 10, 11 and 12 at the current step.



Error-generation/output-operation conditions
When one SSPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	C03 "Non-registered program specification error"	C2C "Program number error"
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 1 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple SSPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range *4	None of programs inside the specified range are running		
Error	None	None	C03 "Non-registered program specification error"	C2C "Program number error"
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 2 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.
* 3 --- In this case, non-registered programs inside the specified range are not treated as a target of operation with EXPG, ABPG, SSPG and PSPG commands. This will not affect error generation or output operation.
* 4 --- In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

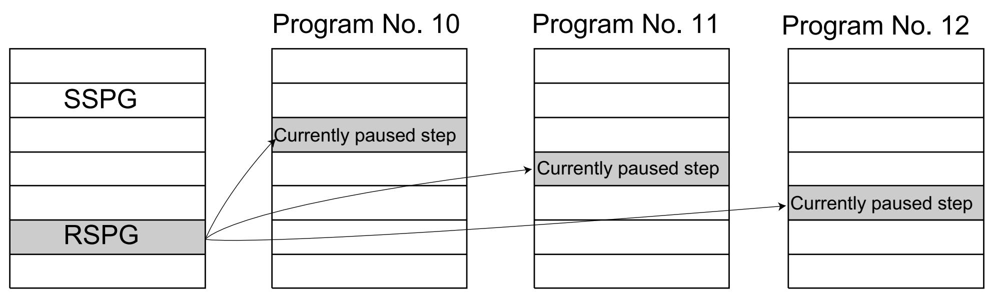
- RSPG (Resume program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	RSPG	Program number	(Program number)	CC

[Function] Resume the programs from the one specified in operand 1 through the other specified in operand 2. Specification in operand 1 only is allowed.

(Note 1) Resuming a program will also resume the operation the program had been executing before the pause.

[Example 1] RSPG 10 12 Resume program Nos. 10, 11 and 12 from the paused step.



Error-generation/output-operation conditions
When one RSPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 1 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple RSPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range *4	None of programs inside the specified range are running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.
* 2 --- Program number error indicates specification of a number smaller than 1 or exceeding 64.
* 3 --- In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.
* 4 --- In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

1.10 Position Operation

- PGET (Read position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PGET	Axis number	Position number	CC

[Function] Read to variable 199 the data of the axis number specified in operand 1 in the position data specified in operand 2.
If a PGET command is executed when the position data table contains no data to be acquired (the position data field on the teaching pendant shows "X.XXX" or the position data field in the PC software is blank), no data will be assigned to variable 199 (the PGET command will not be executed).

[Example 1]	PGET	2	3	Read to variable 199 the data of axis 2 at position 3.
[Example 2]	LET	1	2	Assign 2 to variable 1.
	LET	2	3	Assign 3 to variable 2.
	PGET	*1	*2	Read to variable 199 the data of the content of variable 1 (axis 2) at the content of variable 2 (position 3).

- PPUT (Write position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PPUT	Axis number	Position number	CP

[Function] Write the value in variable 199 to the axis number specified in operand 1 in the position data specified in operand 2.

[Example 1]	LET	199	150	Assign 150 to variable 199.
	PPUT	2	3	Write the content of variable 199 (150) to axis 2 at position 3.

[Example 2]	LET	199	150	Assign 150 to variable 199.
	LET	1	2	Assign 2 to variable 1.
	LET	2	3	Assign 3 to variable 2
	PPUT	*1	*2	Write the content of variable 199 (150) to the content of variable 1 (axis 2) at the content of variable 2 (position 3).

- PCLR (Clear position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PCLR	Position number	Position number	CP

[Function] Clear the position data from the one specified in operand 1 through the other specified in operand 2.
Once the data has been deleted, the position data field no longer contains any data; it does not store a value of "0.000." The position data field on the teaching pendant shows "X.XXX," while the position data field in the PC software becomes blank.

[Example 1]	PCLR	10	20	Clear the data from position Nos. 10 through 20.
[Example 2]	LET	1	10	Assign 10 to variable 1.
	LET	2	20	Assign 20 to variable 2.
	PCLR	*1	*2	Clear the data of the content of variable 1 (position 10) through the content of variable 2 (position 20).

- PCPY (Copy position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PCPY	Position number	Position number	CP

[Function] Copy the position data specified in operand 2 to the position number specified in operand 1.

[Example 1]	PCPY	20	10	Copy the data of position No. 10 to position No. 20.
[Example 2]	LET	1	20	Assign 20 to variable 1.
	LET	2	10	Assign 10 to variable 2.
	PCPY	*1	*2	Copy the data of the content of variable 2 (position 10) to the content of variable 1 (position 20).

- PRED (Read current position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PRED	Axis pattern	Position number	CP

[Function] Read the current position of the axis specified in operand 1 to the position specified in operand 2.

[Example 1] PRED 11 10 Read the current positions of axes 1 and 2 to position No. 10.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 1 3 Assign 3 to variable 1.
PRED *1 10

[Example 3] LET 1 10 Assign 10 to variable 1.
PRED 11 *1 Read the current positions of axes 1 and 2 to the content of variable 1 (position 10).

- PRDQ (Read current axis position (1 axis direct))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PRDQ	Axis number	Variable number	CP

[Function] Read the current position of the axis number specified in operand 1 to the variable specified in operand 2.
The current position can be obtained more quickly than when a PRED command is used.
The current position of a synchronized slave axis can also be read.

[Example] PRDQ 2 100 Read the current position of axis 2 to variable 100.

- PTST (Check position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PTST	Axis pattern	Position number	CC

[Function] Check if valid data is contained in the axis pattern specified in operand 1 at the position number specified in operand 2.
If the data specified by the axis pattern is not available (the position data field on the teaching pendant shows "X.XXX" or the position data field in the PC software is blank), the output will turn ON. "0" is treated as a valid data value.

[Example 1] PTST 11 10 300 Turn ON output 300 if there are no valid values of axes 1 and 2 at position 10.
Output 300 will turn OFF if the position data is given as follows:

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET   1    3         Assign 3 to variable 1.
PTST  *1   10   300
```

[Example 3]

```
LET   1    11        Assign 11 to variable 1.
PTST  11   *1   600  Turn ON flag 600 if there are no valid values in the data
                     of axes 1 and 2 at the content of variable 1 (position
                     11).
                     Flag 600 will turn ON if the position data is given as
                     follows:
```

Position data display in PC software

No.	Axis 1	Axis 2	Vel	Acc	Dcl
10	100.000	50.000			
11					

- PVEL (Assign speed data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PVEL	Speed	Position number	CP

[Function] Write the speed specified in operand 1 to the position number specified in operand 2.

(Note) If a negative value is written with a PVEL command, an alarm will generate when that position is specified in a movement operation, etc. Exercise caution.

[Example 1]	PVEL	100	10	Write speed 100 mm/s to position No. 10.
[Example 2]	LET	1	100	Assign 100 to variable 1.
	LET	2	10	Assign 10 to variable 2.
	PVEL	*1	*2	Write the content of variable 1 (speed 100 mm/s) to the content of variable 2 (position 10).

- PACC (Assign acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PACC	Acceleration	Position number	CP

[Function] Write the acceleration specified in operand 1 to the position number specified in operand 2.

(Note) Range check is not performed for a PACC command. Be careful not to exceed the limit set for each actuator.

[Example 1]	PACC	0.3	10	Write acceleration 0.3 G to position No. 10.
[Example 2]	LET	100	0.3	Assign 0.3 to variable 100.
	LET	2	10	Assign 10 to variable 2.
	PACC	*100	*2	Write the content of variable 100 (acceleration 0.3 G) to the content of variable 2 (position 10).

- PDCL (Assign deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PDCL	Deceleration	Position number	CP

[Function] Assign the deceleration data specified in operand 1 to the deceleration item in the position data specified in operand 2.
The deceleration is set in G and may include up to two decimal places.

[Example 1] PDCL 0.3 3 Assign 0.3 to the deceleration data at position No. 3.

- PAXS (Read axis pattern)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAXS	Variable number	Position number	CP

[Function] Store the axis pattern at the position specified in operand 2 to the variable specified in operand 1.

[Example 1]	PAXS	1	99	Read the axis pattern at position 99 to variable 1. If the position is given as follows, "1" (binary 01) will be read to variable 1.
[Example 2]	LET	1	3	Assign 3 to variable 1.
	LET	2	101	Assign 101 to variable 2.
	PAXS	*1	*2	Read the axis pattern at the content of variable 2 (position 101) to the content of variable 1 (variable 3). If the point is given as follows, "3" (binary 11) will be stored in variable 3.

The table below shows different positions and corresponding values stored in a variable.

Position data display in PC software

No.	Axis 1	Axis 2		
98			....	0 0 = 0 + 0 = 0
99	100.000		....	0 1 = 0 + 1 = 1
100		150.000	....	1 0 = 2 + 0 = 2
101	100.000	50.000	....	1 1 = 2 + 1 = 3

- PSIZ (Check position data size)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PSIZ	Variable number	Prohibited	CP

[Function] Set an appropriate value in the variable specified in operand 1 in accordance with the parameter setting.

- When “Other parameter No. 23, PSIZ function type” = 0
  The maximum number of position data that can be stored in the controller will be set. (Regardless of whether the data are used or not.)
- When “Other parameter No. 23, PSIZ function type” = 1
  The number of point data used will be set.

[Example] PSIZ 1
When “Other parameter No. 23, PSIZ function type” = 0
The maximum number of position data that can be stored in variable 1 will be set.
When “Other parameter No. 23, PSIZ function type” = 1
The number of point data currently used will be set in variable 1.

- GVEL (Get speed data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GVEL	Variable number	Position number	CP

[Function] Obtain speed data from the speed item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GVEL 100 10 Set the speed data at position No. 10 in variable 100.

Position data display in PC software

No.	Axis 1	Axis 2	Vel	Acc	Dcl
1					
2					
• • •					
10	50.000	100.000	200	0.30	0.30
• •					

If the position data is set as above when the command is executed, 200 will be set in variable 100.

- GACC (Get acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GACC	Variable number	Position number	CP

[Function] Obtain acceleration data from the acceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GACC 100 10 Set the acceleration data at position No. 10 in variable 100.

Position data display in PC software

No.	Axis 1	Axis 2	Vel	Acc	Dcl
1					
2					
• • •					
10	50.000	100.000	200	0.30	0.30
• •					

If the position data is set as above when the command is executed, 0.3 will be set in variable 100.

- GDCL (Get deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GDCL	Variable number	Position number	CP

[Function] Obtain deceleration data from the deceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GDCL 100 10 Set the deceleration data at position No. 10 in variable 100.

Position data display in PC software

No.	Axis 1	Axis 2	Vel	Acc	Dcl
1					
2					
• • •					
10	50.000	100.000	200	0.30	0.30
• •					

If the position data is set as above when the command is executed, 0.3 will be set in variable 100.

1.11 Actuator Control Declaration

● VEL (Set speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	VEL	Speed	Prohibited	CP

[Function] Set the actuator travel speed in the value specified in operand 1.
The unit is mm/s.
The maximum speed will vary depending on the model of the actuator connected. Set a speed not exceeding the applicable maximum speed.

(Note 1) Decimal places cannot be used. An error will generate
(Note 2) The minimum speed is 1 mm/s.

[Example 1]	VEL	100		Set the speed to 100 mm/s.
	MOVP	1		Move to point 1 at 100 mm/s.
[Example 2]	VEL	500		Set the speed to 500 mm/s.
	MOVP	2		Move to point 2 at 500 mm/s.
[Example 3]	LET	1	300	Assign 300 to variable 1.
	VEL	*1		Set the speed to the content of variable 1 (300 mm/s).

- OVRD (Override)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OVRD	Speed ratio	Prohibited	CP

[Function] Reduce the speed in accordance with the ratio specified in operand 1 (speed coefficient setting). The speed ratio is set in a range from 1 to 100%.
A speed command specifying a speed below 1 mm/sec can be generated using OVRD.

[Example 1]

VEL	100	Set the speed to 100 mm/s.
OVRD	50	Reduce the speed to 50%. As a result, the actual speed will become 50 mm/s.

Command limit speed for smooth operation: Travel distance per encoder pulse [mm/pulse]/time [msec]
Command limit speed that can be generated: Travel distance per encoder pulse [mm/pulse]/time [256 msec]
(Smoothness of actual operation cannot be guaranteed. Movement must be checked on the actual machine.)
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001 mm] x Gear ratio numerator)
/ (Encoder resolution [pulses/rev] x Gear ratio denominator
/ (2 ^ Encoder division ratio)

Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001 μm/pulse) x 1000
/ (2 ^ Encoder division ratio)

(Reference) Use the values of the following parameters for the above calculation formulas:

Encoder resolution:	Axis-specific parameter No. 42
Encoder division ratio:	Axis-specific parameter No. 43
Screw lead:	Axis-specific parameter No. 47
Gear ratio numerator:	Axis-specific parameter No. 50
Gear ratio denominator:	Axis-specific parameter No. 51

- ACC (Set acceleration)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ACC	Acceleration	Prohibited	CP

[Function] Set the travel acceleration of the actuator.
The maximum acceleration will vary depending on the load and model of the actuator connected.
The acceleration is set in G and may include up to two decimal places.

(Note) If the position data contains no acceleration AND acceleration is not set by an ACC command, the actuator will move based on the default value set in "All-axis parameter No. 11, Default acceleration."

[Example 1] ACC 0.3 Set the acceleration to 0.3 G.

(Note) Setting an acceleration exceeding the specified range for the actuator may generate an error. It may also result in a failure or shorter product life.

- DCL (Set deceleration)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	DCL	Deceleration	Prohibited	CP

[Function] Set the travel deceleration of the actuator.
The maximum deceleration will vary depending on the load and model of the actuator connected.
The deceleration is set in G and may include up to two decimal places.

(Note) If the position data contains no deceleration AND deceleration is not set by a DCL command, the actuator will move based on the default value set in “All-axis parameter No. 12, Default deceleration.”
A DCL command cannot be used with CIR and ARC commands.

[Example] DCL 0.3 Set the deceleration to 0.3 G.

(Note) Setting a deceleration exceeding the specified range for the actuator may generate an error.
It may also result in a failure or shorter product life.

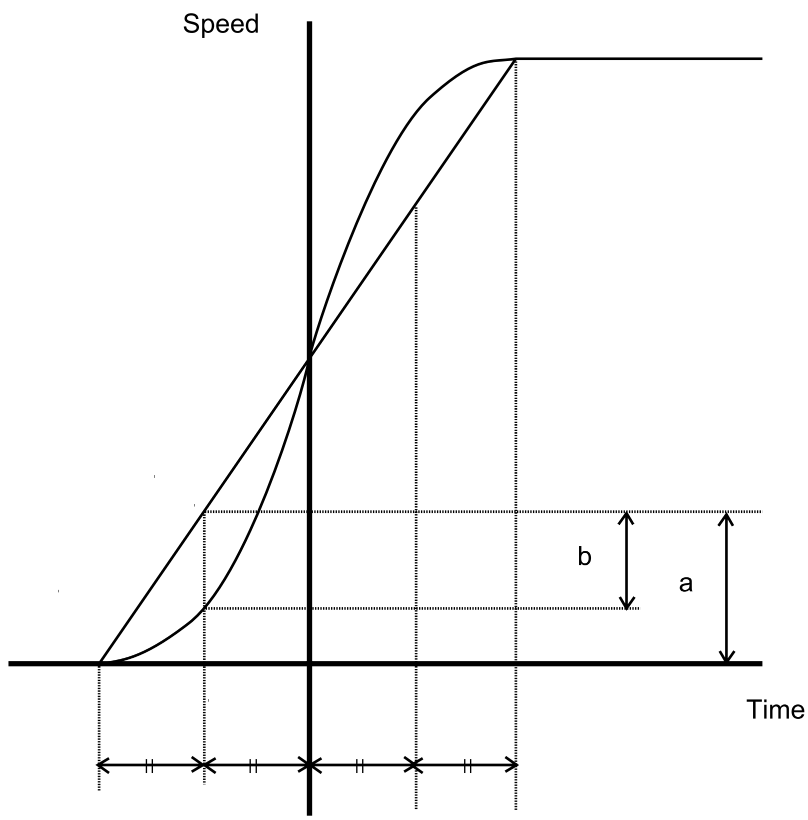
- SCRV (Set sigmoid motion ratio)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SCRV	Ratio	Prohibited	CP

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand 1. The ratio is set as an integer in a range from 0 to 50 (%).

$$\frac{b}{a} \text{ X } 100\ (\%)$$

If the ratio is not set using this command or 0% is set, a trapezoid motion will be implemented.
A SCRV command can be used with the following commands:
MOVP, MOVL, MVP○, MVL○, JBWF, JBWN, JFWF, JFWN
MOVD, MODI



[Example 1] SCRV 30 Set the sigmoid motion ratio to 30%.

- OFST (Set offset)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OFST	Axis pattern	Offset value	CP

[Function] Reset the target value by adding the offset value specified in operand 2 to the original target value when performing the actuator movement specified in operand 1.
The offset is set in mm, and the effective resolution is 0.001 mm.
A negative offset may be specified as long as the operation range is not exceeded.
An OFST command is processed with respect to soft axes before a BASE shift.

(Note) An OFST command cannot be used outside the applicable program. To use OFST in multiple programs, the command must be executed in each program.
An OFST command cannot be used with MVPI, MVLI and MVDI commands.

[Example 1]

```
OFST    10    50    Add 50 mm to the specified position of axis 2.
  :
OFST    10    0     Return the offset of axis 2 to 0.
```

[Example 2]

The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
10 (binary) → 2 (decimal)

```
LET     1     2     Assign 2 to variable 1.
OFST    *1    50
  :
OFST    *1    0
```

[Example 3]

```
LET     2     50    Assign 50 to variable 2.
OFST    1     *2    Add the content of variable 2 (50 mm) to the specified
                    position of axis1.
```

- DEG (Set arc angle)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	DEG	Angle	Prohibited	CP

[Function] Set a division angle for the interpolation implemented by a CIR (move along circle) or ARC (move along arc) command.
When CIR or ARC is executed, a circle will be divided by the angle set here to calculate the passing points.
The angle is set in a range from 0 to 120 degrees.
If the angle is set to "0," an appropriate division angle will be calculated automatically so that the actuator will operate at the set speed (maximum 180 degrees).
The angle is set in degrees and may include up to one decimal place.

(Note) If a CIR or ARC command is executed without setting an angle with this command, the default value registered in "All-axis parameter No. 30, Default division angle" will be used.

[Example] DEG 10 Set the division angle to 10 degrees.

- BASE (Specify axis base)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	BASE	Axis number	Prohibited	CP

[Function] Count the axes sequentially based on the axis number specified in operand 1 being the first axis.
A BASE command can be used with PRED, PRDQ, AXST, actuator-control and zone commands. Note that each zone range is assigned to the actuator via parameter.

[Example 1]

```
HOME    1          Axis 1 returns to the home.
BASE    2          Axis 2 is considered the first axis.
HOME    1          Axis 2 returns to the home.
```

[Example 2]

```
LET     1     2    Assign 2 to variable 1.
BASE    *1         The content of variable 1 (axis 2) will be considered as the
                   first axis.
```

- GRP (Set group axes)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	GRP	Axis pattern	Prohibited	CP

[Function] Allow only the position data of the axis pattern specified in operand 1 to become valid.
The program assumes that there are no data for other axes not specified.
When multiple programs are run simultaneously, assigning axes will allow the same position data to be used effectively among the programs.
A GRP command can be used with operand axis-pattern specification commands excluding an OFST command, as well as with servo operation commands using position data.
A GRP command is processed with respect to soft axes before a BASE shift.

[Example 1] GRP 10 Data of axis 2 becomes valid.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
10 (binary) → 2 (decimal)

```
LET    1    2    Assign 2 to variable 1.
GRP    *1
```

- HOLD (Hold: Declare axis port to pause)

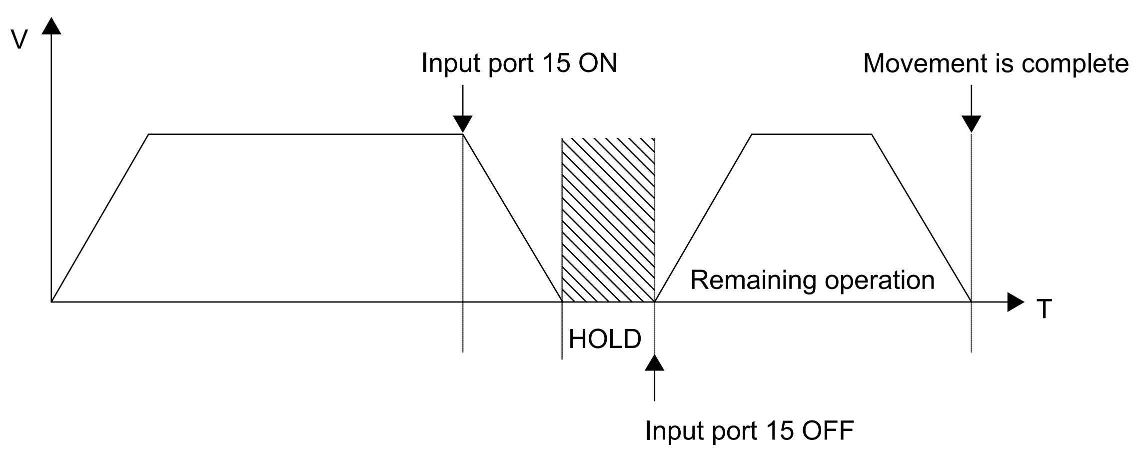
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	HOLD	(Input port, global flag)	(HOLD type)	CP

[Function] Declare an input port or global flag to pause while a servo command is being executed. When operation is performed on the input port or global flag specified in operand 1, the current servo processing will pause. (If the axes are moving, they will decelerate to a stop.) If nothing is specified in operand 1, the current pause declaration will become invalid.

A HOLD type can be specified in operand 2.
[HOLD type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)
2 = Contact b (Deceleration stop → Servo OFF (The drive source will not be cut off))
The HOLD type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current HOLD type will be used.
Using other task to issue a servo ON command to any axis currently stopped via a HOLD servo OFF will generate an “Error No. C66, Axis duplication error.” If the servo of that axis was ON prior to the HOLD stop, the system will automatically turn on the servo when the HOLD is cancelled. Therefore, do not issue a servo ON command to any axis currently stopped via a HOLD servo OFF.
If any axis currently stopped via a HOLD servo OFF is moved by external force, etc., from the stopped position, and when the servo of that axis was ON prior to the HOLD stop, the axis will move to the original stopped position when the HOLD is cancelled before resuming operation.

(Note 1) The input port or global flag specified by a HOLD declaration will only pause the axes used in the task (program) in which the HOLD is declared. The declaration will not be valid on axes used in different tasks (programs).
(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in J☐W☐ and PATH operations.)
(Note 3) Following a pause of home return, the operation will resume from the beginning of the home-return sequence.

[Example] HOLD 15 0 The axes will decelerate to a stop when input port 15 turns ON.



- CANC (Cancel: Declare axis port to abort)

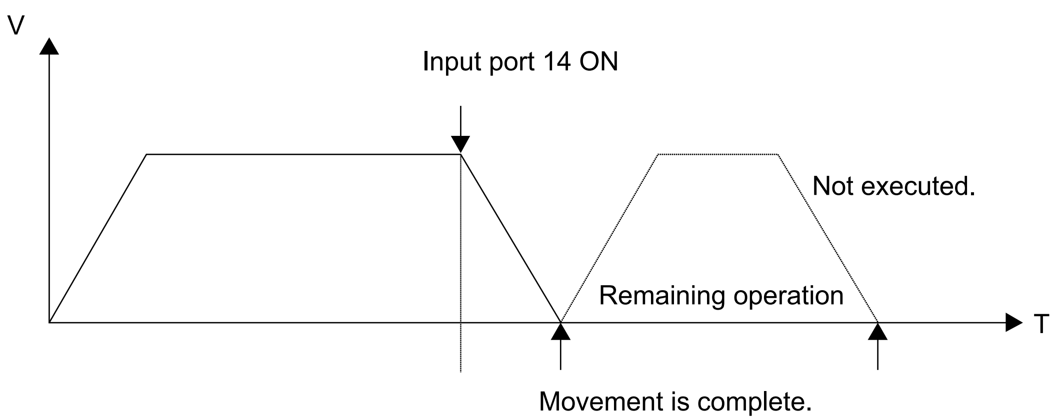
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CANC	(Input port, global flag)	(CANC type)	CP

[Function] Declare an input port or global flag to abort while a servo command is being executed. When operation is performed on the input port or global flag specified in operand 1, the current servo processing will be aborted. (If the axes are moving, they will decelerate to a stop before the processing is aborted.)
If nothing is specified in operand 1, the current abort declaration will become invalid.

A CANC type can be specified in operand 2.
[CANC type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)
The CANC type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current CANC type will be used.

(Note 1) The input port or global flag specified by a CANC command will only abort the axes used in the task (program) in which the CANC is declared. The declaration will not be valid on axes used in different tasks (programs).
(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in JXWX and PATH operations.)

[Example] CANC 14 0 The axes will decelerate to a stop when input port 14 turns ON.



- VLMX (Specify VLMX speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	VLMX	Prohibited	Prohibited	CP

[Function] Set the actuator travel speed to the VLMX speed (normally maximum speed).
Executing a VLMX command will set the value registered in "Axis-specific parameter No. 29, VLMX speed" as the travel speed.

(Note) If the VLMX speed is specified in a continuous position travel command (PATH, PSPL), the target speed to each position will become a composite VLMX speed not exceeding the maximum speed of each axis set in "Axis-specific parameter No. 28, Maximum operating speed of each axis." To make the target speed constant, a desired speed must be expressly specified using a VEL command.

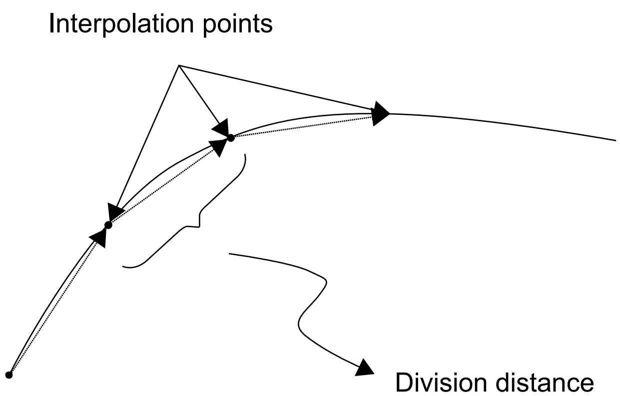
[Example]

```
VEL     1000  ┐
MOVP    1     │ The speed becomes 1000 mm/sec in this section.
MOVP    2     ┘
VLMX          ┐
MOVP    3     │ The speed becomes VLMX mm/sec in this section.
MOVP    4     ┘
```

- DIS (Set division distance at spline movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	DIS	Distance	Prohibited	CP

[Function] Set a division distance for the interpolation implemented by a PSPL (move along spline) command.
When a PSPL command is executed, a passing point will be calculated at each distance set here and the calculated passing points will be used as interpolation points.
If the distance is set to "0," an appropriate division distance will be calculated automatically so that the actuator will operate at the set speed
The distance is input in mm.



(Note) If a PSPL command is executed without setting a distance with a DIS command, the default value registered in "All-axis parameter No. 31, Default division distance" will be used.

[Example] DIS 10 Set the division distance to 10 mm.

- POTP (Set PATH output type)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	POTP	0 or 1	Prohibited	CP

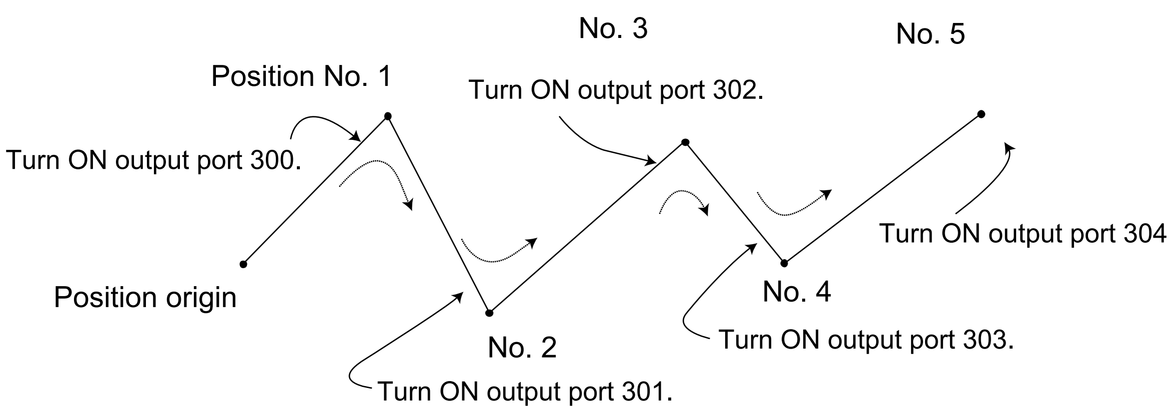
[Function] Set the output type in the output field to be used when a PATH or PSPL command is executed.
When a PATH or PSPL command is executed, the output will operate as follows in accordance with the setting of the POTP command.

(1) POTP [Operand 1] = 0 (ON upon completion of operation)
The output port or flag will turn ON upon completion of operation.
(2) POTP [Operand 1] = 1 (Increment and output on approaching each position; ON upon completion of operation for the last position)
During PATH or PSPL operation, the output port number or flag number specified in the output field will be incremented and turned ON when each specified position approaches. At the last position, however, the output will turn ON upon completion of operation. This setting provides a rough guide for output in sequence control.

(Note 1) The default value of POTP, before it is set, is “0.”
(Note 2) If POTP = 1 and there is no valid data at the specified position, the output number will be incremented but the output will not turn ON. (The output number will be incremented regardless of the size of position numbers specified in operands 1 and 2 in a PATH or PSPL command.)

[Example]

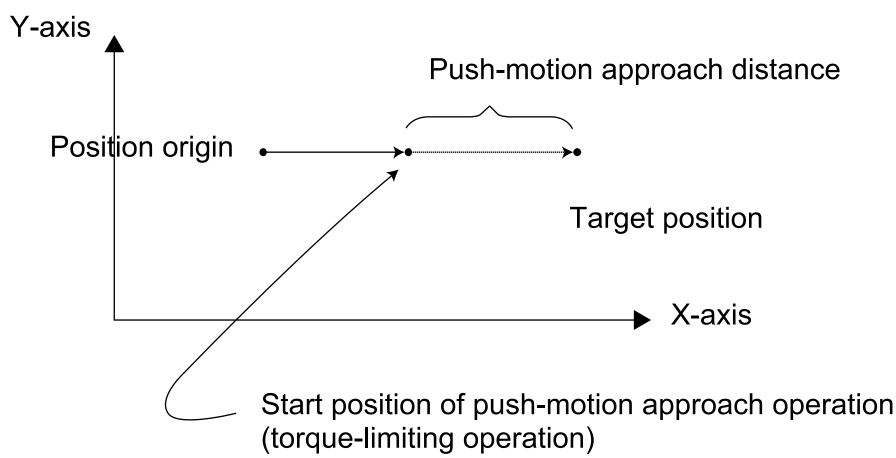
POTP	1			
PATH	1	5	300	Turn ON output port Nos. 300 through 304 sequentially each time a specified position approaches during a pass movement from position Nos. 1 through 5, starting from the first position.



- PAPR (Set push-motion approach distance, speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPR	Distance	Speed	CP

[Function] Set the operation to be performed when a PUSH command is executed.
Set the distance (push-motion approach distance) over which push-motion approach operation (torque-limiting operation) will be performed in operand 1 (in mm), and set the speed (push-motion approach speed) at which push-motion approach operation (torque-limiting operation) will be performed in operand 2 (in mm/sec).
The push-motion approach distance specified in operand 1 may contain up to three decimal places, while the speed specified in operand 2 cannot contain any decimal place.



[Example]	PAPR	100	30	Set the push-motion approach distance in a PUSH command to 100 mm and the push-motion approach speed to 30 mm/sec.
	MOVP	2		Move to position No. 2.
	PUSH	10		Move by push-motion from position No. 2 to position No. 10.

(Note) The push-motion approach speed in an OVRD command will be clamped by the minimum speed of 1 mm/sec. (Correct push-motion operation is not guaranteed at the minimum speed. Operation at slow push-motion approach must be checked on the actual machine by considering the effects of mechanical characteristics, etc.)

- QRTN (Set quick-return mode)

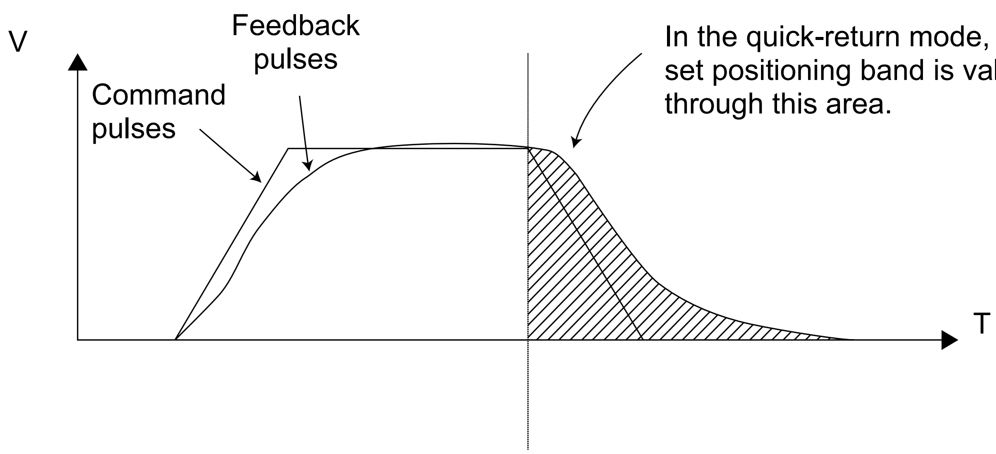
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	QRTN	0 or 1	Prohibited	CP

[Function] Set and cancel the quick-return mode.

(1) QRTN [Operand 1] = 0 (Normal mode)
Positioning is deemed complete when all command pulses have been output and the current position is inside the positioning band.

* If a deceleration command is currently executed in the quick-return mode, the system will wait for all command pulses to be output.

(2) QRTN [Operand 1] = 1 (Quick-return mode)
Positioning is deemed complete when "a normal deceleration command is currently executed (excluding deceleration due to a stop command, etc.) or all command pulses have been output" AND "the current position is inside the positioning band." This setting is used to perform other processing during deceleration, in conjunction with a PBND command.



(Note 1) The quick-return mode will be cancelled when the program ends. (The positioning band set by a PBND command will not be cancelled.)

(Note 2) If a given axis is used even once in the quick-return mode, the program will not release the right to use the axis until the QRTN is set to "0" (normal mode) or the program ends. Any attempt to use the axis from other program will generate an "Error No. C66, Axis duplication error."

(Note 3) Following a return from a normal deceleration command in the quick-return mode, the next positioning will start after all command pulses for the previous positioning have been output. Therefore, in the quick-return mode a simple reciprocating operation will require a longer tact time because of the extra completion check. In this sense, this setting should be used only if you wish to reduce the overall tact time by performing other processing during deceleration.

(Note 4) The quick-return mode represents very irregular processing. Therefore, be sure to revert to the normal mode when the overlay processing is completed in the necessary section.

(Note 5) The quick-return mode cannot be used with a push-motion travel command or arc interpolation command.

1.12 Actuator Control Command

- SV☐☐ (Turn ON/OFF servo)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SV☐☐	Axis pattern	Prohibited	PE

[Function] Turn ON/OFF the servos of the axes specified by the axis pattern in operand 1.

```
SV☐☐
 └┬─ON  .............. Turn ON the servo.
  └─OF  .............. Turn OFF the servo.
```

[Example 1] SVON 11 Turn ON the servos of axes 1 and 2. Nothing will occur if the axis servos are already ON.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    1    3    Assign 3 to variable 1.
SVON   *1
```

> ⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

- HOME (Return to home)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	HOME	Axis pattern	Prohibited	PE

[Function] Perform home return of the axes specified by the axis pattern in operand 1.
The servo of each home-return axis will turn ON automatically.
The output will turn OFF at the start of home return, and turn ON when the home return is completed.

(Note) Following a pause of home return, the operation will resume from the beginning of the home-return sequence.
The home-return operation of an absolute-encoder axis is a movement to the rotation data reset position and may not necessarily be a movement to the preset home coordinate (including 0). If an output function specification value of “12” (All-valid-axed home (=0) output) or “14” (All-valid-axes preset home coordinate output) is stored in the I/O parameter “Output function setting nnn,” use a MOVP command, not a HOME command, when moving each absolute-encoder axis for the purpose of turning ON the applicable output.
If the operation is stopped or cancelled while a HOME command is being executed for an absolute-encoder axis in a mode other than the absolute reset mode provided by the PC software or teaching pendant, an “actual-position soft limit error” may generate depending on the position. It is not recommended to perform home return other than for the purpose of adjusting an absolute-encoder axis.

[Example 1] HOME 11 Axes 1 and 2 return to the home.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     1     3     Assign 3 to variable 1.
HOME    *1
```

⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
Put the slider or rod away from the mechanical end when turning the servo ON.

- MOVP (Move PTP by specifying position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MOVP	Position number	Prohibited	PE

[Function] Move the actuator to the position corresponding to the position number specified in operand 1, without interpolation (PTP stands for "Point-to-Point").
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

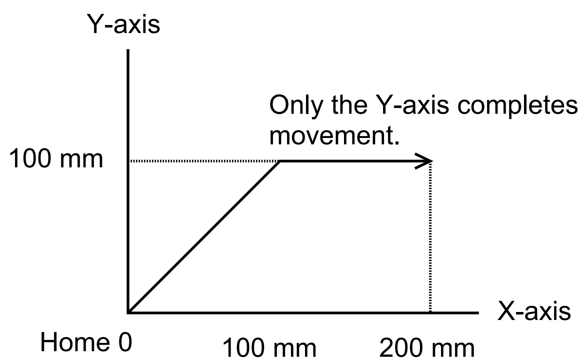
[Example 1]	VEL	100		Set the speed to 100 mm/s.
	MOVP	1		Move the axes to the position corresponding to position No. 1 (200, 100).
[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	2	Assign 2 to variable 1.
	MOVP	*1		Move the axes to the position corresponding to the content of variable 1 (position No. 2, or (100, 100)).

Position data display in PC software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	200.000	100.000			
2	100.000	100.000			

(Note) If no position data is available and acceleration and deceleration are not specified by an ACC (DCL) command, each axis will move according to all-axis parameter No. 11, "Default acceleration" and all-axis parameter No. 12, "Default deceleration."

Travel path from the home to the position corresponding to position No. 1 (200, 100)



Each axis moves at 100 mm/s.

- MOVL (Move by specifying position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MOVL	Position number	Prohibited	PE

[Function] Move the actuator to the position corresponding to the position number specified in operand 1, with interpolation.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

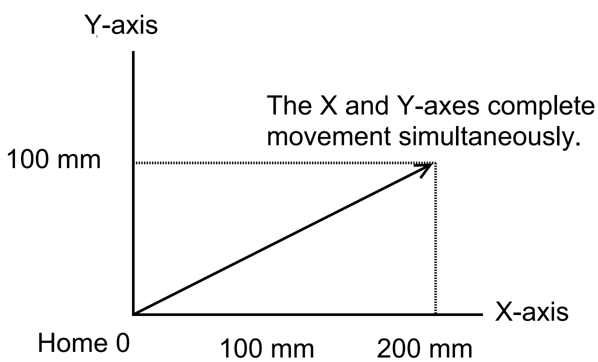
[Example 1]	VEL	100		Set the speed to 100 mm/s.
	MOVL	1		Move the axes to the position corresponding to position No. 1 (200, 100), with interpolation.
[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	2	Assign 2 to variable 1.
	MOVL	*1		Move the axes to the position corresponding to the content of variable 1 (position No. 2, or (100, 100)), with interpolation.

Position data display in PC software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	200.000	100.000			
2	100.000	100.000			

(Note) If no position data is available and acceleration and deceleration are not specified by an ACC (DCL) command, each axis will move according to all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration.”

Travel path from the home to the position corresponding to position No. 1 (200, 100)



The tip of each axis moves at 100 mm/s.

- MVPI (Move via incremental PTP)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MVPI	Position number	Prohibited	PE

[Function] Move the actuator, without interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1. The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

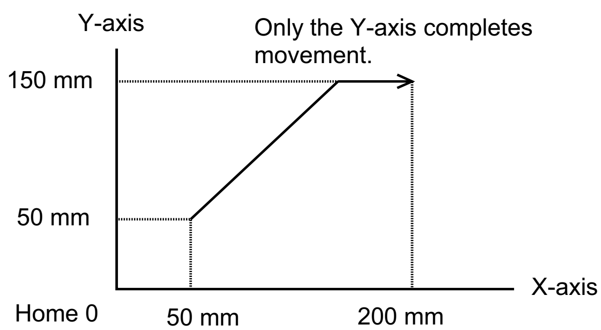
[Example 1]	VEL	100		Set the speed to 100 mm/s.
	MVPI	1		If the current position is (50, 50) and position No. 1 is set to (150, 100), the axes will move 150 in the X direction and 100 in the Y direction (200, 150) from the current position.
[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	2	Assign 2 to variable 1.
	MVPI	*1		Move from the current position by the travel distance corresponding to the content of variable 1 (position No. 2, or (100, 100)).

Position data display in PC software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	150.000	100.000			
2	100.000	100.000			

(Note) If no position data is available and acceleration and deceleration are not specified by an ACC (DCL) command, each axis will move according to all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration.”

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



Each axis moves at 100 mm/s.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001 mm] x Gear ratio numerator)
/ (Encoder resolution [pulses/rev] x Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001 μm/pulse) x 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution: Axis-specific parameter No. 42
Encoder division ratio: Axis-specific parameter No. 43
Screw lead: Axis-specific parameter No. 47
Gear ratio numerator: Axis-specific parameter No. 50
Gear ratio denominator: Axis-specific parameter No. 51

● MVLI (Move via incremental interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MVLI	Position number	Prohibited	PE

[Function] Move the actuator, with interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1. The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

[Example 1]	VEL	100		Set the speed to 100 mm/s.
	MVLI	1		If the current position is (50, 50) and position No. 1 is set to (150, 100), the axes will move 150 in the X direction and 100 in the Y direction (200, 150) from the current position, with interpolation.

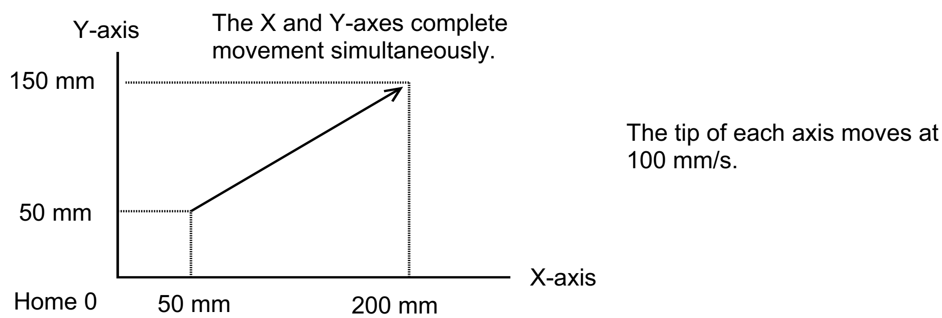
[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	2	Assign 2 to variable 1.
	MVLI	*1		Move from the current position by the travel distance corresponding to the content of variable 1 (position No. 2, or (100, 100)).

Position data display in PC software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	150.000	100.000			
2	100.000	100.000			

(Note) If no position data is available and acceleration and deceleration are not specified by an ACC (DCL) command, each axis will move according to all-axis parameter No. 11, "Default acceleration" and all-axis parameter No. 12, "Default deceleration."

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001 mm] x Gear ratio numerator) / (Encoder resolution [pulses/rev] x Gear ratio denominator / (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001 μm/pulse) x 1000 / (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution: Axis-specific parameter No. 42
Encoder division ratio: Axis-specific parameter No. 43
Screw lead: Axis-specific parameter No. 47
Gear ratio numerator: Axis-specific parameter No. 50
Gear ratio denominator: Axis-specific parameter No. 51

- MOVD (Move via direct value specification)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MOVD	Target position	(Axis pattern)	PE

[Function] Move the axis specified by the axis pattern in operand 2, to the target position corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The target position is set in mm, and the set value is valid to the third decimal place.

[Example 1]	MOVD	100	10	Move axis 2 to position 100.
[Example 2]	LET	1	100	Assign 100 to variable 1.
	MOVD	*1	11	Move all axes to the content of variable 1 (100).

- MVDI (Move relatively via direct value specification)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	MVDI	Travel distance	(Axis pattern)	PE

[Function] Move the axis specified by the axis pattern in operand 2 from its current position by the travel distance corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The travel distance is set in mm, and the set value is valid to the third decimal place.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001 mm] x Gear ratio numerator) / (Encoder resolution [pulses/rev] x Gear ratio denominator / (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001 μm/pulse) x 1000 / (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:

Encoder resolution:	Axis-specific parameter No. 42
Encoder division ratio:	Axis-specific parameter No. 43
Screw lead:	Axis-specific parameter No. 47
Gear ratio numerator:	Axis-specific parameter No. 50
Gear ratio denominator:	Axis-specific parameter No. 51

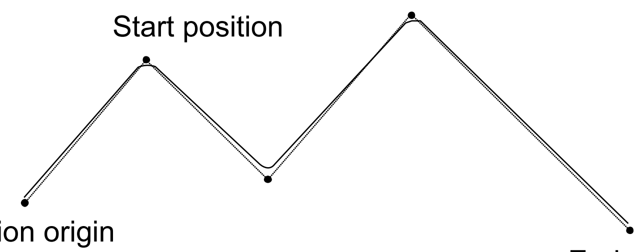
[Example 1] MVDI 30 11 Move all axes from the current position by 30 mm in the positive direction.

[Example 2] LET 1 -100 Assign -100 to variable 1.
MVDI *1 1 Move axis 1 from the current position in accordance with the content of variable 1 (-100), or by 100 mm in the negative direction.

- PATH (Move along path)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PATH	Start position number	End position number	PE

[Function] Move continuously from the position specified in operand 1 to the position specified in operand 2.
The output type in the output field can be set using an actuator-declaration command POTP.
Increasing the acceleration will make the passing points closer to the specified positions.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.

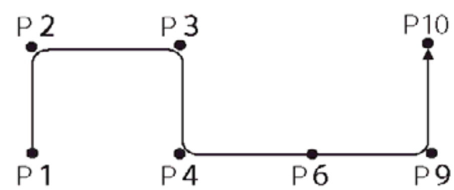


(Note 1) Multi-dimensional movement can be performed using a PATH command.
In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

(Note 2) Continuous movement through positions is possible even when the specified positions are not continuous.
To do this, specify each discontinuous position number as both the start position number and end position number in a PATH command, as shown in the example. In this example, position No. 6 is discontinuous.

[Example] Move continuously through position Nos. 1, 2, 3, 4, 6, 9 and 10 in this order.

```
PATH    1    4
PATH    6    6 (Discontinuous position)
PATH    9    10
```



[Example 1]

```
VEL     100          Set the speed to 100 mm/s.
PATH    100   120    Move continuously from position Nos. 100 to 120.
```

[Example 2]

```
VEL     100          Set the speed to 100 mm/s.
LET     1     50     Assign 50 to variable 1.
LET     2     100    Assign 100 to variable 2.
PATH    *1    *2     Move continuously along the positions from the content of
                     variable 1 (position No. 50) to the content of variable 2
                     (position No. 100).
```

- J□W□ (Jog)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	J□W□	Axis pattern	Input, output, flag number	PE

[Function] The axes in the axis pattern specified in operand 1 will move forward or backward while the input or output port or flag specified in operand 2 is ON or OFF.

JBWF..........Move backward while the specified port is OFF.
JBWN..........Move backward while the specified port is ON.
JFWF..........Move forward while the specified port is OFF.
JFWN..........Move forward while the specified port is ON.

(Note 1) This command is also valid on an axis not yet completing home return. In this case, the maximum speed will be limited by “All-axis parameter No. 15, Maximum jog speed before home return.” Since coordinate values do not mean anything before home return, pay due attention to prevent contact with the stroke ends.

[Example 1]

```
VEL    100          Set the speed to 100 mm/s.
JBWF   11     10    Move axes 1 and 2 backward while input 10 is OFF.
```

[Example 2]

The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
VEL    100          Set the speed to 100 mm/s.
LET    1      3     Assign 3 to variable 1.
JBWF   *1     10
```

[Example 3]

```
VEL    100          Set the speed to 100 mm/s.
LET    5      20    Assign 20 to variable 5.
JFWN   10     *5    Move axis 2 forward while the content of variable 5 (input
                    20), is ON.
```

- STOP (Stop movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	STOP	Axis pattern	Prohibited	CP

[Function] Decelerate and stop the axes specified by the axis pattern in operand 1.

(Note 1) A STOP command can be used with all active servo commands other than a SVOF command.

(Note 2) A STOP command only issues a deceleration-stop command (operation stop) to a specified axis pattern and does not wait for stopping to complete. Issuing other servo commands to a decelerating axis will either become invalid or generate an “axis duplication error,” etc.
Set a timer, etc., in the program so that the next servo command will be issued after a sufficient deceleration-stop processing time elapses.
Even when a STOP command is to be issued to an axis currently stopped, provide a minimum interval of 0.1 second before the next servo command is issued.

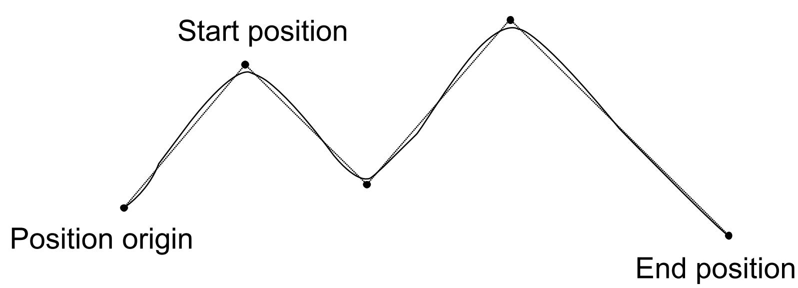
[Example 1] STOP 11 Decelerate and stop axes 1 and 2.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 1 3 Assign 3 to variable 1.
STOP *1

- PSPL (Move along spline)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PSPL	Start position number	End position number	PE

[Function] Continuously move from the specified start position to end position via interpolation along a spline-interpolation curve.
The output type in the output field can be set using an actuator-declaration command POTP.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.



(The above diagram is only an example.)

(Note) If the acceleration and deceleration are different between points, the speeds will not be connected smoothly.

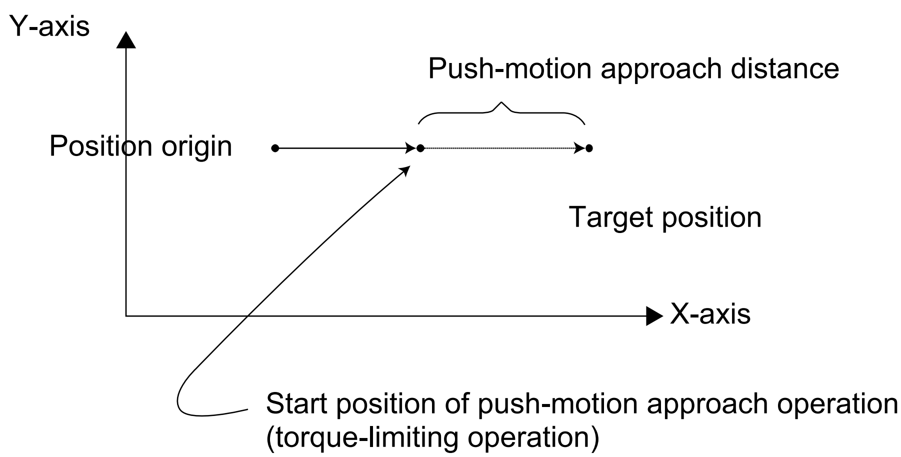
In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

[Example]	VEL	100		Set the speed to 100 mm/s.
	PSPL	100	120	Continuously move from position Nos. 100 to 120 along a spline-interpolation curve.

- PUSH (Move by push motion)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PUSH	Target position number	Prohibited	PE

[Function] Perform push-motion operation until the target position specified in operand 1 is reached. The axes move in a normal mode from the position origin to the push-motion approach start position as determined by a PAPR command, after which push-motion approach operation (toque-limiting operation) will be performed. The speed of push-motion approach operation (toque-limiting operation) is determined by the push-motion approach speed specified by a PAPR command. If the output field is specified, the output will turn ON when a contact is confirmed, and turn OFF when a missed contact is detected.



The push force can be adjusted using "Driver parameter No. 38, Push torque limit at positioning" (default value: 70%).

(Note 1) A PUSH command only moves a single axis. If multiple axes are specified, an "Error No. C91, Multiple push-axes specification error" will generate.

(Note 2) A push-motion approach speed exceeding the maximum speed permitted by the system will be clamped at the maximum speed. (The maximum system speed is not the maximum practical speed. Determine a practical speed by considering the impact upon contact, etc.)

(Note 3) Push-motion operation cannot be performed with a synchro controller.

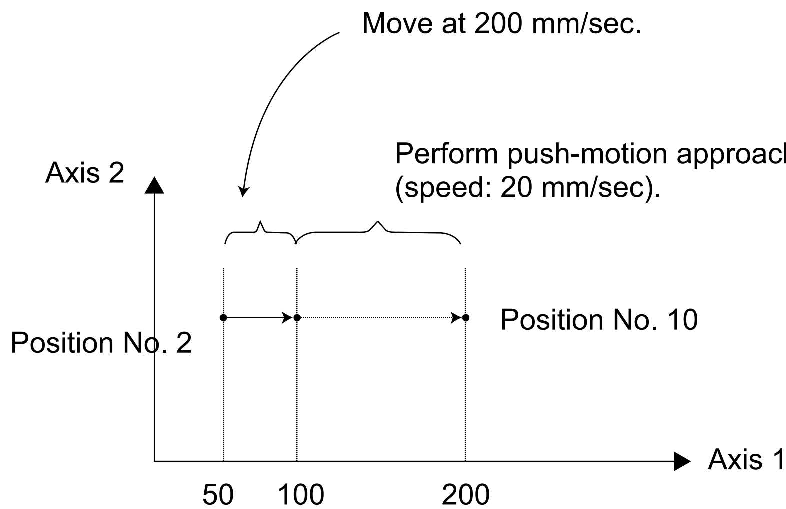
[Example]	PAPR	100	20
	MOVP	2	
	PUSH	10	

Set the push-motion approach distance to 100 mm and push-motion approach speed to 20 mm/sec.
Move from the current position to position No. 2.
Perform push-motion movement from position Nos. 2 to 10.

The diagram below describes a push-motion movement based on the position data shown in the table below:

Position data display in PC software

Position No.	Axis 1	Axis 2	Vel	Acc	Dcl
1					
2	50.000	100.000			
• • • •					
10	200.000		200	0.30	0.30
• •					



- PTRQ (Change push torque limit parameter)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PTRQ	Axis pattern	Ratio	CC

[Function] Change the push torque limit parameter of the axis pattern specified in operand 1 to the value in operand 2. Operand 2 is set as an integer (unit: %).
A PTRQ command temporarily rewrites "Driver parameter No. 38: Push torque limit at positioning."

(Note 1) If a push torque limit is not set by a PTRQ command, the value set in "Driver parameter No. 38: Push torque limit at positioning" will be used.
(Note 2) The new push torque limit will remain effective even after the program ends. Therefore, when building a system using the PTRQ command, in every program explicitly specify a push torque limit using a PTRQ command before each push-motion operation. Assuming that the push torque limit will be reset to the original value when push-motion operation ends in one program can cause an unexpected problem in another program, because a different push torque limit will be used if the program is aborted due to an error, etc.
(Note 3) The new value set by a PTRQ command will become ineffective after a power-on reset or software reset.
(Note 4) A PTRQ command does not rewrite "Driver parameter No. 38: Push torque limit at positioning" (main CPU flash memory (non-volatile memory)).

[Example]

PTRQ	1	50	Change the push torque limit parameter for axis 1 to 50%.
PAPR	100	20	Set the push-motion approach distance to 100 mm and the push-motion approach speed to 20 mm/sec.
MOVP	2		Move to position No. 2.
PUSH	10		Move by push motion from position No. 2 to position No. 10.

- CIR2 (Move along circle 2 (arc interpolation))

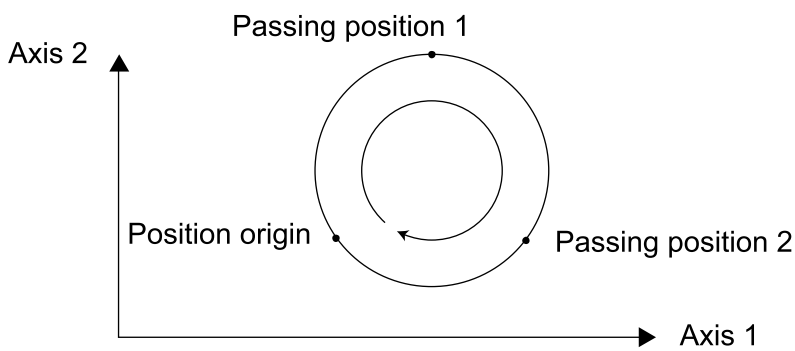
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CIR2	Passing position 1 number	Passing position 2 number	PE

[Function] Move along a circle originating from the current position and passing positions 1 and 2, via arc interpolation.
The rotating direction of the circle is determined by the given position data.
The diagram below describes a CW (clockwise) movement. Reversing passing positions 1 and 2 will change the direction of movement to CCW (counterclockwise).

The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting in the position data specified in operand 1	Setting in the position data specified in operand 1
2	Setting by VEL command	Setting by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

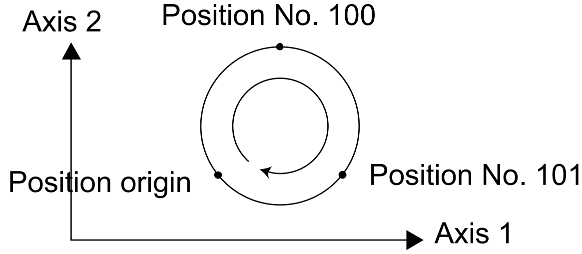
If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.)

[Example]

VEL	100		Set the speed to 100 mm/s.
CIR2	100	101	Move along a circle (circular interpolation) passing position Nos. 100 and 101.



- ARC2 (Move along circle 2 (arc interpolation))

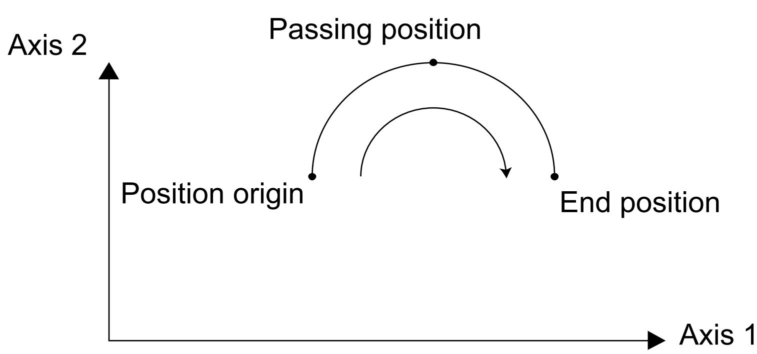
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ARC2	Passing position number	End position number	PE

[Function] Move along an arc originating from the current position, passing the specified position and terminating at the end position, via arc interpolation.

The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting in the position data specified in operand 1	Setting in the position data specified in operand 1
2	Setting by VEL command	Setting by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

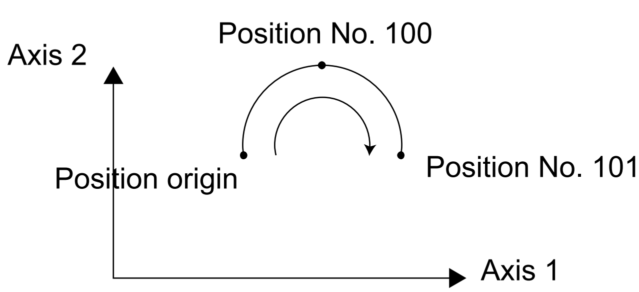
If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.)

[Example]

VEL	100		Set the speed to 100 mm/s.
ARC2	100	101	Move along an arc (circular interpolation) from the current position to position No. 101 by passing position No. 100.



● CHVL (Change speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CHVL	Axis pattern	Speed	CP

[Function] Change the speed of the axes operating in other task.
When a CHVL command is executed, the speed of the axes specified in operand 1 will change to the value specified in operand 2.

(Note 1) This command is not valid on an axis operated by a CIR, ARC, PSPL, PUSH, or ARCH command.
(Note 2) Executing a CHVL command for an axis operating in sigmoid motion (SCRV command) will generate an “Error No. CC1, Speed-change condition error.”
(Note 3) This is a temporary speed-change command issued from other task to the active packet (point). It is not affected by the data declared by VEL.

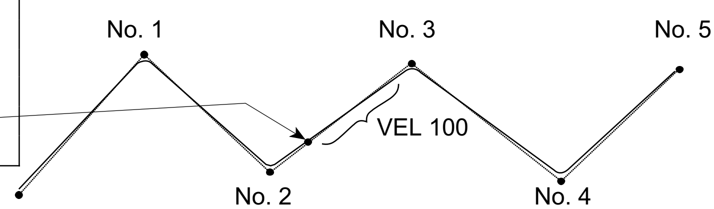
Program 1	Program 2
	VEL 300
	•
	•
	MOVP 1
CHVL 11 100	MOVP 2
	MOVP 3
	•
	•

If CHVL is executed in program 1 while MOVP 2 is executed in program 2, the travel speed of MOVP 2 will become 100 mm/sec.
The speeds of other move commands will remain 300 mm/sec.

The axis pattern can be specified indirectly using a variable.
When program 1 is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 1 3 Assign 3 to variable 1.
CHVL *1 100

(Note 4) Since this command is valid only for the packet that is active at the time of execution of the command for an axis subject to continuous motion in a PATH command, etc., caution must be exercised against the timing shift. The packet handling will be put on hold during speed-change processing, so caution must also be exercised against the locus shift.

Program 1	Program 2
•	VEL 300
•	•
•	•
CHVL 11 100 →	PATH 1 5
•	
•	
•	



If CHVL is executed in program 1 while PATH is executed in program 2, or specifically during the PATH movement from point No. 2 to point No. 3, the speed specified by CHVL (100 mm/sec in the above example) will become valid only during the PATH movement to point No. 3. Other travel speeds will remain at the speed specified by VEL (300 mm/sec in the above example).

(Note 5) Override of the CHVL call task will be applied, so caution must be exercised.
(Note 6) The maximum speed of the specified axis completing home return will be clamped by the minimum value set in “Axis-specific parameter No. 28, Maximum operating speed of each axis” or “Axis-specific parameter No. 27, Maximum speed limited by maximum motor speed” with respect to the specified axis and related interpolation axes currently operating. To prevent the maximum speed from being limited due to the effect of other axis whose maximum speed is lower than the speed specified in the CHVL command, issue a CHVL command in multiple steps corresponding to the respective axes having different maximum speeds. In particular, specification of a CHVL command in a separate step is recommended for a rotating axis.

[Example] CHVL 11 500 ⇒ CHVL 1 500
CHVL 10 500

● ARCD (Move along arc via specification of end position and center angle (arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ARCD	End position number	Center angle	PE

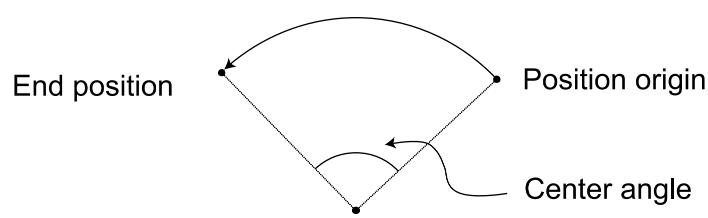
[Function] Move along an arc originating from the current position and terminating at the end position, via arc interpolation.
Specify the end position of movement in operand 1, and the center angle formed by the position origin and end position in operand 2. The center angle is set in a range from –359.999 to –0.001 or from 0.001 to 359.999. A positive value indicates CCW (counterclockwise) movement, while a negative value indicates CW (clockwise) movement.

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The center angle is set in degrees and may include up to three decimal places.

The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting in the position data specified in operand 1	Setting in the position data specified in operand 1
2	Setting by VEL command	Setting by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.)

[Example]

VEL	100		Set the speed to 100 mm/s.
ARCD	100	120	Move along an arc from the position origin to position No. 100 for a center angle of 120 degrees (CCW direction).

- ARCC (Move along arc via specification of center position and center angle (arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ARCC	Center position number	Center angle	PE

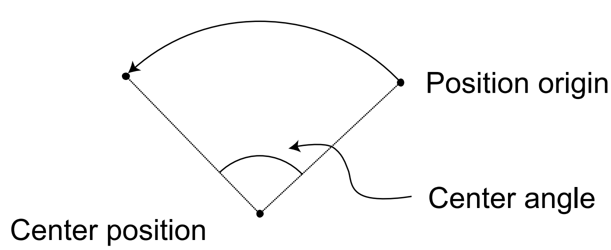
[Function] Move along an arc originating from the current position by keeping a specified radius from the center position, via arc interpolation.
Specify the center position in operand 1, and the center angle formed by the position origin and end position in operand 2. The center angle is set in a range from –3600 to 3600 degrees (±10 revolutions). A positive value indicates CCW (counterclockwise-direction) movement, while a negative value indicates CW (clockwise-direction) movement (setting unit: degree).

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The center angle is set in degrees and may include up to three decimal places.

The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting in the position data specified in operand 1	Setting in the position data specified in operand 1
2	Setting by VEL command	Setting by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

If speed is not set, a "C88 speed specification error" will generate.
If acceleration/deceleration is not valid, a "C89 acceleration/deceleration specification error" will generate.



(Note) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.)

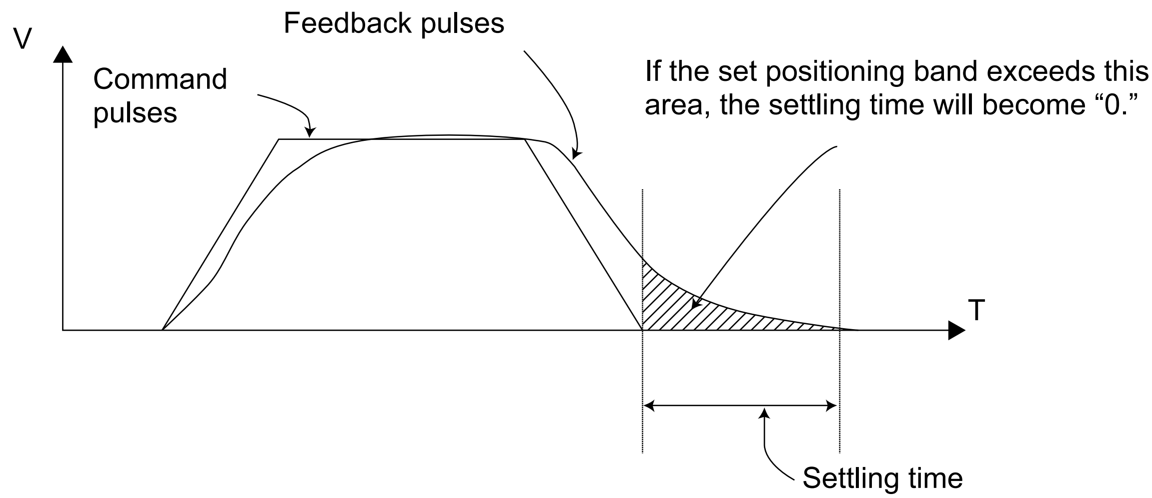
[Example]

VEL	100		Set the speed to 100 mm/s.
ARCC	100	120	Move along an arc from the position origin for a center angle of 120 degrees around position No. 100 being the center (CCW direction).

- PBND (Set positioning band)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PBND	Axis pattern	Distance	CP

[Function] Set the position complete width for the axes in the axis pattern specified in operand 1. The distance in operand 2 is set in mm.
As a rule, positioning is deemed complete when all command pulses have been output and the current position is inside the positioning band. Therefore, this command is effective if you wish to reduce the tact time by shortening the approximate positioning settling time. (Normally a setting of approx. 3 to 5 mm will have effect, but the effect must be confirmed on the actual machine.)
(This command can be combined with a QRTN command for special purposes. Refer to the section on QRTN command for details.)



(Note 1) If positioning band is not set with a PBND command, the value set in "Axis-specific parameter No. 58, Positioning band" will be used.
(Note 2) If the positioning band is changed, the new setting will remain valid even after the program ends. Therefore, to build a system using PBND commands, a positioning band must be expressly specified with a PBND command before operation of each program. An assumption that the positioning band will be reset to the original value when the operation ends in other program may lead to an unexpected problem, because the positioning band will become different from what is anticipated in case the applicable program is aborted due to error, etc.
(Note 3) The value set in "Axis-specific parameter No. 58, Positioning band" will not be written by a PBND command.

[Example 1] PBND 11 5 Set the positioning band for axes 1 and 2 to 5 mm after this command.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     1     3     Assign 3 to variable 1.
PBND    *1    5
```

- CIR (Move along circle)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CIR	Passing position 1 number	Passing position 2 number	PE

[Function] Move along a circle originating from the current position and passing the positions specified in operands 1 and 2.
Therefore, reversing the settings of operands 1 and 2 will implement a circular movement in the reverse direction.
The output will turn OFF at the start of circular movement, and turn ON when the movement is complete.
Difference from CIR2:
CIR processing resembles moving along a polygon with a PATH command, while CIR2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command. (Normally CIR2 is used.)

(Note 1) If the division angle is set to “0” with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than “0” with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)
In the case of circular movement, the axes will return from passing position 2 to the start position at the speed declared by a VEL command. Therefore, a VEL command must always be used with a CIR command.

(Note 3) The acceleration is selected in the order of the acceleration in the data at passing position 1, followed by the value in “All-axis parameter No. 11, Default acceleration.”
The deceleration will become the same value as the valid acceleration selected above. Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.)

[Example 1]	VEL	100		Set the speed to 100 mm/s.
	CIR	100	101	Move along a circle from the current position by passing positions 100 and 101 sequentially.

[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	5	Assign 5 to variable 1.
	LET	2	6	Assign 6 to variable 2.
	CIR	*1	*2	Move along a circle from the current position by passing the contents of variables 1 and 2 (positions 5 and 6) sequentially.

- ARC (Move along arc)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ARC	Passing position number	End position number	PE

[Function] Move along an arc from the current position to the position specified in operand 2, by passing the position specified in operand 1.
The output will turn OFF at the start of arc movement, and turn ON when the movement is complete.
Difference from ARC2:
ARC processing resembles moving along a polygon with a PATH command, while ARC2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command.
(Normally ARC2 is used.)

(Note 1) If the division angle is set to “0” with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than “0” with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)

(Note 3) The acceleration is selected in the order of the acceleration in the data at passing position 1, followed by the value in “All-axis parameter No. 11, Default acceleration.”
The deceleration will become the same value as the valid acceleration selected above.
Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) This command is valid on arbitrary orthogonal planes. (Axis 2 may be selected automatically prior to axis 1 in accordance with the position data.).

[Example 1]	VEL	100		Set the speed to 100 mm/s.
	ARC	100	101	Move along an arc from the current position to position 101 by passing position 100.

[Example 2]	VEL	100		Set the speed to 100 mm/s.
	LET	1	5	Assign 5 to variable 1.
	LET	2	6	Assign 6 to variable 2.
	ARC	*1	*2	Move along an arc from the current position to the content of variable 2 (position 6) by passing the content of variable 1 (position 5).

1.13 Structural IF

● IF☐☐ (Structural IF)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	IF☐☐	Variable number	Data	CP

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
If the input condition is not satisfied and the IF☐☐ command is not executed, the program will proceed to the step next to the corresponding EDIF.
A maximum of 15 nests are supported when IS☐☐ and DW☐☐ are combined.

```
IF☐☐
   ┌ EQ ............ Operand 1 = Operand 2
   │ NE ............ Operand 1 ≠ Operand 2
 ──┤ GT ............ Operand 1 > Operand 2
   │ GE ............ Operand 1 ≥ Operand 2
   │ LT ............ Operand 1 < Operand 2
   └ LE ............ Operand 1 ≤ Operand 2
```

[Example 1]

```
┌─ 600  IFEQ   1    1    Select an axis.
│  ┌─   IFGE   2    0    Select a moving direction.
│  │    JFWN   01   5    Move axis 1 forward.
│  ├─   ELSE
│  │    JBWN   01   5    Move axis 1 backward.
│  └─   EDIF
├─────  ELSE
│  ┌─   IFLT   2    0    Select a moving direction.
│  │    JBWN   10   5    Move axis 2 backward.
│  ├─   ELSE
│  │    JFWN   10   5    Move axis 2 forward.
│  └─   EDIF
└─      EDIF
```

Jog by selecting axis 1/axis 2 by variable 1 and forward/backward (+/–) by variable 2.
Nothing will happen if flag 600 is OFF, in which case the program will proceed to the step next to the last EDIF.

(Note) Using a GOTO command to branch out of or into an IF☐☐-EDIF syntax is prohibited.

- IS□□ (Compare strings)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	IS□□	Column number	Column number, character literal	CP

[Function] Compare the character strings in the columns specified in operands 1 and 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.
If the input condition is not satisfied and the IS□□ command is not executed, the program will proceed to the step next to the EDIF.
A maximum of 15 nests are supported when IF□□ and DW□□ are combined.

```
IS□□
 ├─ EQ  .............. Operand 1 = Operand 2
 └─ NE  .............. Operand 1 ≠ Operand 2
```

[Example 1]

```
        SCPY  10   'GOFD' (Move
                   forward)
        SCPY  14   'GOBK' (Move
                   backward)
        LET   1    5
        LET   2    14
        SLEN  4                      Set the number of comparing characters to 4.
  600   ISEQ  1    '1AXS' (Axis 1)   Select an axis.
        ISEQ  5    10                Select a moving direction.
        JFWN  01   5                 Move axis 1 forward.
        ELSE
        JBWN  01   5                 Move axis 1 backward.
        EDIF
        ELSE
        ISNE  *1   *2                Select a moving direction.
        JFWN  10   5                 Move axis 2 backward.
        ELSE
        JBWN  10   5                 Move axis 2 forward.
        EDIF
        EDIF
```

Jog by selecting axis 1/axis 2 by columns 1 to 4 and forward/backward by columns 5 to 8.
Nothing will happen if flag 600 is OFF, in which case the program will proceed to the step next to the last EDIF.
If columns 1 to 8 contain the following data, axis 1 will be moved forward.

1 2	3 4	5 6	7 8	
1A	XS	GO	FD	

(Note) Using a GOTO command to branch out of or into an IS□□-EDIF syntax is prohibited.

- ELSE (Else)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	ELSE	Prohibited	Prohibited	CP

[Function] An ELSE command is used arbitrarily in conjunction with an IF☐☐ or IS☐☐ command to declare the command part to be executed when the condition is not satisfied.

[Example 1] Refer to the sections on IF☐☐ and IS☐☐.

- EDIF (End IF☐☐)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	EDIF	Prohibited	Prohibited	CP

[Function] Declare the end of an IF☐☐ or IS☐☐ command.

[Example 1] Refer to the sections on IF☐☐ and IS☐☐.

1.14 Structural DO

- DW□□ (DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	DW□□	Variable number	Data	CP

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and execute the subsequent commands up to EDDO while the condition is satisfied.
The program will proceed to the step next to the corresponding EDDO if the condition is no longer satisfied.
A LEAV command can be used to forcibly end a loop.
If the input condition is not satisfied and the DW□□ command is not executed, the program will proceed to the step next to the corresponding EDDO.
A maximum of 15 nests are supported when IF□□ and IS□□ are combined.

DW□□
- EQ .... Operand 1 = Operand 2
- NE .... Operand 1 ≠ Operand 2
- GT .... Operand 1 > Operand 2
- GE .... Operand 1 ≥ Operand 2
- LT .... Operand 1 < Operand 2
- LE .... Operand 1 ≤ Operand 2

[Example 1] 008 DWEQ 1 0 Repeat the command up to an EDDO command while variable 1 contains "0."

:
:
EDDO

If DW□□ is specified at the start and input 8 is OFF, nothing will occur and the program will proceed to the step next to EDDO.

(Note) Using a GOTO command to branch out of or into a DW□□-EDDO syntax is prohibited.

- LEAV (Pull out of DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	LEAV	Prohibited	Prohibited	CP

[Function] Pull out of a DO□□ loop and proceed to the step next to EDDO.

[Example 1] DWEQ 1 0 Repeat the commands up to an EDDO command while variable 1 contains '0.'

:
600 LEAV Forcibly end the loop if flag 600 is ON and proceed to the step next to an EDDO command.

:
EDDO

● ITER (Repeat)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ITER	Prohibited	Prohibited	CP

[Function] Forcibly switch the control to EDDO while in a DO☐☐ loop.

[Example 1]

	DWEQ	1	0	Repeat the commands up to an EDDO command while variable 1 contains "0."
	:			
600	ITER			Forcibly switch the control to an EDDO command and perform end judgment, if flag 600 is ON.
	:			
	EDDO			

● EDDO (End DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	EDDO	Prohibited	Prohibited	CP

[Function] Declare the end of a loop that began with DW☐☐.
If the DW☐☐ condition is not satisfied, the program will proceed to the step next to this command.

[Example 1] Refer to the section on DW☐☐.

1.15 Multi-Branching

- SLCT (Start selected group)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SLCT	Prohibited	Prohibited	CP

[Function] Branch to the step next to any WH☐☐ or WS☐☐ command that exists before an EDSL command and whose condition is satisfied, or to the step next to an OTHE command if none of the conditions are satisfied.
A SLCT command must be followed by a WH☐☐, WS☐☐ or EDSL command.
A maximum of 15 nests are supported.

(Note) Using a GOTO command to branch out of or into a SLCT-EDSL syntax is prohibited.

[Example 1]

	SCPY	1	'Right'	Assign 'right' to columns 1 and 2.
	:			
600	SLCT			Jump to a WXXX whose condition is satisfied.
	WSEQ	1	'Right'	If 'right' is stored in columns 1 and 2, this command will be executed.
	:			
	WSEQ	1	'Left'	If 'left' is stored, this command will be executed.
	:			
	OTHE			If the content of columns 1 and 2 is neither of the above, this command will be executed.
	:			
	EDSL			If flag 600 is OFF, the processing will move here upon execution of any of the conditions.

- WH□□ (Select if true; variable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	WH□□	Variable number	Data	CP

[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W□□□ command or an OTHE or EDSL command when the comparison result of the content of the variable specified in operand 1 with the value specified in operand 2 satisfies the condition.

```
WH□□
    EQ  ........... Operand 1 = Operand 2
    NE  ........... Operand 1 ≠ Operand 2
    GT  ........... Operand 1 > Operand 2
    GE  ........... Operand 1 ≥ Operand 2
    LT  ........... Operand 1 < Operand 2
    LE  ........... Operand 1 ≤ Operand 2
```

[Example 1]

LET	1	20	Assign 20 to variable 1.
LET	2	10	Assign 10 to variable 2.
:			
SLCT			Execute multi-branching.
WHEQ	1	10	(1) will be executed if the content of variable 1 is 10.
:			Since variable 1 contains 20, however, the next
(1)			condition will be referenced.
:			
WHGT	1	*2	This command will be executed if the content of variable
:			1 is greater than the content of variable 2.
(2)			Since variable 1 (= 20) > variable 2 (=10), (2) will be
			executed.
OTHE			This command will be executed if none of the conditions
:			are satisfied. In this example, since (2) was executed,
(3)			(3) will not be executed.
:			The processing will move here if any of the conditions
EDSL			were satisfied and the applicable command executed. In
:			this example, (2) and (4) will be executed.
(4)			
:			

* If multiple conditions are likely to be satisfied, remember that the first W□□□ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

- WS□□ (Select if true; character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	WS□□	Column number	Column number, character literal	CP

[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W□□□ command or an OTHE or EDSL command when the comparison result of the character strings in the columns specified in operands 1 and 2 satisfies the condition.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

```
WS□□
 ├─ EQ  .............. Operand 1 = Operand 2
 └─ NE  .............. Operand 1 ≠ Operand 2
```

[Example 1]

```
SLEN   3              Set the number of comparing characters to 3.
SCPY   1     'ABC'    Assign 'ABC' to column 1.
LET    1     2        Assign 2 to variable 1.
 :
SLCT                  Execute multi-branching.
WSEQ   1     'XYZ'    (1) will be executed if columns 1 to 3 contain 'XYZ.'
 :                    Since columns 1 to 3 contain 'ABC,' however, this
(1)                   command will not be executed.
 :
WSEQ   2     *1       (2) will be executed if the content of the number of
 :                    characters specified by SLEN after column 2 is the
(2)                   same as the content of the column specified in variable
 :                    1.
OTHE                  This command will be executed if none of the conditions
 :                    are satisfied. In this example, since (2) was executed,
(3)                   (3) will not be executed.
 :
EDSL                  The processing will move here if any of the conditions
 :                    were satisfied and the applicable command executed. In
(4)                   this example, (2) and (4) will be executed.
 :
```

* If multiple conditions are likely to be satisfied, remember that the first W□□□ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

- OTHE (Select other)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	OTHE	Prohibited	Prohibited	CP

[Function] This command is used between SLCT and EDSL commands to declare the command to be executed when none of the conditions are satisfied.

[Example 1] Refer to the sections on SLCT, WH□□ and WS□□.

- EDSL (End selected group)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	EDSL	Prohibited	Prohibited	CP

[Function] Declare the end of a SLCT command.

[Example 1] Refer to the sections on SLCT, WH□□ and WS□□.

1.16 System Information Acquisition

- AXST (Get axis status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	AXST	Variable number	Axis number	CP

[Function] Store in the variable specified in operand 1 the status (axis error number) of the axis specified in operand 2.

(Note 1) If the obtained result is “0,” it means no axis error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.

[Example] AXST 1 2 Read the error number for axis 2 to variable 1.

If 3188 (decimal) is stored in variable 1 after the execution of this command:

3188 ÷ 16 = 199 ,,,4
199 ÷ 16 = 12 (= C) ,,,7

$3188 = 12\ (= C) \times 16^2 + 7 \times 16^2 + 4$
$= C74$ (HEX) (Hexadecimal number)

Therefore, an “Error No. C74, Actual-position soft limit over error” is present.

- PGST (Get program status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PGST	Variable number	Program number	CP

[Function] Store in the variable specified in operand 1 the status (program error number) of the program specified in operand 2.

(Note 1) If the obtained result is "0," it means no program error is present.
(Note 2) Although the error lists are written in hexadecimals, the status to be stored (program error number) is a decimal. Therefore, the decimal program error numbers must be converted to hexadecimals.

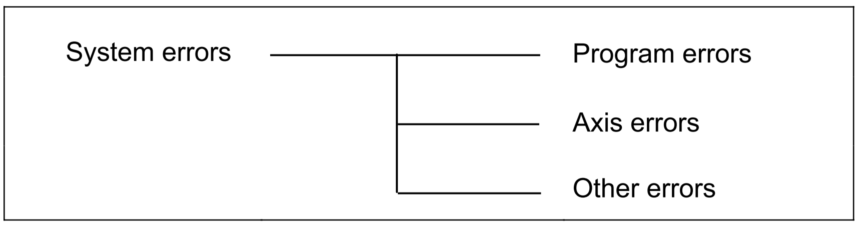
[Example] PGST 1 2 Read the error number for program No. 2 to variable 1.

- SYST (Get system status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SYST	Variable number	Prohibited	CP

[Function] Store the system status (top-priority system error number) in the variable specified in operand 1.

(Note 1) If the obtained result is “0,” it means no system error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.
(Note 3) Relationship of error statuses



* An axis error that generates during operation with a program command will be registered both as a program error and an axis error.

[Example] SYST 1 Read the system error number to variable 1.

1.17 Zone

- WZNA (Wait for zone ON, with AND)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WZNA	Zone number	Axis pattern	CP

[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter Nos. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter Nos. 88, 91, 94 and 97" irrespective of this command.

[Example 1]

```
WZNA    1    11
```

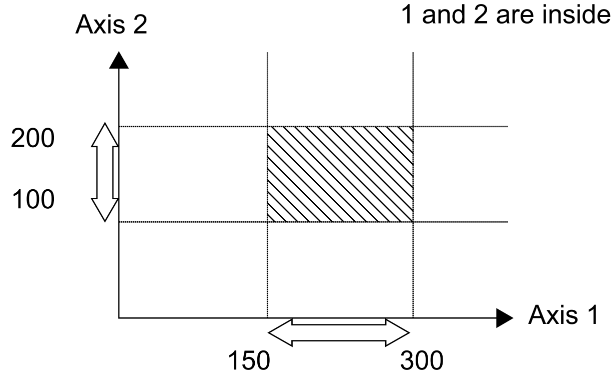
If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     5    3     Assign 3 to variable 5.
WZNA    1    *5
```

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001 mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001 mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.



- WZNO (Wait for zone ON, with OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WZNO	Zone number	Axis pattern	CP

[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter Nos. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter Nos. 88, 91, 94 and 97" irrespective of this command.

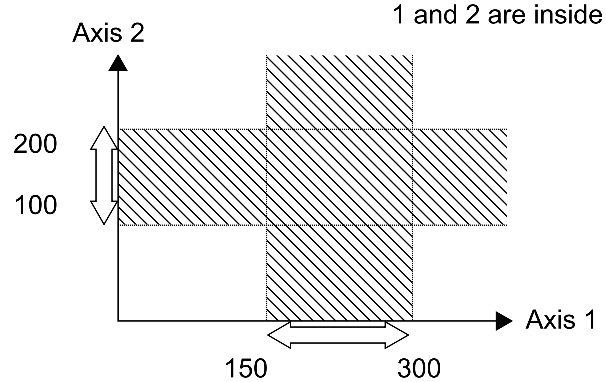
[Example 1] WZNO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3     Assign 3 to variable 5.
WZNO   1    *5
```

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001 mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001 mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.



- WZFA (Wait for zone OFF, with AND)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WZFA	Zone number	Axis pattern	CP

[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter Nos. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter Nos. 88, 91, 94 and 97" irrespective of this command.

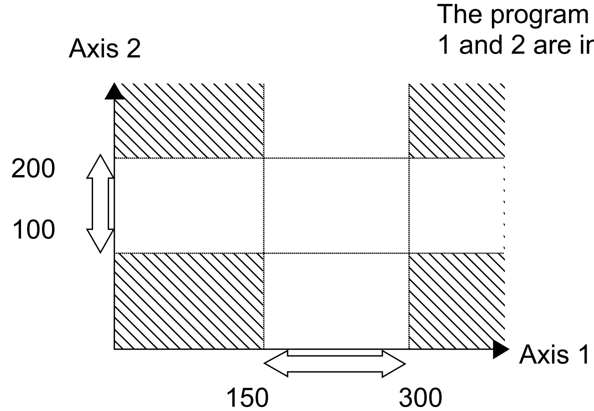
[Example] WZFA 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes OFF (inside the shaded area shown in the diagram below)

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

LET	5	3	Assign 3 to variable 5.
WZFA	1	*5	

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001 mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001 mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.



- WZFO (Wait for zone OFF, with OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WZFO	Zone number	Axis pattern	CP

[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter Nos. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter Nos. 88, 91, 94 and 97" irrespective of this command.

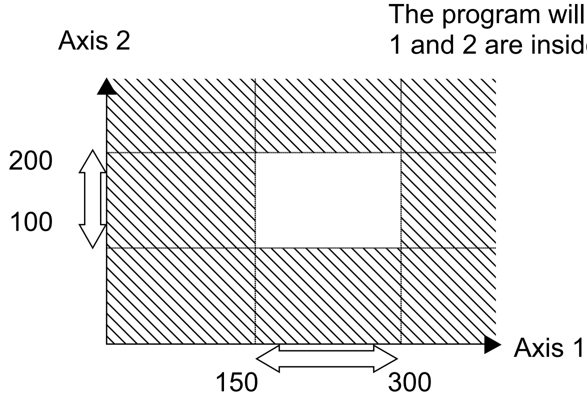
[Example 1] WZFO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes OFF (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     5     3       Assign 3 to variable 5.
WZFO    1     *5
```

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001 mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001 mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.



1.18 Communication

- OPEN (Open channel)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OPEN	Channel number	Prohibited	CP

[Function] Open the channel specified in operand 1.
The specified channel will be enabled to send/receive hereafter.
Prior to executing this command, a SCHA command must be used to set an end character.

[Example]

```
SCHA    10
OPEN    0
        Specify 10 (= LF) as the end character.
        Open channel 0.
```

> Note: If "OPEN 0" is executed, communication with the teaching pendant or PC software will be cut off.

- CLOS (Close channel)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CLOS	Channel number	Prohibited	CP

[Function] Close the channel specified in operand 1.
The specified channel will be disabled to send/receive hereafter.

[Example]

```
CLOS    0
Close channel 0.

LET     1       0
CLOS    *1
        Assign 0 to variable 1.
        Close the content of variable 1 (channel 0).
```

- READ (Read)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	READ	Channel number	Column number	CC

[Function] Read a character string from the channel specified in operand 1 to the column specified in operand 2.
Read will end when the character specified by a SCHA command is received.
Either a local or global column may be specified.
A return code will be stored in a local variable (variable 99 under the factory setting) immediately after this command is executed.
Whether or not the command has been executed successfully can be checked based on this return code. Define appropriate processing to handle situations where the command execution failed due to an error.
Setting “0” in operand 2 will specify a dummy read (receive buffer cleared and receive disabled) (the return code will indicate that the command was successfully executed).

[Example]

SCHA	10		Set LF (= 10) as the end character.
OPEN	0		Open channel 0.
READ	0	2	Read a character string from channel 0 to column 2 until LF is received.
TRAN	1	99	Assign the return code (variable 99) to variable 1.
CLOS	0		Close the channel.
SLCT			The processing flow branches out in accordance with each return code. (Note) Using a GOTO command to branch out of a BGPA-EDPA syntax or to other branch processing within the syntax is prohibited.
WHEQ : (1) :	1	0	If the content of variable 1 is “0” (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution.
WHEQ : (2) :	1	1	If the content of variable 1 is “1” (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary.
WHEQ : (3) :	1	2	If the content of variable 1 is “2” (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary.
OTHE : (4) :			If the content of variable 1 is not “0,” “1” or “2,” (4) will be executed. In (4), define appropriate error handling, if necessary.
EDSL			Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here.

(Note) A READ command must be executed before the other side sends the end character.

```
SCHA    10
OPEN    0
READ    0        2
   <---------------------- Other side
CLOS    0
```

- Return code of the READ command

  The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24.” The default variable number is 99.

  0: READ completed successfully (Receive complete)
  1: READ timeout (the timeout value is set by a TMRD command) (Continue to receive)
  2: READ timer cancelled (the wait status is cancelled by a TIMC command) (Continue to receive)
  3: READ SCIF overrun error (Receive disabled)
  4: READ SCIF receive error (framing error or parity error) (Receive disabled)
  5: READ factor error (program abort error) (Receive disabled)
  (Cannot be recognized by SEL commands)
  6: READ task ended (program end request, etc.) (Receive disabled)
  (Cannot be recognized by SEL commands)
  7: READ SCIF receive error due to other factor (Receive disabled)
  8: READ SIO overrun error (Receive disabled)
  9: READ SIO parity error (Receive disabled)
  10: READ SIO framing error (Receive disabled)
  11: READ SIO buffer overflow error (Receive disabled)
  12: READ SIO receive error due to other factor (Receive disabled)
  13 ~ 20: Used only in Ethernet (optional)
  21: READ SIO receive temporary queue overflow error (Receive disabled)
  22: READ SIO slave receive queue overflow error (Receive disabled)

- TMRW (Set READ/WRIT timeout value)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	TMRW	Read timer setting	(Write timer setting)	CP

[Function] Set the timeout to be applied to a READ/WRIT command.
With the ASEL controller, a write timer setting cannot be specified.
The timer setting specified in operand 1 will set the maximum time the program will wait for the character string read to end when a READ command is executed.
If the end character could not be read before the timer is up during the execution of the READ command, a timeout will occur and the program will move to the next step.
(Whether or not a timeout has occurred can be checked from the return code that will be stored in variable 99 (factory setting) immediately after the READ command is executed. If necessary, define appropriate processing to handle a timeout.)
Setting the timer to "0" will allow the READ command to wait infinitely, without timeout, until the end character is read.
The timer setting is input in seconds (setting range: 0 to 99.00 seconds) including up to two decimal places.
A variable can be specified indirectly in operand 1.

(Note) TMRW is set to "0" in the default condition before TMRW setting is performed.

[Example]

SCHA	10		Set LF (=10) as the end character.
TMRW	30		Set the READ timeout value to 30 seconds.
OPEN	0		Open channel 0.
READ	0	2	Read the character string from channel 0 to column 2 until LF is read.
TRAN	1	99	Assign the return code to variable 1.
CLOS	0		Close the channel.
SLCT			The processing flow branches out in accordance with each return code. (Note) Using a GOTO command to branch out of a BGPA-EDPA syntax or to other branch processing within the syntax is prohibited.
WHEQ : (1) :	1	0	If the content of variable 1 is "0" (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution.
WHEQ : (2) :	1	1	If the content of variable 1 is "1" (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary.
WHEQ : (3) :	1	2	If the content of variable 1 is "2" (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary.
OTHE : (4) :			If the content of variable 1 is not "0," "1" or "2," (4) will be executed. In (4), define appropriate error handling, if necessary.
EDSL			Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here.

Read completes successfully within 30 seconds → Variable No. 1 = 0
Timeout occurs → Variable No. 1 = 1
* The return code of READ command may not be limited to 0 or 1. The variable to store the return code can be set in “Other parameter No. 24.” Refer to the explanation of READ command for details.

- WRIT (Write)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	WRIT	Channel number	Column number	CC (Note 1)

[Function] Write the character string in the column specified in operand 2 to the channel specified in operand 1.
The operation will end when the character specified by a SCHA command is written.
Either a local or global column can be specified.

[Example]

SCHA	10		Set LF (= 10) as the end character.
OPEN	0		Open channel 0.
WRIT	0	2	Write the character string in column 2 to channel 0 until LF is written.
CLOS	0		Close the channel.

Once the channel has been opened, a WRIT command can be executed (data can be sent) for other tasks besides the one that opened the channel. Accordingly, if a READ command is executed for a channel-opening task and then a WRIT command is executed for other task, the response from the other side can be received without delay after the applicable data is sent from the PSEL.

The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24.” The default variable number is 99.
0: WRIT completed successfully
1: WRIT timeout (the timeout value is set by a TMRW command)
2: WRIT timer cancelled (the wait status is cancelled by a TIMC command)
3 ~ 4: For future expansion
5: WRIT factor error (program abort error) (Cannot be recognized by SEL commands)
6: WRIT task ended (program end request, etc.) (Cannot be recognized by SEL commands)

- SCHA (Set end character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SCHA	Character code	Prohibited	CP

[Function] Set the end character to be used by a READ or WRIT command.
Any character from 0 to 255 (character code used in BASIC, etc.) can be specified.

[Example] Refer to the sections on READ and WRIT commands.

1.19 String Operation

- SCPY (Copy character string)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SCPY	Column number	Column number, character literal	CC

[Function] Copy the character string in the column specified in operand 2 to the column specified in operand 1.
Copy will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, copy will be performed for the entire length of the literal.

[Example]

```
SCPY   1     'ABC'   Copy 'ABC' to column 1.

SLEN   10            Set the copying length to 10 bytes.
SCPY   100   200     Copy 10 bytes from column 200 to column 100.
```

● SCMP (Compare character strings)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SCMP	Column number	Column number, character literal	EQ

[Function] Compare the column specified in operand 1 with the column specified in operand 2.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

[Example]	SCMP	1	'ABC'	600	Flag 600 will turn ON if columns 1 to 3 contain 'ABC.'
	SLEN	5			Set the comparing length to five bytes.
	SCMP	10	30	999	Turn ON flag 999 if five bytes from columns 30 and 10 match.

- SGET (Get character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SGET	Variable number	Column number, character literal	CP

[Function] Assign one character from the column specified in operand 2 to the variable specified in operand 1.
If a character-string literal is specified in operand 2, the first character will be assigned.

[Example]

SGET	1	100	
Assign one byte from column 100 to variable 1.			
LET	1	3	Assign 3 to variable 1.
LET	2	1	Assign 1 to variable 2.
SCPY	1	‘A’	Copy ‘A’ to column 1.
SGET	*1	*2	Assign ‘A’ from the content of variable 2 (column 1) to the content of variable 1 (variable 3).

- SPUT (Set character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SPUT	Column number	Data	CP

[Function] Set the data specified in operand 2 in the column specified in operand 1.

[Example]	SPUT	5	10	Set 10 (LF) in column 5.
	LET	1	100	Assign 100 to variable 1.
	LET	2	50	Assign 50 to variable 2.
	SPUT	*1	*2	Set the content of variable 2 (50 (‘2’)) in the content of variable 1 (column 100).

- STR (Convert character string; decimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	STR	Column number	Data	CC

[Function] Copy to the column specified in operand 1 a decimal character string converted from the data specified in operand 2.
The data will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a 10-digit integer including eight or more valid digits, conversion of the values in the eighth and subsequent digits will not be guaranteed (the values through the seventh digits will be converted properly.)

[Example]

SLEN	5.3		Set a length consisting of five integer digits and three decimal digits.
STR	1	123	The following values will be set in columns 1 to 9:

1	2	3	4	5	6	7	8	9
		1	2	3	.	0	0	0

LET	1	10	Assign 10 to variable 1.
LET	102	987.6543	Assign 987.6543 to variable 102.
SLEN	2.3		Set a length consisting of two integer digits and three decimal digits.
STR	*1	*102	The following values will be set in columns 10 to 15:

10	11	12	13	14	15
8	7	.	6	5	4

Since the data exceeds the specified length, “9” in the 100’s place and “3” in the fourth decimal place will be cut off.

- STRH (Convert character string; hexadecimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	STRH	Column number	Data	CC

[Function] Copy to the column specified in operand 1 a hexadecimal character string converted from the data specified in operand 2.
Only the integer part will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a negative value, eight columns will be required to covert the entire data.

[Example]

SLEN	5		Set a format consisting of five integer digits.
STRH	1	255	The following values will be set in columns 1 to 5:

1	2	3	4	5
			F	F

LET	1	10	Assign 10 to variable 1.
LET	102	987.6543	Assign 987.6543 to variable 102.
SLEN	2.3		Set a length consisting of two integer digits and three decimal digits.
STRH	*1	*102	The following values will be set in columns 10 and 11:

10	11
D	B

".3" in the SLEN command and ".6543" in variable 102, which are the decimal part, will be ignored.
The integer part is expressed as '3DB' in hexadecimal.
Since the length is two digits, however, "3" in the third digit will be cut off.

- VAL (Convert character string data; decimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	VAL	Variable number	Column number, character literal	CC

[Function] Convert the decimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 18 characters or less.

[Example]

SCPY	10	'1234'	Set '1234' in column 10.
SLEN	4		Set the converting length to four bytes.
VAL	1	10	Assign 1234, which is a binary converted from '1234' in column 10, to variable 1.

LET	1	100	Assign 100 to variable 1.
LET	2	20	Assign 20 to variable 2.
SCPY	20	'1234'	Copy '1234' to column 20.
SCPY	24	'.567'	Copy '.567' to column 24.
SLEN	8		Set the converting length to eight bytes.
VAL	*1	*2	Assign 1234.567, which is a binary converted from '1234.567' in the content of variable 2 (column 20) to the content of variable 1 (variable 100).

- VALH (Convert character string data; hexadecimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	VALH	Variable number	Column number, character literal	CC

[Function] Convert the hexadecimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
Only the integer part will be converted, with the decimal part being ignored.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 8 characters or less.

[Example]

SCPY	10	‘1234’	Set ‘1234’ in column 10.
SLEN	4		Set the converting length to four bytes.
VALH	1	10	Assign 4660, which is a binary converted from hexadecimal ‘1234’ in column 10, to variable 1.

LET	1	100	Assign 100 to variable 1.
LET	2	20	Assign 20 to variable 2.
SCPY	20	‘ABCD’	Copy ‘ABCD’ to column 20.
SLEN	4		Set the converting length to four bytes.
VALH	*1	*2	Assign 43981, which is a binary converted from hexadecimal ‘ABCD’ in the content of variable 2 (column 20) to the content of variable 1 (variable 100).

- SLEN (Set length)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	SLEN	Character string length	Prohibited	CP

[Function] Set the length to be processed by a string command.
This must always be set before using the following commands:

SCMP	....	Decimal part is invalid.
SCPY	....	Decimal part is invalid.
ISXX	....	Decimal part is invalid.
WSXX	....	Decimal part is invalid.
STRH	....	Decimal part is invalid.
VAL, VALH	....	Decimal part is invalid.
STR	....	Decimal part is valid.

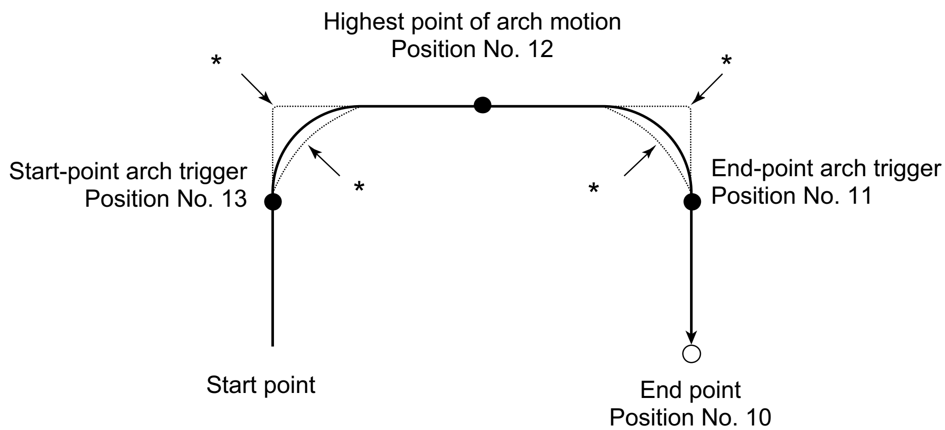
[Example] Refer to the examples of the above commands:

1.20 Arch-Motion-Related

- ARCH (Arch motion)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ARCH	Position number	Position number	PE

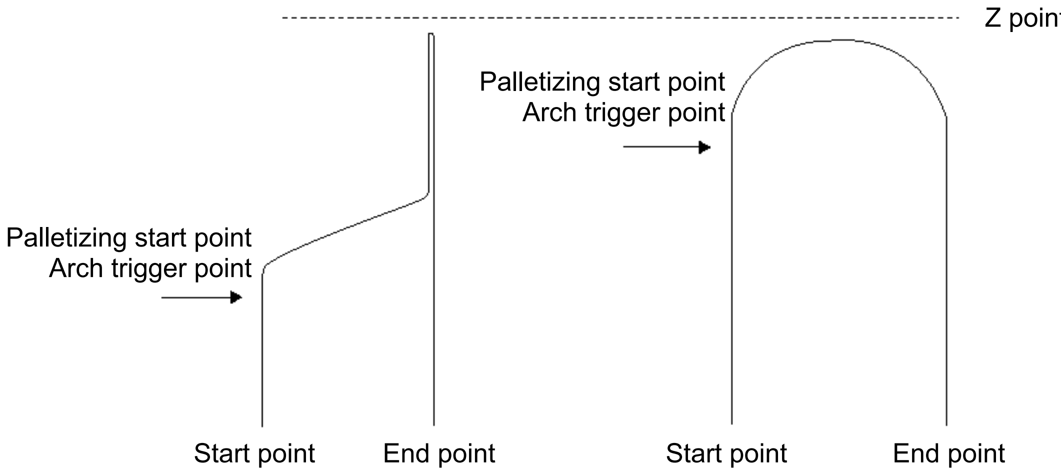
Perform arch motion from the current point and move to the specified points.
- Move to the points specified in operand 1, via arch motion.
- Movements in directions other than the arch-motion Z-axis direction will begin after rising from the current point to the start-point arch trigger. After the Z point specified in operand 2 (as the highest point) is passed and movements in directions other than the arch-motion Z-axis direction are complete, the axes will come down to the end-point arch trigger and reach the specified point.
- Palletizing arch triggers must be set using an ATRG command.

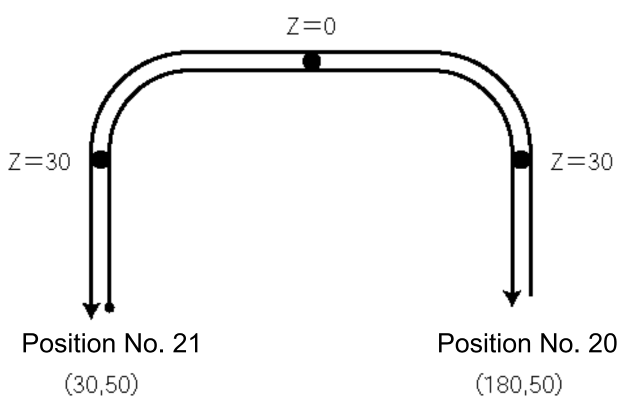



```
ACHZ   2
ATRG   13    11
  |
  |
  |
ARCH   10    12
```

* When the operation is resumed after a pause, depending on the position where the operation is resumed the locus may follow the lines (dotted lines) indicated by asterisks in the diagram for the composite section from ascent to horizontal movement or from horizontal movement to descent. Be careful not to cause interference.
- The arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis component of the point data specified in operand 1, if any, plus the arch-motion Z-axis offset. If there is no arch-motion Z component, the arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis coordinate of the start point plus the arch-motion Z-axis offset. (Normally the offset is added to all arch-motion positions, such as the arch triggers and Z point.)
- An error will generate if the start-point arch trigger is set below the start point or the end-point arch trigger is set below the end point. (Note: Up/down has nothing to do with +/– on the coordinate system.)
- The arch-motion Z-axis up direction refers to the direction toward the Z point from the start point (the down direction refers to the opposite direction), and has nothing to do with the size of coordinate value. Therefore, be sure to confirm the actual operating direction when using this command.

- The arch-motion Z-axis will come down after a rise-process command value is output. Therefore, one of the following operations will be performed depending on how the arch-trigger point and Z point are set.
  If the resulting operation is undesirable, change the arch trigger and/or Z point to improve the efficiency of movement.





The table below shows a program and data to cause the actuator to perform arch-motion operation by moving back and forth along the above path.

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					VEL	200			速度200mm/sec
2					ACHZ	2			ｱｰﾁﾓｰｼｮﾝZ軸=2軸
3					ATRG	22	22		ｱｰﾁﾄﾘｶﾞﾎﾟｼﾞｼｮﾝNo23
4					MOVP	20			ﾎﾟｼﾞｼｮﾝ20ﾍPTP移動
5					TAG	1			
6					ARCH	21	23		ﾎﾟｼﾞｼｮﾝ21ﾍｱｰﾁﾓｰｼｮﾝ
7					ARCH	20	23		ﾎﾟｼﾞｼｮﾝ20ﾍｱｰﾁﾓｰｼｮﾝ
8					GOTO	1			

No.	Axis1	Axis2	Vel	Acc	Dcl
20	180.000	50.000			
21	30.000	50.000			
22		30.000			
23		0.000			

- ACHZ (Declare arch-motion Z-axis)

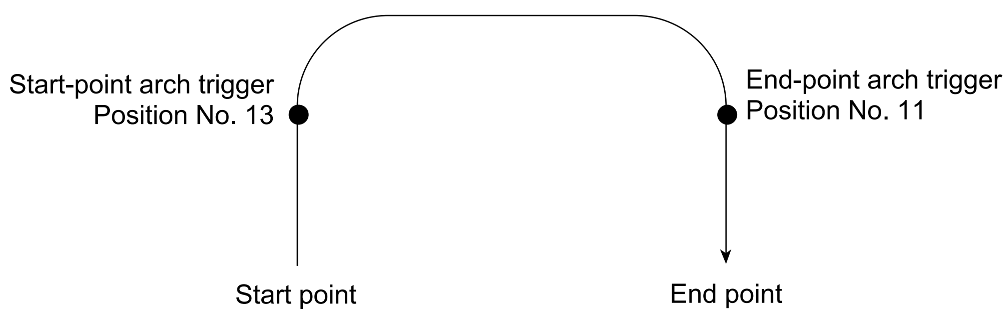
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ACHZ	Axis number	Prohibited	CP

Specify the axis number representing the arch-motion Z direction.
The axis number specified in operand 1 will be set as the axis number representing the arch-motion Z direction.
If the output field is specified, the output will turn ON after this command is executed.

- ATRG (Set arch triggers)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ATRG	Position number	Position number	CP

Set the arch triggers used for arch motion.
(This setting becomes valid when an ARCH command is executed.)
Set the arch-motion Z-axis position data in the point data specified in operand 1 as the start-point arch trigger, and set the arch-motion Z-axis position data in the point data specified in operand 2 as the end-point arch trigger.



ATRG 13 11

(Refer to "Palletizing Setting" – "Arch triggers" under "How to Use.")
For an arch-motion operation, set it so that a horizontal movement will begin when the start-point arch trigger is reached during ascent from the start point, and that the end-point arch trigger will be reached after a horizontal movement is completed during descent.
If the output field is specified, the output will turn ON after this command is executed.

- OFAZ (Set arch-motion Z-axis offset)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	OFAZ	Offset value	Prohibited	CP

Set the offset in the arch-motion Z-axis direction.
The value specified in operand 1 will be set as the offset in the arch-motion Z-axis direction.
The offset amount is set in mm and the effective resolution is 0.001 mm.
A negative value can also be specified as the offset, as long as the operation range will not be exceeded.
This offset is valid only at the end point of ARCH (arch motion) operation.
If the output field is specified, the output will turn ON after this command is executed.

1.21 Palletizing-Related

- BGPA (Declare start of palletizing setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	BGPA	Palletizing number	Prohibited	CP

Declare the start of a palletizing setting.
Once this command is executed, palletizing setting for the palletizing number specified in operand 1 will be enabled.
(In the case of an ACHZ, AEXT, OFAZ or ATRG command, setting is enabled without declaring BGPA.)
The input range of palletizing number is from 1 to 10.
When the palletizing setting is complete, execute EDPA.
Nested BGPAs are not supported. To declare start of another palletizing setting, execute an EDPA command and then execute a BGPA command again.
If the output field is specified, the output will turn ON after this command is executed.
Palletizing numbers are in the local range. Therefore, a given palletizing setting is valid only within the program in which it is set.

(Note) Using a GOTO command to branch out of or into a BGPA-EDPA syntax is prohibited.

- EDPA (Declare end of palletizing setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	EDPA	Prohibited	Prohibited	CP

Declare the end of a palletizing setting.
If a palletizing-setting command (excluding BGPA, ACHZ, ATRG, AEXT and OFAZ) is executed before another BGPA is declared following an execution of this command (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PAPI (Set palletizing counts)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPI	Count	Count	CP

Set counts in the palletizing-axis directions.
The count specified in operand 1 will apply to the preferential-axis (PX-axis) direction, while the count specified in operand 2 will apply to the PY-axis direction.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PAPN (Set palletizing pattern)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPN	Pattern number	Prohibited	CP

Set a palletizing pattern.
The palletizing pattern specified in operand 1 will be set (1 = Pattern 1, 2 = Pattern 2).
If this command is not declared, pattern 1 will be used.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PASE (Declare palletizing axes)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PASE	Axis number	Axis number	CP

Set the two axes to be used in palletizing (PX and PY-axes).
The axis specified in operand 1 will be set as the preferential axis (PX-axis).
The axis specified in operand 2 will be set as the PY-axis.
This command is used in conjunction with PAPT and PAST.
It cannot be used together with a 3-point teaching (PAPS) command. Whichever is set later will be given priority.

It is recommended to use a 3 or 4-points teaching (PAPS) command if the palletizing requires high accuracy.

If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PAPT (Set palletizing pitches)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPT	Pitch	Pitch	CP

Set palletizing pitches.
The value specified in operand 1 will be set as the pitch for the preferential axis (PX-axis), while the value specified in operand 2 will be set as the pitch for the PY-axis.
This command is used in conjunction with PASE and PAST.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PAST (Set palletizing reference point)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAST	(Position number)	Prohibited	CP

Set the reference point used in palletizing.
If a value is set in operand 1, that position number specified in operand 1 will be used to store the reference point data.
If no value is set in operand 1, the position-number setting for storing reference point data will become invalid.
This command is used in conjunction with PASE and PAPT.
If this command is not set, coordinates (0, 0) are used as the reference point. If this command is set, the set coordinates are used as the reference point in calculating the position coordinates of palletizing points.
Coordinates in both the PX and PY-axis directions must always be set as the reference-point coordinates.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

- PAPS (Set palletizing points) For 3-point teaching

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPS	Position number	(Palletizing position setting type)	CP

Set palletizing positions in 3-point teaching.

It can also be used to set palletizing positions in 4-point teaching, in which case the pallet plane can be set to any quadrilateral other than a square, rectangle or parallelogram.

In operand 1, set the position number of the start point needed to set palletizing positions in 3-point teaching. If “n” is set as the position number for the start point, position data for the end point in the PX-axis direction will be stored in position No. n+1, while position data for the end point in the PY-axis direction will be stored in position No. n+2.

In the case of 4-point teaching, position data for the end point should be stored in position No. n+3.

In operand 2, specify the applicable palletizing position setting type.

[Palletizing position setting type]
If operand 2 is “0” or blank, 3-point teaching will be specified.
As shown in Fig. 1 (a), palletizing positions will be set on the quadrilateral pallet plane determined by the three points including the start point, end point in the PX-axis direction and end point in the PY-axis direction.

If operand 2 is “2,” 4-point teaching will be specified.
As shown in Fig. 1 (b), palletizing positions will be set on the quadrilateral pallet plane determined by the four points including the start point, end point in the PX-axis direction, end point in the PY-axis direction, and end point.

Fig. 1 shows two different arrangements of palletizing positions.

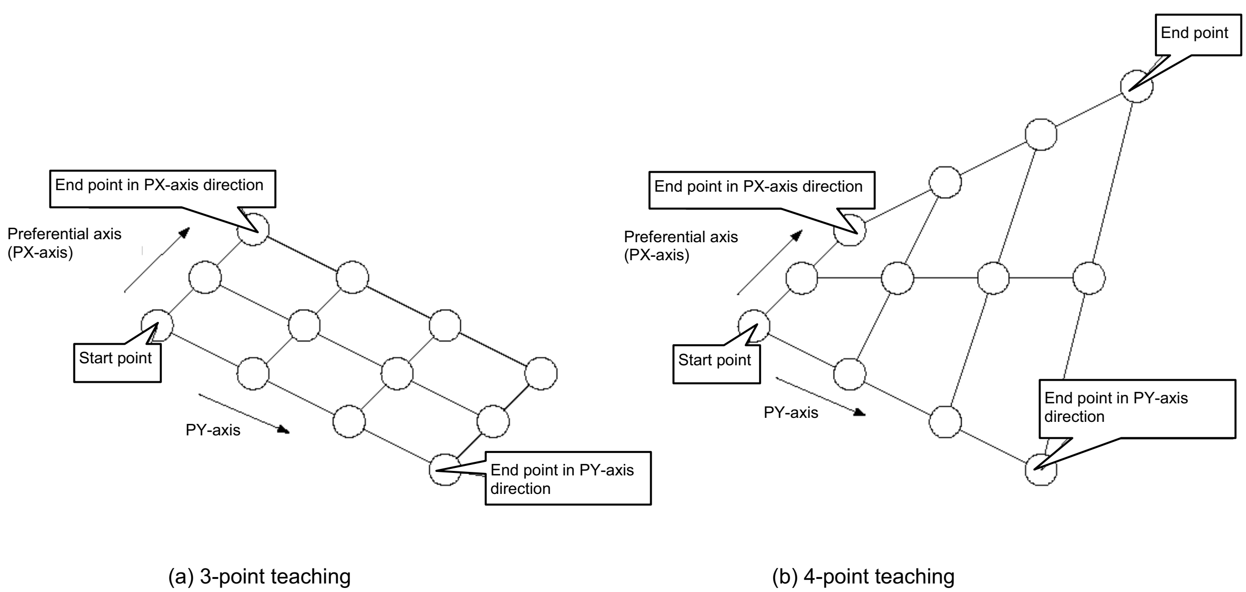


Fig. 1 Layout of Palletizing Positions

- If the valid axis pattern does not match the point data for 3-point teaching or 4-point teaching, an error "CB0, Mismatched valid axes for palletizing 3-point teaching data" will generate. If a PAPS command is executed after specifying the applicable axes using a GRP command, only the point data corresponding to the specified axes, among all axes whose point data is valid, will be used as palletizing point data. Executing a GRP command thereafter with a different setting will have no effect.

- If there are not enough valid axes, an error "CAE, Insufficient valid axes for palletizing 3-point teaching data" will generate.

- This command cannot be used with a PASE (set palletizing axes) command. Whichever was set later will be given priority. (A single PAPS command can substitute a set of PASE, PAPT and PAST commands.)

- If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error, "CB5, BGPA not declared at palletizing setting" will generate.

- If the output field is specified, the output will turn ON after this command is executed.

- PSLI (Set zigzag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PSLI	Offset amount	(Count)	CP

Set a zigzag palletizing.
The value specified in operand 1 will be set as the offset amount for even-numbered rows.
The count specified in operand 2 will be set as the count for even-numbered rows.
(Refer to “Palletizing Setting” – “Zigzag setting” under "How to Use.")
If operand 2 is not specified, the count for even-numbered rows will become the same as the count for odd-numbered rows.
If a setting is performed by 3-point teaching with PAPS (set palletizing points), the PX and PY-axes need not be parallel with the physical axes. In this case, the offset will apply in parallel with the PX-axis. If the offset is a positive value, the absolute value of offset will be applied toward the end-point direction of the PX-axis. If the offset is a negative value, the absolute value will be applied toward the start-point direction.
If this command is executed before a BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

1.22 Palletizing Calculation Command

- PTNG (Get palletizing position number)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PTNG	Palletizing number	Variable number	CP

Assign the palletizing position number for the palletizing number specified in operand 1 to the variable specified in operand 2.
If the output field is specified, the output will turn ON after this command is executed.

- PINC (Increment palletizing position number by 1)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PINC	Palletizing number	Prohibited	CC

Increment by 1 the palletizing position number for the palletizing number specified in operand 1.
If the incremented value is considered normal as a palletizing position number calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully incremented, and turn OFF if the increment failed.

- PDEC (Decrement palletizing position number by 1)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PDEC	Palletizing number	Prohibited	CC

Decrement by 1 the palletizing position number for the palletizing number specified in operand 1.
If the decremented value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully decremented, and turn OFF if the decrement failed.

- PSET (Set palletizing position number directly)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PSET	Palletizing number	Data	CC

Set the value specified in operand 2 as the palletizing position number for the palletizing number specified in operand 1.
If the specified value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be set. If not, the value will not be set.
If the output field is specified, the output will turn ON when the palletizing position number was successfully updated, and turn OFF if the update failed.

- PARG (Get palletizing angle)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PARG	Palletizing number	Axis number	CP

Obtain the palletizing angle.
Calculate the palletizing angle (degrees) from the physical axis specified in operand 2 for the palletizing number specified in operand 1, and store the result in variable 199.

This command need not be executed, if not necessary.
If this command is executed after PAPS (set 3 palletizing points for teaching) is executed, the angle formed by the preferential axis and the specified physical axis will be calculated automatically. If this command is executed before PAPS is executed, or after both PAPS and PASE are executed in this order, an error will generate.
If point data is not available for two valid axes, an error "CAE, Insufficient valid axes for palletizing 3-point teaching data" will generate.
If the axis corresponding to the axis number in operand 2 does not specify one of the two valid axes associated with the point data, an error "CBA, Reference-axis/PX/PY-axis mismatch error at palletizing angle acquisition" will generate.
If the reference point data is the same as the point data at the PX-axis end point in 3-point teaching, an error "Reference-point/PX-axis end point duplication error at palletizing angle acquisition" will generate, and angle calculation will be disabled.
The actual operating direction may have been reversed depending on the mechanism of the rotating axis and the setting of axis-specific parameter No. 6, "Operating-direction reversing selection." To use the value obtained by this command, be sure to confirm the actual operating direction.
If the output field is specified, the output will turn ON after this command is executed.

- PAPG (Get palletizing calculation data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PAPG	Palletizing number	Position number	CP

Store the position coordinate data of the palletizing points for the palletizing number specified in operand 1, in the position number specified in operand 2.
If the output field is specified, the output will turn ON after this command is executed.

1.23 Palletizing Movement Command

- PMVP (Move to palletizing points via PTP)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PMVP	Palletizing number	Prohibited	PE

Move to the calculated palletizing points via PTP.
The axes will move to the palletizing points specified in operand 1, via PTP.
Executing this command will not increment the palletizing position number by 1.

- PMVL (Move to palletizing points via interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	PMVL	Palletizing number	Prohibited	PE

Move to the calculated palletizing points via interpolation.
The axes will move to the palletizing points specified in operand 1, via interpolation.
Executing this command will not increment the palletizing position number by 1.

1.24 Building of Pseudo-Ladder Task

- CHPR (Change task level)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	CHPR	0 or 1	Prohibited	CP

[Function] Specify “1” (User HIGH) if you wish the target task to be processed before other tasks.
This command can also be used with non-ladder tasks.
Task level change (0: User NORMAL, 1: User HIGH) is not a required component, but specifying User HIGH will require a TSLP command explained below. (Without TSLP, tasks of the User NORMAL level will not be processed.)

- TPCD (Specify processing to be performed when input condition is not specified)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	TPCD	0 or 1	Prohibited	CP

[Function] Specify the processing to be performed when input condition is not specified.
(0: Execute, 1: Follow the input condition in the last executed step)
In a ladder task, always input “1” (Follow the input condition in the last executed step) in operand 1.
In a non-ladder task, always input “0” (Execute). (The default value is “0.”)

- TSLP (Task sleep)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Prohibited	Prohibited	TSLP	Time	Prohibited	CP

[Function] Set the time during which the applicable task will sleep, in order to distribute the processing time to other tasks.
If the task level is set to User HIGH, this command must always be specified.
The applicable task will sleep during the set time.
The time in operand 1 is set in msec.
An appropriate time setting must be examined on the actual system. (Normally approx. 1 to 3 is set.)
(If the ladder statement becomes long, state this command multiple times between steps, as necessary.)
This command can also be used with non-ladder tasks.

1.25 Extended Command

- ECMD1 (Get motor current value (as percentage of rated current))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ECMD	1	Axis number	CC

[Function] Store the motor current value (percentage of the rated current) corresponding to the "axis number" specified in operand 2, in variable 99.

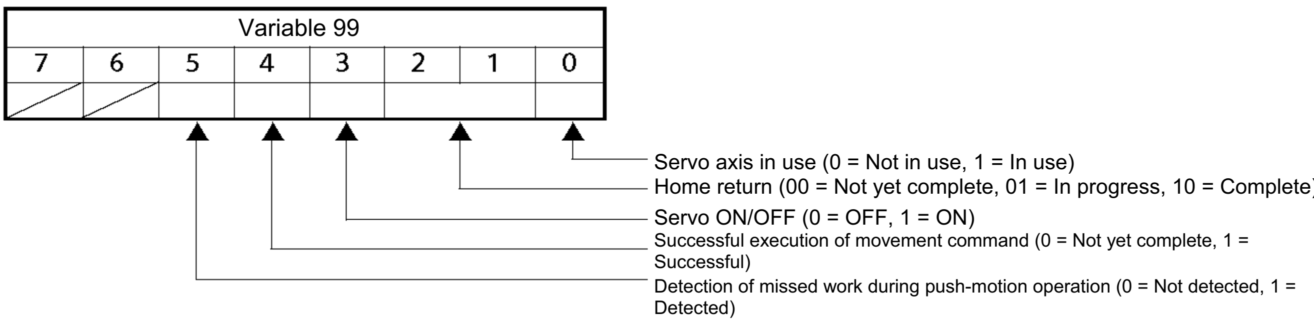
Note: The current value data (percentage of the rated current) obtained by this command has been processed by feedback current filtering and includes analog error.

[Example] ECMD 1 2 Extended command 1
Store the motor current value (percentage of the rated current) of axis 2, in variable 99.

- ECMD5 (Get axis operation status)

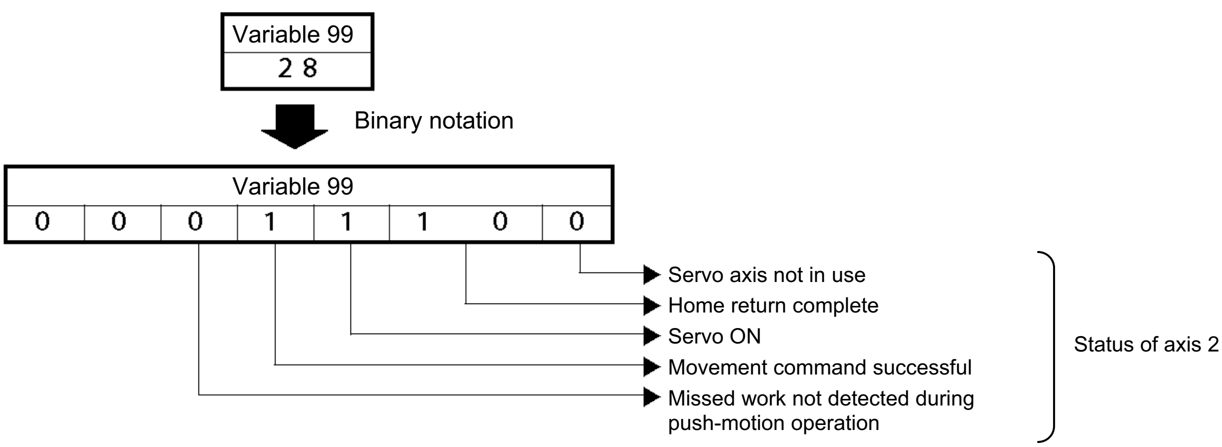
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ECMD	5	Axis number	CC

[Function] Store the status of the axis specified in operand 2, in variable 99.
The axis status is indicated by the ON/OFF level of each bit, as shown below. Accordingly, the obtained value must be converted to a binary value for interpretation.



(Note) If an invalid axis number is specified in operand 2, “C44, SEL data error” will generate.

[Example] ECMD 5 2 Store the status of axis 2 in variable 99. If 28 (decimal value) was stored in variable 99 after the command was executed, the status of axis 2 is interpreted as follows.



- ECMD20 (Get parameter value)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
Optional	Optional	ECMD	20	Variable number	CC

[Function] Store the value of the specified parameter in variable 99, using the data stored in the three consecutive variables starting from the one corresponding to the variable number specified in operand 2.
If variable No. n is set in operand 2, the data in variable No. n will indicate the parameter type, data in variable No. n+1 will indicate the device number (or axis number), and data in variable No. n+2 will indicate the parameter number, respectively. The ranges of parameter type, device number (or axis number) and parameter number are specified below. If an out-of-range value is specified, "C44, SEL data error" will generate.

	I/O	Common to all axes	Axis-specific	Driver	Encoder	I/O device	Other
Parameter type	0	1	2	3	4	5	7
Device number/axis number	0	0	1 ~ 2	1 ~ 2	1 ~ 2	0 ~ 7	0
Parameter number	1 ~ 300	1 ~ 120	1 ~ 200	1 ~ 97	1 ~ 30	1 ~ 82	1 ~ 100

Specify an integer variable in operand 2 (integer variables 98, 99, 298, 299, 1098, 1099, 1298 and 1299 cannot be specified, because three consecutive integer variables cannot be allocated if any of these integer variables is specified). If a variable of non-integer type is specified, "C3C, Variable number error" will generate.

(Note) If an invalid axis number is specified in operand 2, "C44, SEL data error" will generate.

[Example 1]

```
LET    10    2      Variable No. 10 = Parameter type (Axis-specific)
LET    11    2      Variable No. 11 = Axis number (Axis 2)
LET    12    42     Variable No. 12 = Parameter number (No. 42)
ECMD   20    10     Extended command 20 (Use variable Nos. 10 through 12)
                    Store the value of axis-specific parameter No. 42 (axis 2),
                    "Encoder resolution," in variable 99.
```

[Example 2]

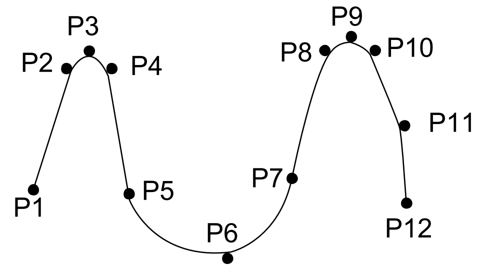
```
LET    1250  0      Variable No. 1250 = Parameter type (I/O)
LET    1251  0      Variable No. 1251 = Device number (0, in the case of I/O
                    parameter)
LET    1252  30     Variable No. 1252 = Parameter number (No. 30)
ECMD   20    1250   Extended command 20 (Use variable Nos. 1250 through
                    1252)
                    Store the value of I/O parameter No. 30, "Input function
                    selection 000," in variable 99.
```

Chapter 4 Key Characteristics of Actuator Control Commands and Points to Note

1. Continuous Movement Commands [PATH, CIR, ARC, PSPL, CIR2, ARC2, ARCD, ARCC]

(1) By running a program with continuous movement commands input in a series of continuous program steps, you can allow the actuators to perform operations continuously without stopping between steps.

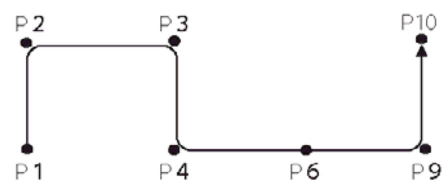
```
PATH   1   5
ARC2   6   7
PATH   8   12
```



(2) Continuous movement through positions is possible even when the specified positions are not continuous.
To do this, specify each discontinuous position number as both the start position number and end position number in a PATH command. In the example, position No. 6 is discontinuous.

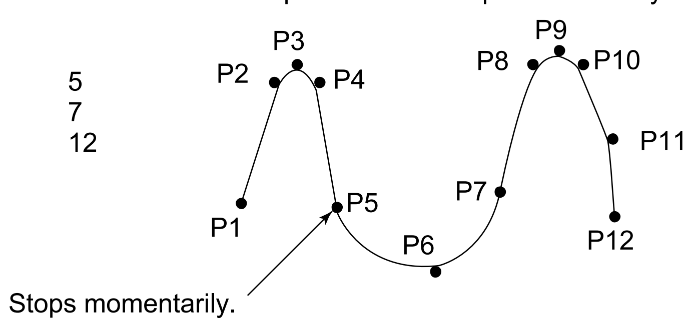
Move continuously through position Nos. 1, 2, 3, 4, 6, 9 and 10 in this order.

```
PATH   1   4
PATH   6   6 (Discontinuous position)
PATH   9   10
```

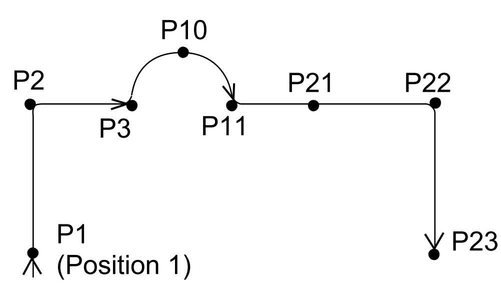


(3) Continuous movement will not be achieved if an input condition is specified for any continuous movement command.

```
     PATH   1   5
20   ARC2   6   7
     PATH   8   12
```



(4) The output field of each command will turn ON as the end position of that command approaches. Only with the last command in a series of continuous movement commands, the output will turn ON upon completion of operation (if there is no input condition).



[Example 1] (POTP = 1)

```
POTP   1
 :
PATH   1    3    600
ARC2   10   11   603
PATH   21   23   604
 :
```

Output field	Timing
600	Turn ON as P1 approaches.
601	Turn ON as P2 approaches.
602	Turn ON as P3 approaches.
603	Turn ON as P11 approaches.
604	Turn ON as P21 approaches.
605	Turn ON as P22 approaches.
606	Turn ON when P23 operation is complete.

[Example 2] (POTP = 0)

```
PATH   1    3    600
ARC2   10   11   603
PATH   21   23   604
```

Output field	Timing
600	Turn ON as P3 approaches.
603	Turn ON as P11 approaches.
604	Turn ON when P23 operation is complete.

[Example 3] If an input condition is specified, the output will turn ON upon completion of operation in the step before the one in which the input condition is specified.

```
     POTP   1
      :
     PATH   1    3    600
20   ARC2   10   11   603
     PATH   21   23   604
```

Output field	Timing
600	Turn ON as P1 approaches.
601	Turn ON as P2 approaches.
602	Turn ON when P3 operation is complete.
603	Turn ON as P11 approaches.
604	Turn ON as P21 approaches.
605	Turn ON as P22 approaches.
606	Turn ON when P23 operation is complete.

(5) When executing continuous movement commands sequentially, the controller is calculating approx. 100 positions ahead. This is why the steps are displayed continuously on the PC screen or teaching-pendant screen, regardless of the actual operation. The last step in the continuous operation section executed by continuous movement commands will wait for the applicable operation to complete.

```
 :
PATH   1    5    ← Actuator operation
ARC    6    7
PATH   8    12   ← Step displayed on the PC software or teaching pendant
BTON   310
 :
```

(6) Do not allow the output fields to duplicate in the continuous operation section executed by continuous movement commands.
Duplicating output fields in the continuous operation section will not achieve the expected result.
The output field will turn OFF at the start of processing of each command.

```
POTP   1                  Do not let outputs 605 and 604 to duplicate, as in the
                          example shown at left.
PATH   1    5    605
 :                        Continuous operation section executed by continuous
                          movement commands
PATH   11   15   604
```

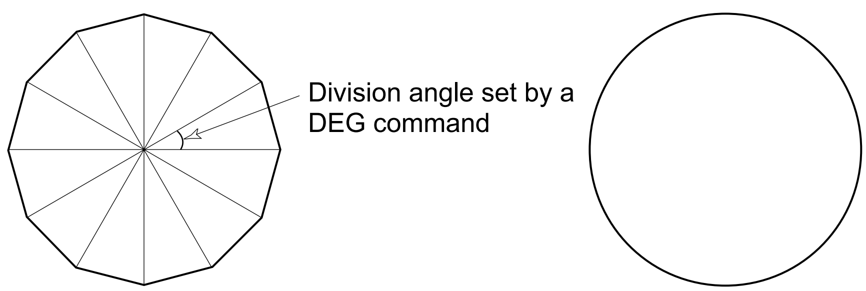
The final output status of duplicate 605 and 604 is indeterminable, because it is affected by the positioning calculation time and the relationship of durations of actual operations.

2. PATH/PSPL Commands

When executing a PATH or PSPL command, pay attention to the locus because it will change if the acceleration/deceleration is different between points.
The locus can be fine-tuned by changing the acceleration/deceleration, but different acceleration/deceleration settings between points will prevent smooth transition of speeds when moving from one position to another.
If there is a large difference in deceleration/acceleration between points and the positioning distance is small, the speed may drop. Exercise caution.

3. CIR/ARC Commands

The processing by a CIR or ARC command resembles moving along a polygon with a PATH command.
A small division angle may cause the speed to drop.
CIR2, ARC2, ARCD and ARCC commands actually perform arc interpolation.



CIR CIR2

4. CIR2/ARC2/ARCD/ARCC Commands

With a CIR2, ARC2, ARCD or ARCC command, the speed can be changed (only in the arc interpolation section) by inputting a speed for the point specified in operand 1. These commands are effective when you must lower the speed partially because the radius is small and the arc locus cannot be maintained inside the allowable range.

The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting in the position data specified in operand 1	Setting in the position data specified in operand 1
2	Setting by VEL command	Setting by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

Chapter 5 Palletizing Function (2-axis Specification)

The SEL language used by the ASEL Controller provides palletizing commands that support palletizing operation. These commands allow simple specification of various palletizing settings and enable arch motion ideal for palletizing.

1. How to Use

Use palletizing commands in the following steps:

(1) Palletizing setting
Set palletizing positions, arch motion, etc., using palletizing setting commands.

(2) Palletizing calculation
Specify palletizing positions using palletizing calculation commands.

(3) Palletizing movement
Execute motion using palletizing movement commands.

2. Palletizing Setting

Use the palletizing setting commands to set items necessary for palletizing operation. The setting items include the following:

(1) Palletizing number setting --- Command: BGPA
At the beginning of a palletizing setting, determine a palletizing number using a BGPA command to declare the start of palletizing setting.
At the end, declare the end of palletizing setting using an EDPA command.

```
BGPA    1    Declare the start of setting for palletizing No. 1.
 |
 |           Set palletizing in these steps.
 |
EDPA         Declare the end of palletizing setting at the end.
```

A maximum of 10 sets (palletizing Nos. 1 to 10) of palletizing setting can be specified for each program.

(2) Palletizing pattern --- Command: PAPN
Select a pattern indicating the palletizing order.
The two patterns illustrated below are available.
The encircled numbers indicate the order of palletizing and are called “palletizing position numbers.”

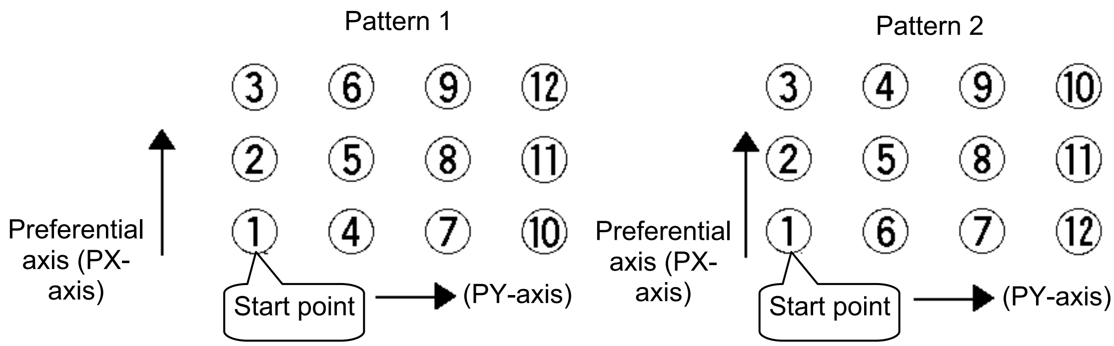


Fig. 1

PAPN   2   When pattern 2 is selected
(Setting is not necessary if pattern 1 is selected.)

The row from 1 to 3 to be placed first is called the “preferential axis (PX-axis),” while the other direction comprising the palletizing plane is called the “PY-axis.”

(3) Palletizing counts --- Command: PAPI
Set the palletizing counts.

PAPI   3   4   Count for preferential axis (PX-axis): 3, Count for PY-axis: 4

(4) Palletizing position setting
Palletizing position setting is performed mainly by method A or B, as explained below. Set the palletizing positions for each palletizing setting based on method A or B.

	Setting method	Commands
A	3-point teaching method Set three position-data points specifying the palletizing positions.	PAPS
B	Method to set palletizing positions in parallel with the actuators Set from the palletizing axes, palletizing reference point and palletizing pitches.	PASE, PAST, PAPT

A. 3-point teaching method

To set the palletizing positions by 3-point teaching, store desired positions in position data fields as three continuous position data and then specify the first position number using a PAPS command. This method allows you to set the PX-axis and PY-axis as three-dimensional axes not parallel with the actuators and not crossing with each other.
In the example shown below, position data ①, ③ and ⑩ are stored in three continuous position data fields.

When three points are taught from position No. 11

Position No. 11	[1]:	Start point (First palletizing position)
Position No. 12	[3]:	Palletizing position corresponding to the end point in the PX-axis direction
Position No. 13	[10]:	Palletizing position corresponding to the end point in the PY-axis direction

The encircled numbers indicate palletizing position numbers (palletizing order).

Use a PAPS command to specify the position number corresponding to the start point.

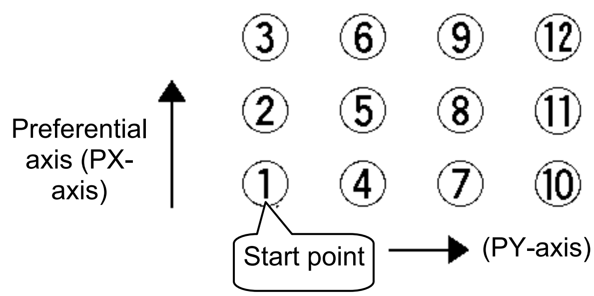


Fig. 1

```
PAPS    11
```

The pitches are calculated automatically from the count set for each axis.
When setting data for 3-point teaching, specify position data for two axes.

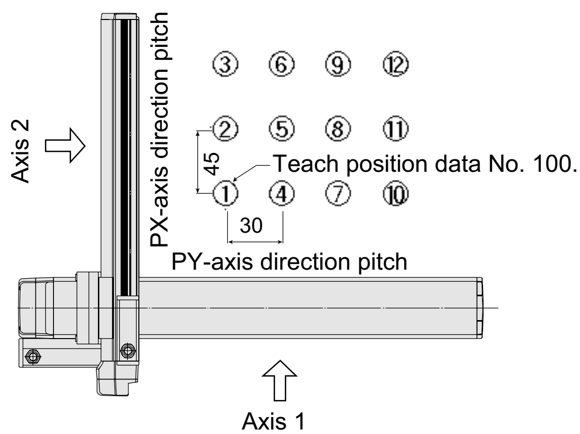
B. Method to set palletizing positions in parallel with the actuators

Palletizing reference point: Store the position data of the start point (palletizing position No. 1) in a position data field and specify the applicable position number using a PAST command, as shown below.

Palletizing pitches: Use a PAPT command to specify the pitches in the PX-axis and PY-axis directions.

Palletizing axes: Use a PASE command to specify the two axes, one representing the PX-axis direction and the other representing the PY-axis direction, to be used in palletizing.

(An actuator axis number parallel with the preferential axis (PX-axis) and another perpendicular to the preferential axis)



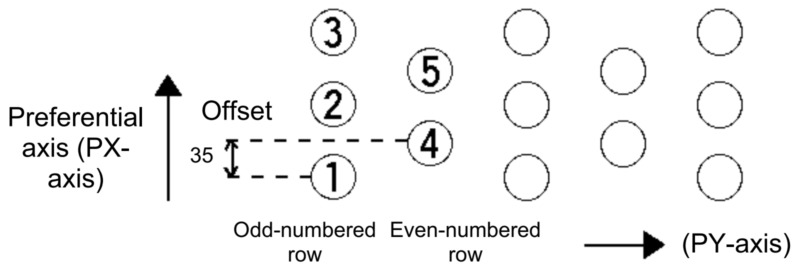
PAST	100		Teach position data No. 100 as the start point.
PAPT	45	30	The PX-axis direction pitch is 45 mm and the PY-axis direction pitch is 30 mm.
PASE	2	1	Set the PX-axis as axis 2 and PY-axis as axis 1.

(Note) When the above palletizing axes, palletizing pitches and palletizing reference point are used, the PX-axis and PY-axis must be parallel with the actuators and crossing with each other.

Select either method A or B for each palletizing setting.

(5) Zigzag setting --- Command: PSLI
Use a PSLI command to set a zigzag layout as shown below.
Zigzag offset: Offset amount in the preferential-axis direction, which will be applied when even-numbered rows are placed.
"Even-numbered rows" refer to the rows occurring at the even numbers based on the row placed first representing the first row.

Zigzag count: Number in the even-numbered rows. Two in the diagram below.



PSLI 35 2

3. Palletizing Calculation

The items that can be operated or obtained using palletizing calculation commands are shown below:

(1) Palletizing position number Commands --- PSET, PINC, PDEC, PTNG
Number showing the ordinal number of a palletizing point.
(In Fig. 1 given in the explanation of palletizing pattern, the encircled numbers are palletizing position numbers.)

Always set this command before executing a palletizing movement command --- PSET

For example, executing a palletizing movement command by setting 1 as the palletizing position number will move the axes to the start point. Executing a palletizing movement command by setting 2 as the palletizing position number will move the axes to the point immediately next to the start point in the PX-axis direction.

(2) Palletizing angle Command --- PARG
This is the angle formed by the physical axis (actuator) and the preferential palletizing axis (PX-axis) (θ in the figure below).
In the figure below, θ will become a negative value if axis 1 is used as the reference for angle calculation.

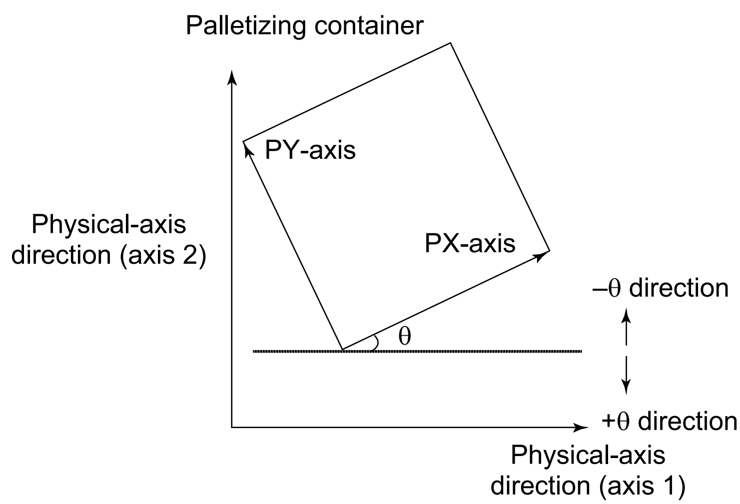


Fig. 4

With ASEL commands, executing a “get palletizing angle” command following a palletizing setting via 3-point teaching will automatically obtain the palletizing angle.

(3) Palletizing calculation data Command --- PAPG
When a palletizing position number is set, this data refers to the position coordinate data of the palletizing point corresponding to that palletizing position number.
Note that this position coordinate data does not reflect normal offset or palletizing Z-axis offset.

4. Palletizing Movement

Palletizing movement commands are used to move the actuator to palletizing points.

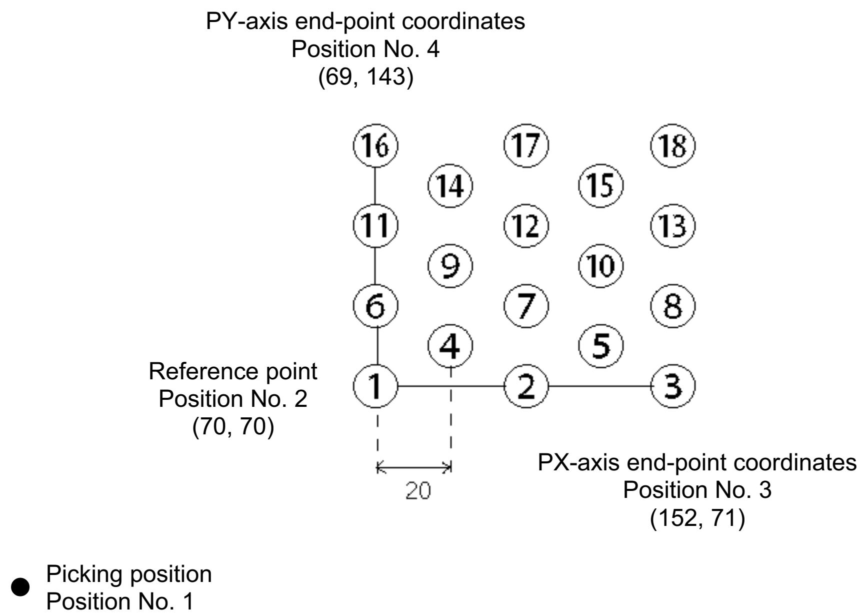
(1) Movement commands to palletizing point --- PMVP, PMVL
Position coordinates of a two-dimensionally placed palletizing point are calculated and movement is performed using the calculated point as the end point. (The axes will move to the palletizing point of the palletizing position number specified in the executed command.)
Two actuator axes will be required to comprise a two-dimensional plane.
PMVP: Move from the current position to a palletizing point via PTP.
PMVL: Move from the current position to a palletizing point via interpolation.

5. Program Examples

(1) Simple program example (two-axis specification) using PAPS (set by 3-point teaching)
The example below specifies movement only and does not cover picking operation.

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					BGPA	1			ﾊﾟﾚﾀｲｽﾞNo1設定開始
2					PAPI	3	7		ﾊﾟﾚﾀｲｽﾞ個数3×7
3					PAPS	2			3点ﾃｨｰﾁﾝｸﾞ設定
4					PSLI	20	3		千鳥ｵﾌｾｯﾄ=20mm
5					EDPA				ﾊﾟﾚﾀｲｽﾞNo1設定終了
6									
7					VEL	200			速度200mm/sec
8					MOVP	1			ﾋﾟｯｸ位置へ移動
9					PSET	1	1		ﾊﾟﾚﾀｲｽﾞ位置No=1ｾｯﾄ
10					TAG	1			
11					PMVP	1			ﾊﾟﾚﾀｲｽﾞ点PTP移動
12					MOVP	1			ﾋﾟｯｸ位置PTP移動
13					PINC	1		600	ﾊﾟﾚﾀｲｽﾞ位置No.+1
14				600	GOTO	1			PINC成功時ﾙｰﾌﾟ先頭
15					EXIT				
16									

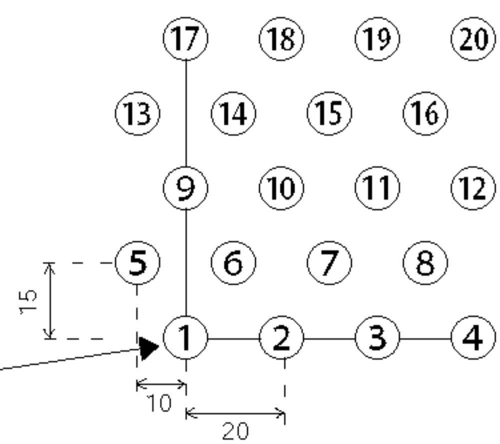
No.	Axis1	Axis2	Vel	Acc	Dcl
1	10.000	10.000			
2	70.000	70.000			
3	152.000	71.000			
4	69.000	143.000			
5					



(2) Simple program example (two-axis specification) using PAPS, PAPT and PAST
The example below specifies movement only and does not cover picking operation.

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					BGPA	2			ﾊﾟﾚﾀｲｽﾞNo2設定開始
2					PAPI	4	5		ﾊﾟﾚﾀｲｽﾞ個数4×5
3					PASE	1	2		PX軸＝1軸,PY軸＝2
4					PAPT	20	15		ﾋﾟｯﾁX＝20,Y=15
5					PAST	11			ﾎﾟｼﾞｼｮﾝNo.11基点
6					PSLI	-10	4		千鳥ｵﾌｾｯﾄ＝-10mm
7					EDPA				ﾊﾟﾚﾀｲｽﾞNo2設定終了
8									
9					VEL	200			速度200mm/sec
10					MOVP	10			ﾋﾟｯｸ位置へ移動
11					PSET	2	1		ﾊﾟﾚﾀｲｽﾞ位置No=1ｾｯﾄ
12					TAG	1			
13					PMVP	2			ﾊﾟﾚﾀｲｽﾞ点PTP移動
14					MOVP	10			ﾋﾟｯｸ位置PTP移動
15					PINC	2		600	ﾊﾟﾚﾀｲｽﾞ位置No.＋1
16				600	GOTO	1			PINC成功時ﾙｰﾌﾟ先頭
17					EXIT				
18									

No.	Axis1	Axis2	Vel	Acc	Dcl
10	10.000	10.000			
11	70.000	70.000			
12					



Reference point
Position No. 11
(70, 70)

● Picking position
Position No. 10

Chapter 6 Pseudo-Ladder Task

With the ASEL Controller, a pseudo-ladder task function can be used depending on the command and extension condition.
The input format is shown below. Note that this function must be used by expert engineers with a full knowledge of PLC software design.

1. Basic Frame

Extension condition	N	Input condition	Command	Operand 1	Operand 2	Output	
E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	
LD		7001	CHPR	1			
			TPCD	1			
			TAG	1			
I		I	I	I			
I		I	I	I			
I		I	I	I			Ladder statement field
I		I	I	I			
I		I	I	I			
I		I	I	I			
LD		7001	TSLP	1 ~ 100			
I		I	I	I			
I		I	I	I			
I		I	I	I			Ladder statement field
I		I	I	I			
I		I	I	I			
I		I	I	I			
LD		7001	TSLP	1 ~ 100			
LD		7001	GOTO	1			
LD		7001	EXIT				

*   * Virtual input 7001: "Normally ON" contact

2. Ladder Statement Field

[1] Extension conditions

LD .... LOAD
A .... AND
O .... OR
AB .... AND BLOCK
OB .... OR BLOCK

All of the above extension conditions can be used in non-ladder tasks.

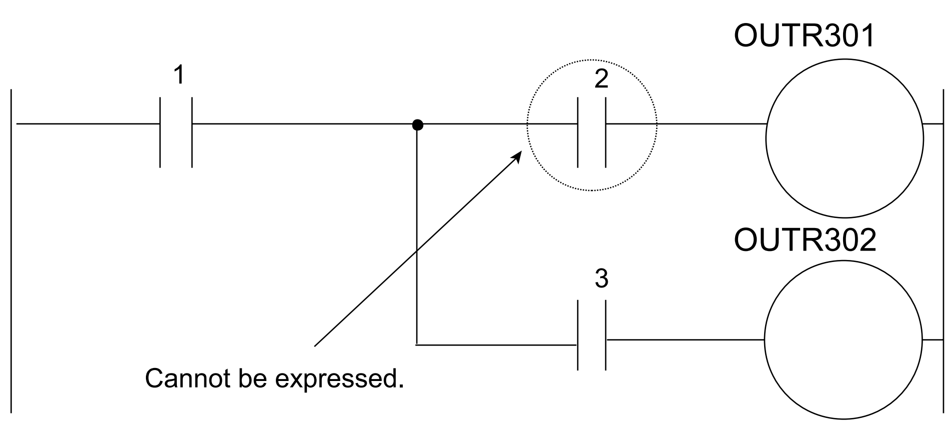
[2] Ladder commands

OUTR .... Ladder output relay (Operand 1 = Output, flag number)
TIMR .... Ladder timer relay (Operand 1 = Local flag number, Operand 2 = Timer setting (sec))

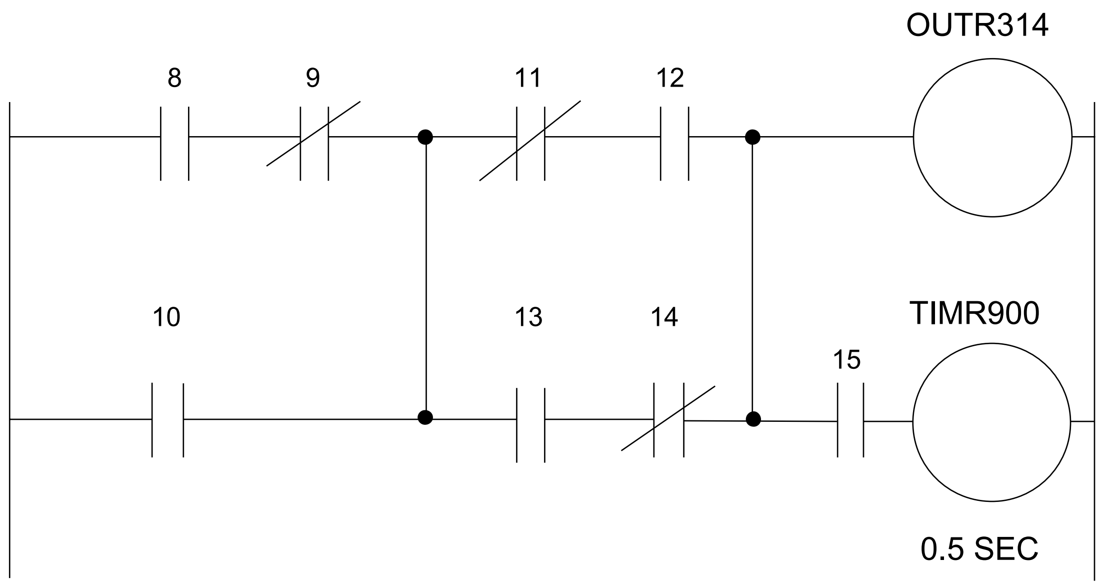
3. Points to Note

- This system only processes software ladders using an interpreter. Therefore, the processing time is much longer than that of a dedicated commercial sequencer.
  (This system is not suitable for large-scale ladder processing.)
- If an extension condition is not specified for steps in which an input condition is specified, the steps will be treated as LD (LOAD).
- Always specify a "normally ON" contact for those steps that must be processed without fail, such as CHPR, TSLP and GOTO. (LD 7001)
  Virtual input 7001: "Normally ON" contact

- The following circuit cannot be expressed. Create an equivalent circuit.



4. Program Example



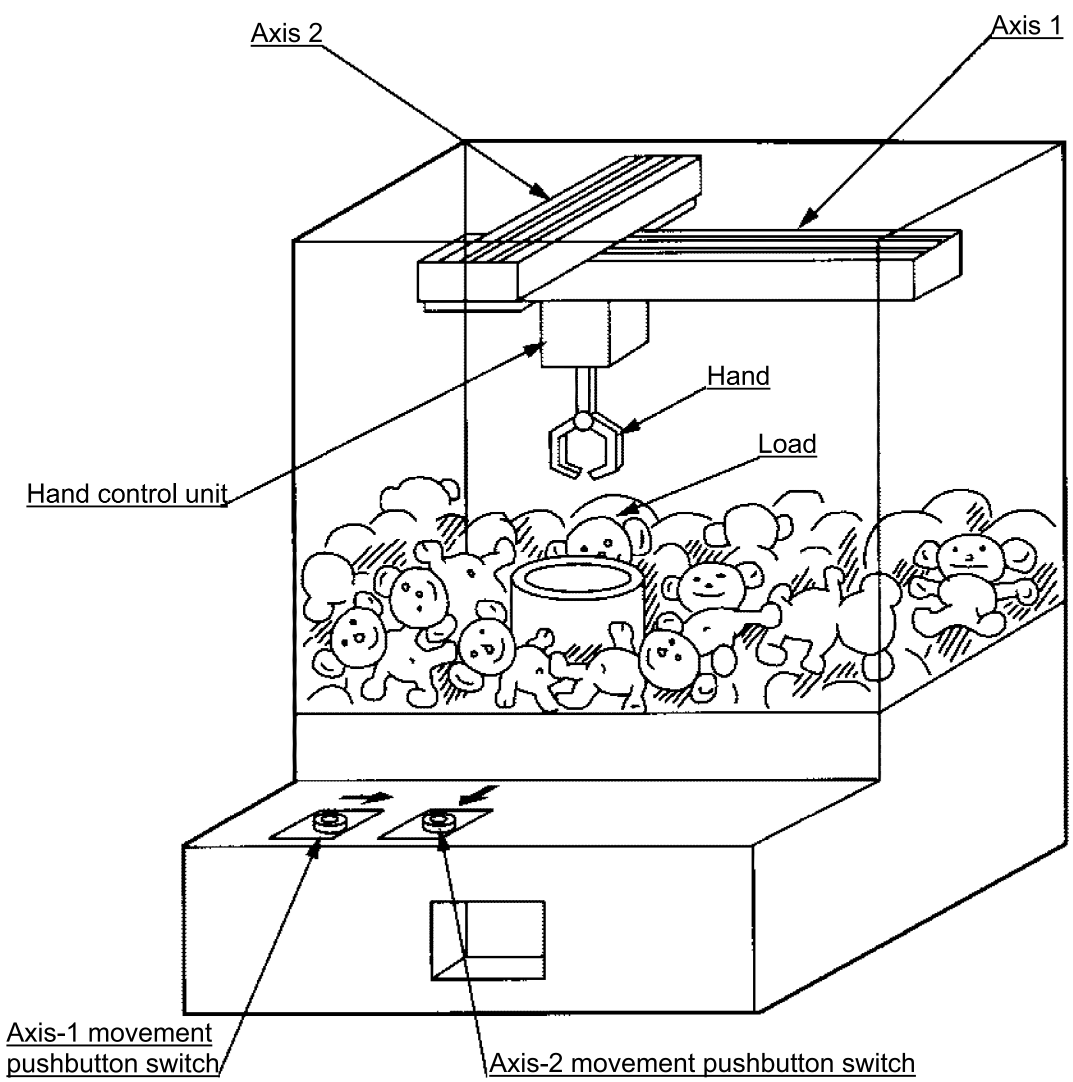
Extension condition	N	Input condition	Command	Operand 1	Operand 2	Output
E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
LD		7001	CHPR	1		
			TPCD	1		
			TAG	1		
LD		8				
A	N	9				
O		10				
LD	N	11				
A		12				
LD		13				
A	N	14				
OB						
AB			OUTR	314		
A		15	TIMR	900	0.5	
LD		7001	TSLP	3		
LD		7001	GOTO	1		
LD		7001	EXIT			

Chapter 7 Application Program Examples

1. Operation by Jog Command [Doll-Picking Game Machine]

(1) Overview of the system

This system is a doll-picking game machine consisting of axis-1 and axis-2 actuators. Pushbutton switches corresponding to the two axes are provided on an external operation switch box, and these switches are used to move the actuators to a desired position to grab and pick up dolls inside the case.

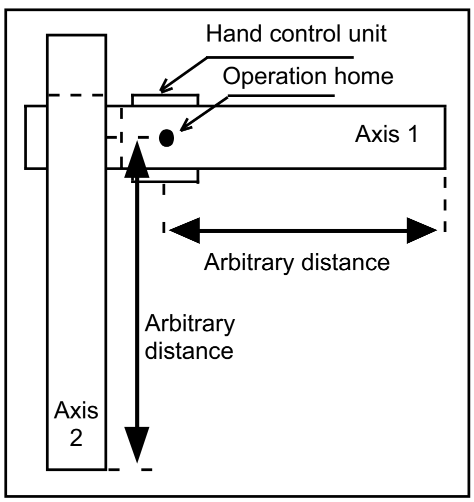


(2) Explanation of the operation

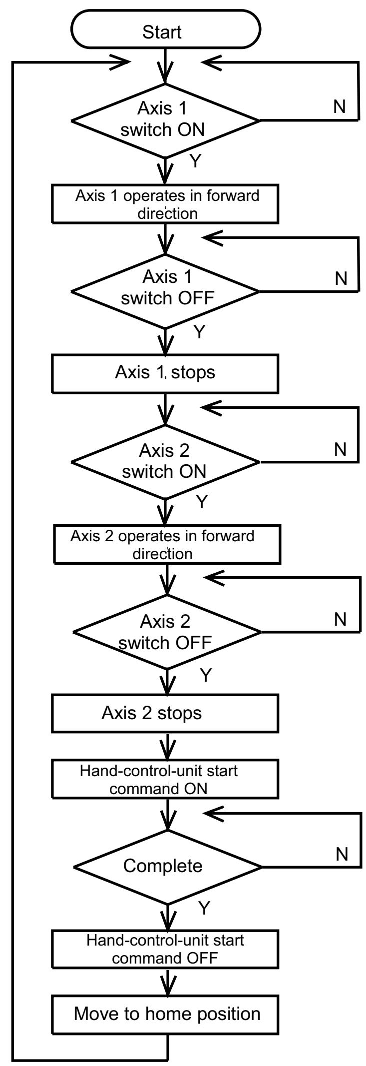
1. Wait for the axis-1 movement pushbutton switch to turn ON.
2. The X-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
3. Wait for the axis-2 movement pushbutton switch to turn ON.
4. The Y-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
5. Output a start command to the hand control unit.
6. Wait for an operation completion input from the hand control unit.
7. Move to the home after the input is received.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position



Operation Flow Chart



I/O Assignments

Category		I/O No.	Signal name	Specification
ASEL	Input	16	Axis-1 movement command	Pushbutton switch
		17	Axis-2 movement command	Pushbutton switch
		18	Hand operation completion	External control unit
	Output	307	Hand start command	24 VDC
	* Flag is not used.			

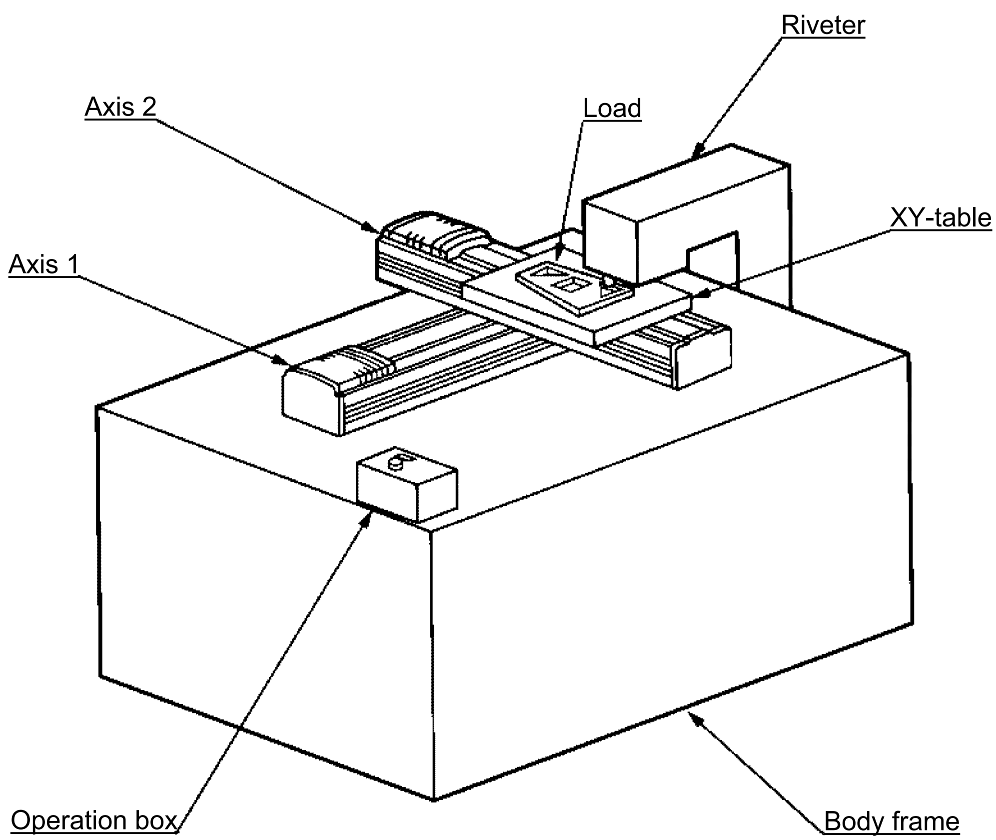
(3) ASEL Controller application program

Step	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1				HOME	11			Axes 1 and 2 return to home (servo ON).
2				VEL	400			Set speed to 400 mm/s.
3				TAG	1			
4				WTON	16			Wait for input from axis-1 movement switch.
5				JFWN	1	16		Move forward while axis-1 movement switch is ON.
6				WTON	17			Wait for input from axis-2 movement switch.
7				JFWN	10	17		Move forward while axis-2 movement switch is ON.
8				BTON	307			Start command for external control unit turns ON.
9				WTON	18			Wait for external control unit to complete operation.
10				BTOF	307			Start command for external control unit turns OFF.
11				JBWF	11	18		Axes 1 and 2 move backward while 18 is ON.
12				GOTO	1			Jump to TAG1.
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								

2. Operation by Point Movement Command [Riveting System]

(1) Overview of the system

This system is a riveting system consisting of an XY-table operated by axis-1 and axis-2 actuators and a riveter. By setting a load on the XY-table at the operation home and turning on the start switch, rivets will be driven at the three points specified on the load.

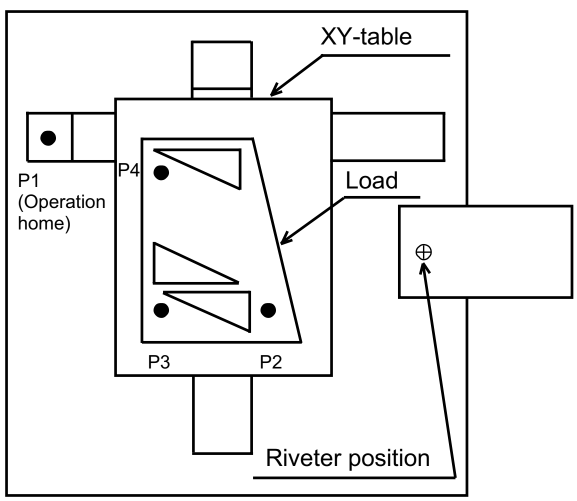


(2) Explanation of the operation

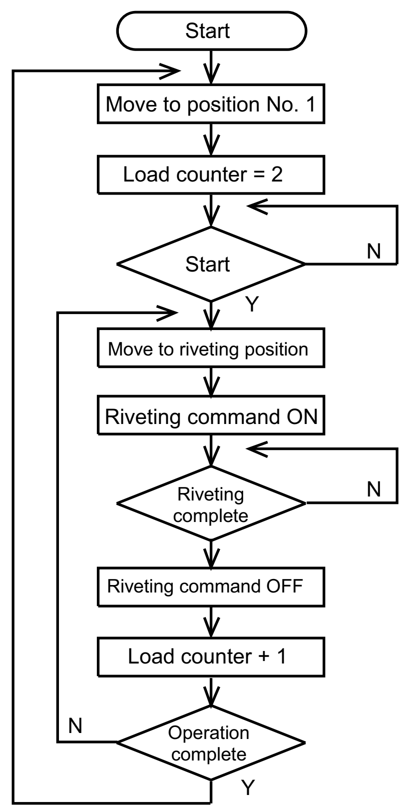
[1] The XY-table moves to the operation home (P1) and waits.
[2] The operator sets a load on the XY-table and turns on the start switch.
[3] The load riveting position No. 1 (P2) moves to the riveting position on the XY-table, and a riveting command is output to the riveter.
[4] When the riveter completes the riveting operation and a completion signal is input, riveting position Nos. 2 (P3) and 3 (P4) move to the riveting position in the same manner.
[5] When all three points have been riveted, the table will return to the operation home.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position



Operation Flow Chart



I/O Assignments

Category		I/O No.	Signal name	Specification
ASEL	Input	16	Start command	Pushbutton switch
		17	Riveting completion	Contact signal
	Output	307	Riveting command	24 VDC
	* Flag is used from 600.			

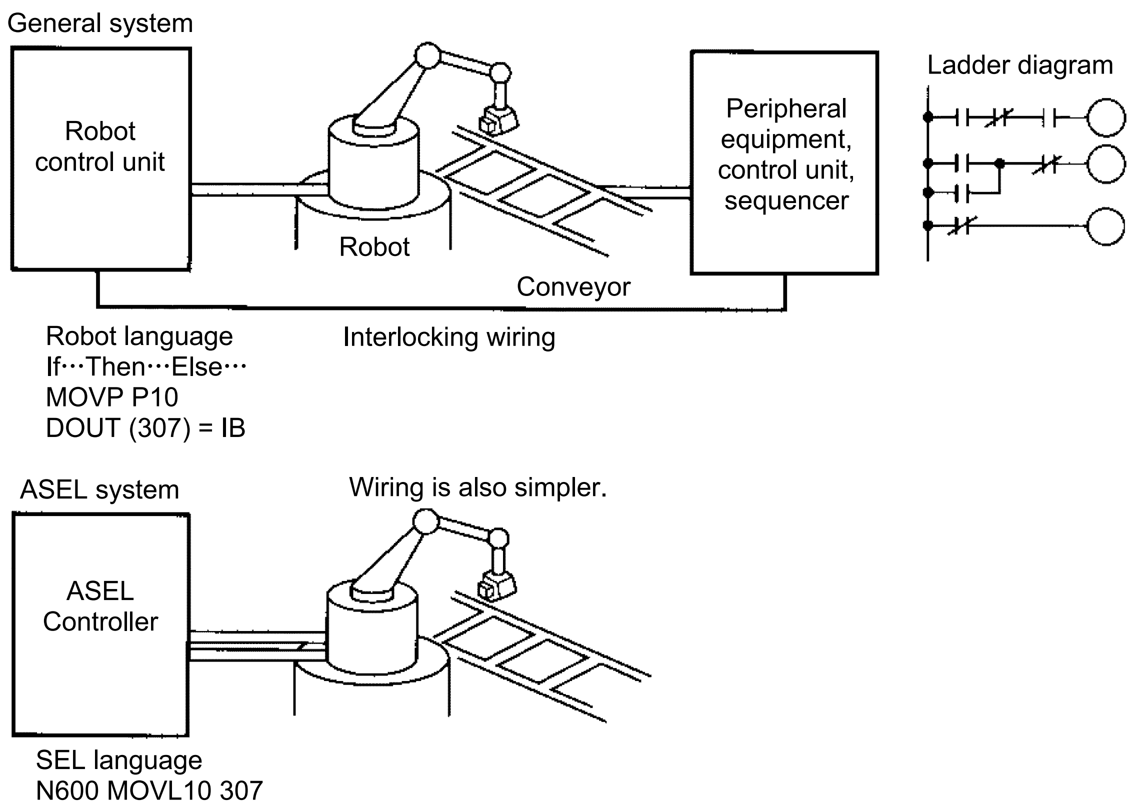
(3) ASEL Controller application program

Step	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1				HOME	11			XY-table returns to home (servo ON).
2				VEL	400			Set speed to 400 mm/s.
3				TAG	1			
4				MOVL	1			Move to position No. 1 (operation home)
5				LET	1	2		Set 2 in load counter.
6				BTOF	600			Clear completion flag.
7				WTON	16			Wait for start command.
8				TAG	2			
9				MOVL	*1			Move to load counter position.
10				BTON	307			Riveting command turns ON.
11				WTON	17			Wait for riveting to complete.
12				BTOF	307			Riveting command turns OFF.
13				ADD	1	1		Increment load counter by 1.
14				CPEQ	1	5	600	Turns ON flag if operation is complete.
15		N	600	GOTO	2			Jump to TAG2 if not complete.
16				GOTO	1			Jump to TAG1 if complete.
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								

Chapter 8 Real-Time Multi-Tasking

1. SEL Language

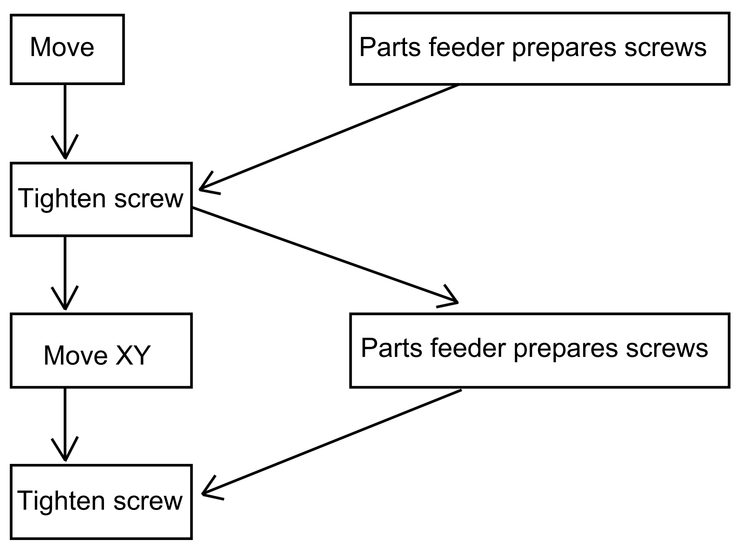
The ASEL Controller allows integrated control of actuators and peripherals with a single controller using its 32-bit RISC CPU and high-speed real-time operating system. There is no need to learn various languages for different units, such as robot language for robots and sequencer language for peripherals. Since SEL language is the only language used, an efficient system can be designed.
The current version of SEL language represents a pioneering evolution of the widely proven programming language, evidenced by higher-performance features and advanced functions. The latest version is also easier to use compared with the conventional SEL language.



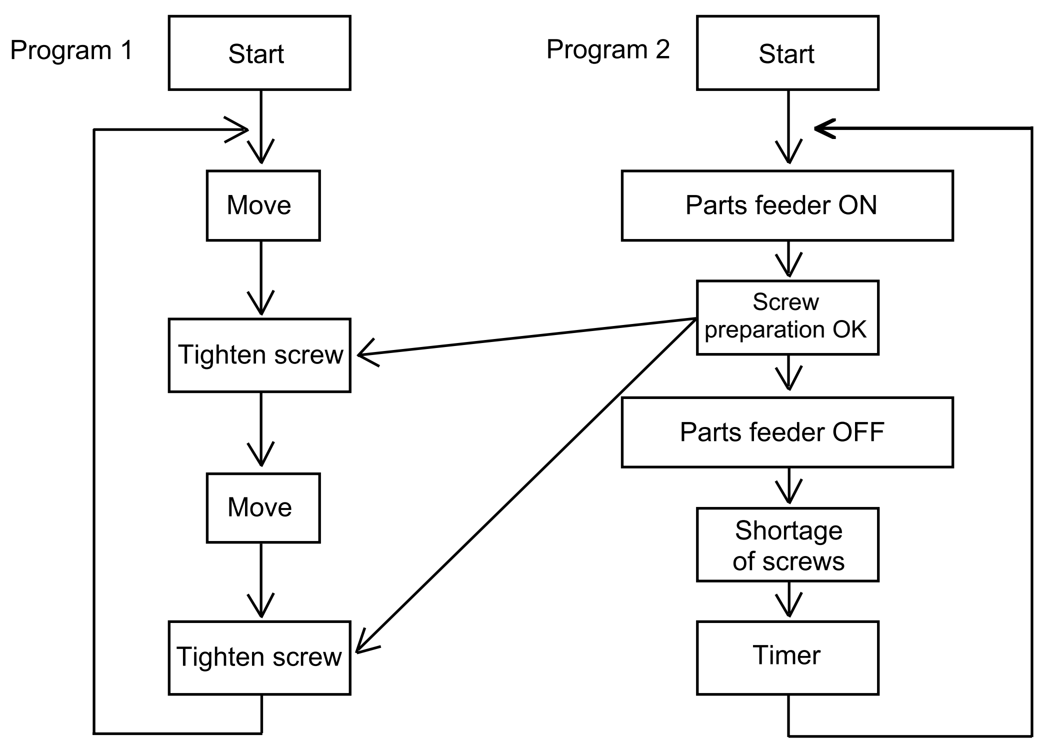
2. Multi-Tasking

"Multi-tasking" operation may not be a familiar term, but it is widely used in computer programming to refer to parallel processing. Simply put, multi-tasking means running several programs in parallel.
Take a screw-tightening robot, for example.
In general, a screw-tightening robot consists of axis-1 and axis-2 actuators and a screw-tightening machine (up/down air cylinder, etc.).

Operation Flow



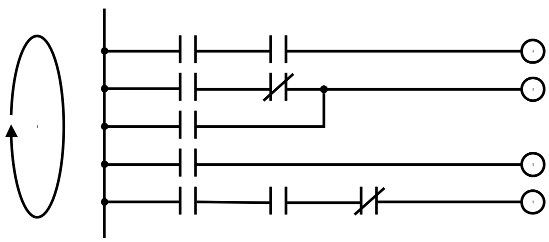
Although the flow chart is simple, the movement of axis-1 and axis 2 actuators and the operation of the parts feeder must take place simultaneously. This requires "multi-tasking" operation.



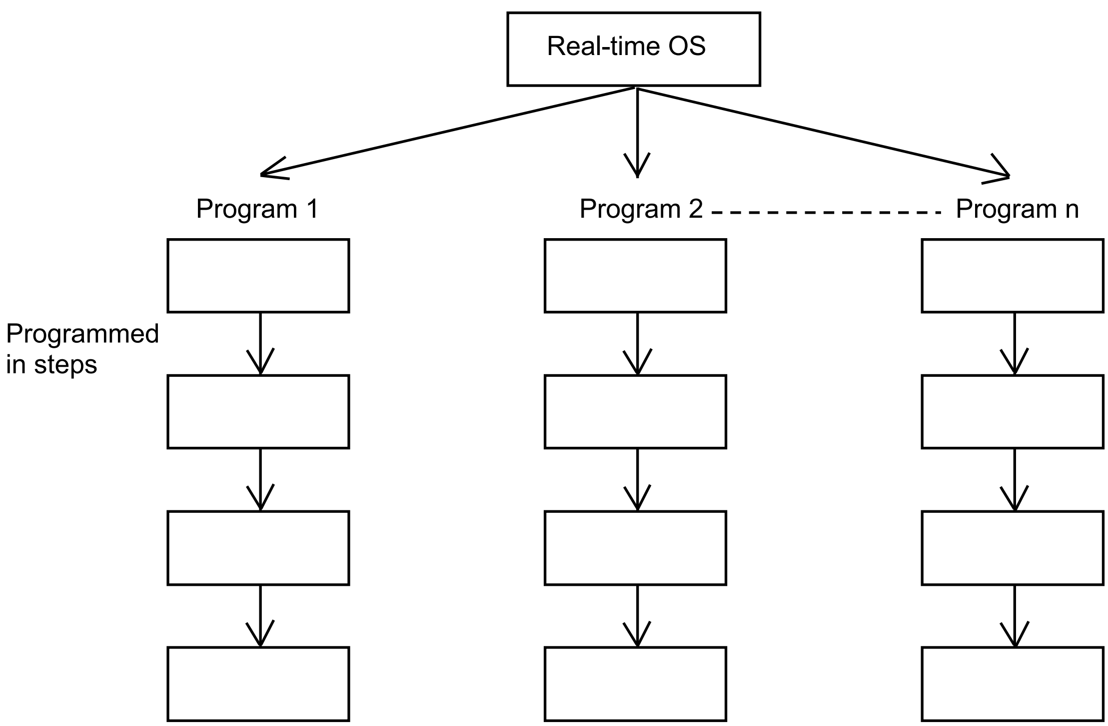
3. Difference from a Sequencer

The parallel processing method has evolved from the traditional method of using a sequence control circuit consisting of relays to a more recent one using a sequencer equipped with a microcomputer. Since a microcomputer basically allows one process for each clock, a sequence control circuit with a microcomputer must scan the entire program to achieve apparent parallel processing. For this reason, a scan time is required, which adds to overhead (dead time).

The microcomputer scans the enter program and outputs only where the condition is satisfied.



On the other hand, a system consisting of a microcomputer and a real-time operating system no longer uses parallel processing scan (by always scanning the entire program), but adopts an event-driven method instead (whereby the system operates only when an event occurs, such as upon receipt of an input signal). Since no extra scan is necessary, the system can operate at high speed. In addition, each program to be processed in parallel is programmed in steps, so the program is easy to understand and maintain.



The programmer need not worry about running all programs in parallel, which is controlled by the real-time operating system.

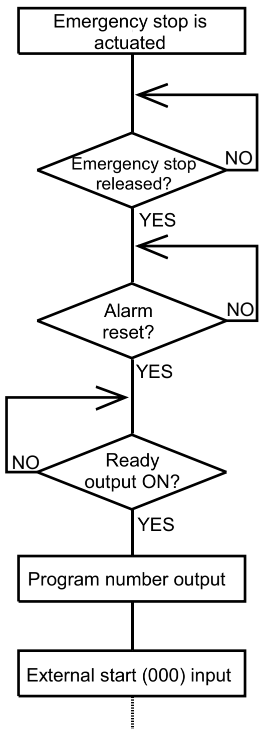
4. Release of Emergency Stop

Default factory settings of parameters

“Other parameter No. 10, Emergency-stop recovery type” = 0
“Other parameter No. 11, Safety-gate open recovery type” = 0
“Other parameter No. 12, Recognition type during automatic operation” = 0

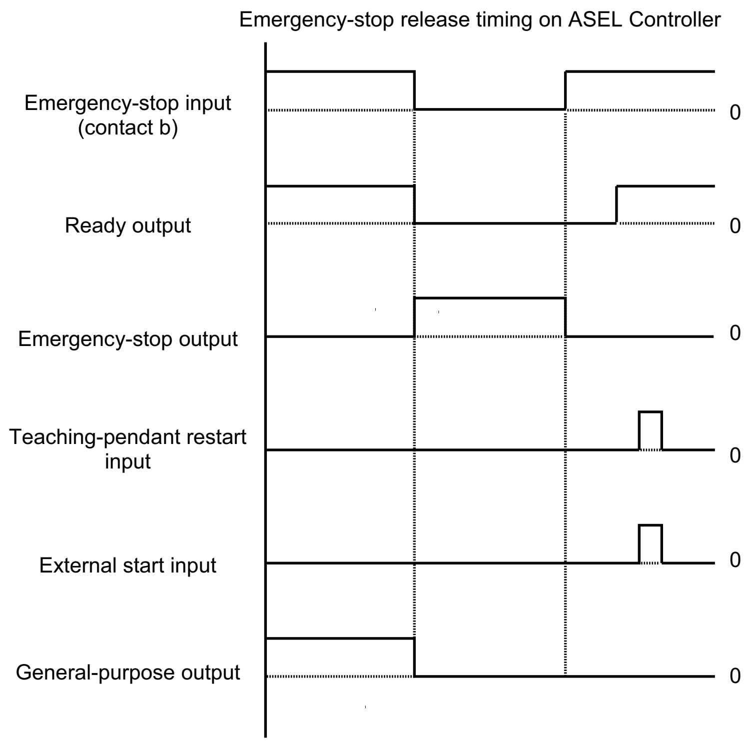
An emergency stop is actuated by turning the emergency-stop contact b input to OFF, and released by turning the input to ON.

(1) Flow chart



The selected program is executed from step 1.

(2) Timing chart



◎ The internal conditions of the controller during an emergency stop are as follows:

- Programs .... Aborted (excluding “I/O processing programs operation when program is aborted”)
- Output ports, local flags, local variables .... Cleared
- Global flags, global variables .... Retained

If the peripherals are to be controlled by program, create a management program beforehand and use the program to control the peripherals. Alternatively, start (EXPG) or abort (ABPG) other programs in accordance with the status of each general-purpose input.

5. Program Switching

Various methods are available to switch between programs, depending on the purpose of programs. The representative methods are explained below.

- Program switching
  - External start
  - Program
    - Single-tasking .... EXIT command
    - Multi-tasking .... EXPG command

First, the program switching methods are largely divided into switching by external start and switching by application program.

(1) External start method .... Refer to Chapter 4, 2.2, “Standing via External Signal Selection” in Part 1.

(2) Program method

○ Single-tasking

Executing an EXIT command (end program) at the end of each program will end the program and cause the system to return to the condition immediately after the power is turned on. However, since the home position is retained, another program can be started by an external start input with the corresponding program number specified.

○ Multi-tasking

Creating a management program and executing EXPG commands (start other program) will allow a series of programs to be run in parallel.

Chapter 9 Example of Building a System

How to build hardware and software is explained in details by using a screw-tightening robot as an example.

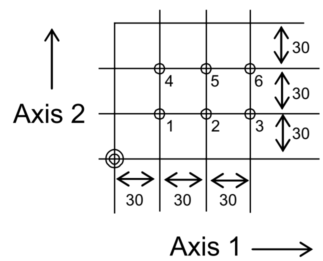
1. Equipment

Screw-tightening machine (for Z-axis)	
Actuators (for axes 1 and 2)	IAI's actuator with 300-mm stroke x 2
Controller	IAI's ASEL controller

2. Operation

(1) Tighten six screws at 30-mm pitches on axes 1 and 2.
1. The actuators move to a screw-tightening position.
2. The Z-axis air cylinder of the screw-tightening machine comes down.
3. The screw-tightening machine starts operating.
4. When the screw tightening is complete, the Z-axis air cylinder rises.
5. The actuators move to the next position.

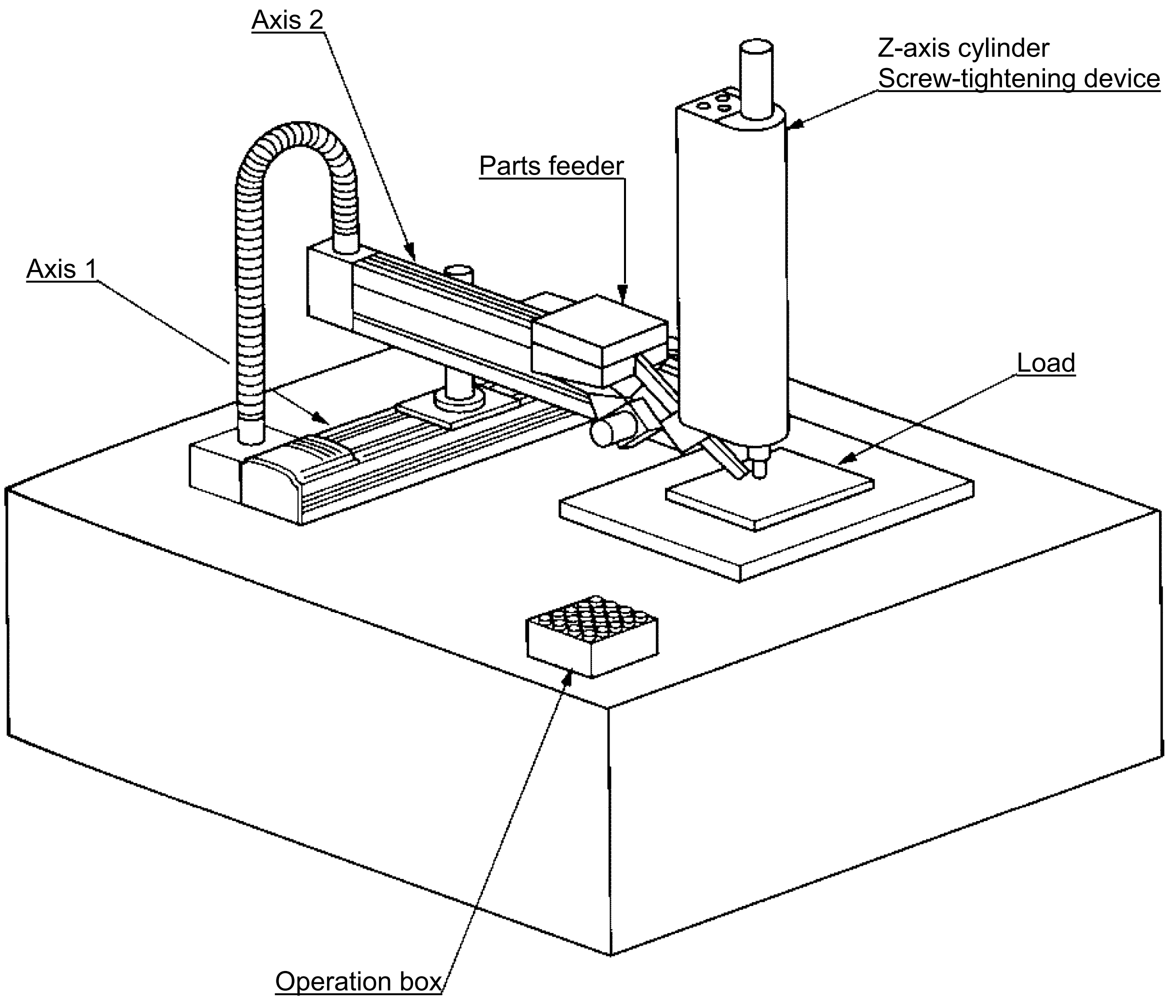
Coordinates



(2) The parts feeder operates in parallel with the above operation.
1. The parts feeder starts when screws are short.
2. The parts feeder stops when the screws are fully loaded.

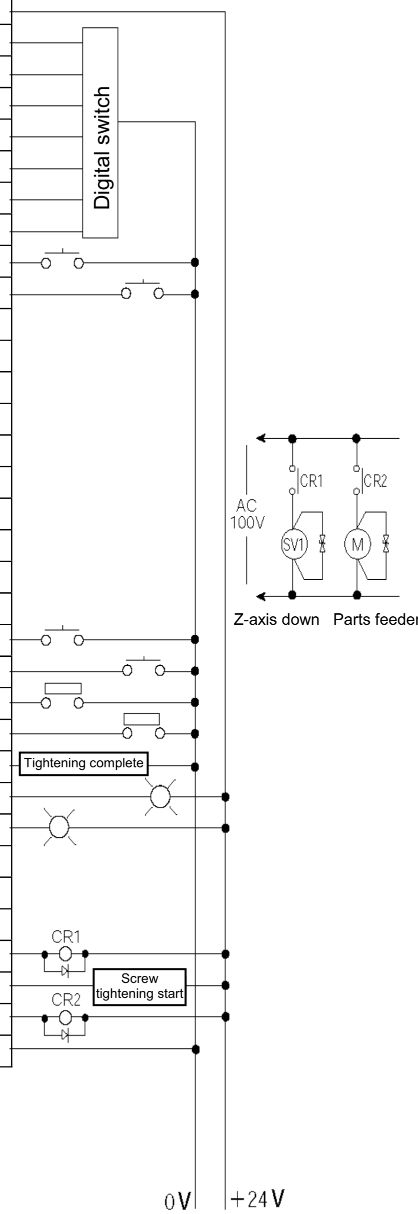
3. Overview of the Screw-Tightening System

This system consists of axis-1 and axis-2 actuators, Z-axis cylinder, screw-tightening device and parts feeder, and tightens the screws fed by the parts feeder at the specified positions on the load.



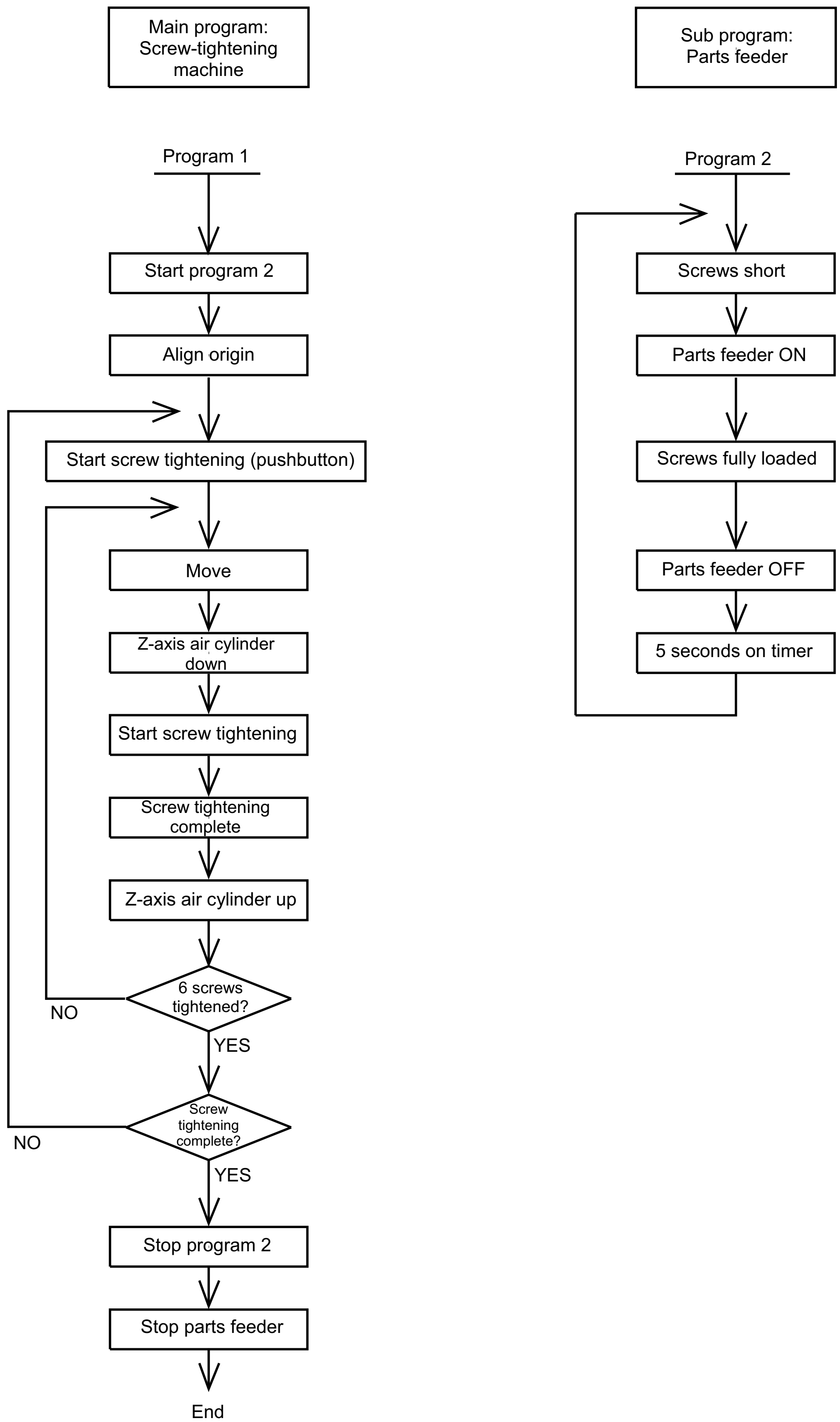
4. Hardware

Pin No.	Category	Port No.	Function
1A	P24		External power supply 24 V
1B	Input	016	Program specification (PRG No. 1)
2A		017	Program specification (PRG No. 2)
2B		018	Program specification (PRG No. 4)
3A		019	Program specification (PRG No. 8)
3B		020	Program specification (PRG No. 10)
4A		021	Program specification (PRG No. 20)
4B		022	Program specification (PRG No. 40)
5A		023	Software reset (restart)
5B		000	Program start
6A		001	General-purpose input
6B		002	General-purpose input
7A		003	General-purpose input
7B		004	General-purpose input
8A		005	General-purpose input
8B		006	General-purpose input
9A		007	General-purpose input
9B		008	General-purpose input
10A		009	General-purpose input
10B		010	General-purpose input
11A		011	General-purpose input (Screw tightening start)
11B		012	General-purpose input (Screw tightening end)
12A		013	General-purpose input (Z-axis air cylinder top)
12B		014	General-purpose input (Parts-feeder all screws tightened)
13A		015	General-purpose input (Screw tightening complete)
13B	Output	300	Alarm output
14A		301	Ready output
14B		302	General-purpose output
15A		303	General-purpose output
15B		304	General-purpose output
16A		305	General-purpose input (Z-axis air cylinder down)
16B		306	General-purpose input (Screw tightening start)
17A		307	General-purpose input (Parts feeder start)
17B	N		External power supply 0 V



5. Software

(1) Control flow chart



(2) Main program
Screw-tightening program No. 1

Application program

Comment	Extension condition	Input condition	Command			Output condition	Comment
	AND, OR	I/O, flag	Command	Operand 1	Operand 2	Output port, flag	
1			EXPG	2			Start program 2.
2			HOME	11			Align home.
3			VEL	100			Speed: 100 mm/sec
4			ACC	0.3			Acceleration: 0.3 G
5			TAG	1			Jump destination at restart
6			WTON	11			Screw-tightening start pushbutton
7			LET	1	1		Set screw counter.
8			TAG	2			Jump destination after tightening one screw
9			MOVL	*1			Move.
10			BTON	305			Z-axis air cylinder down
11			BTON	306			Start screw tightening.
12			WTON	12			Screw tightening complete.
13			BTOF	305	306		Cylinder up, screw tightening stopped.
14			WTON	13			Check Z-axis air cylinder top position.
15			ADD	1	1		Increment screw counter by 1.
16			CPEQ	1	7	900	Compare after tightening six screws.
17		N900	GOTO	2			Go to next screw-tightening cycle after tightening one screw.
18		N17	GOTO	1			Restart screw tightening.
19			ABPG	2			Stop program 2.
20			BTOF	307			Stop parts feeder.
21			EXIT				End of program 1

Position program

No.	X	Y
1	30	30
2	60	30
3	90	30
4	30	60
5	60	60
6	90	60

(3) Sub program
Parts feeder program No. 2

Application program

Comment	Extension condition	Input condition	Command			Output condition	Comment
	AND, OR	I/O, flag	Command	Operand 1	Operand 2	Output port, flag	
1			TAG	1			Jump destination for repeating
2			WTOF	14			Screws short.
3			BTON	307			Start parts feeder.
4			WTON	14			Screws fully loaded.
5			BTOF	307			Stop parts feeder.
6			TIMW	5			5 seconds on restart timer
7			GOTO	1			Repeat.

Chapter 10 Example of Building a System

1. Position Table

Position Table

Up to 1,500 position points can be registered in the ASEL controller.
Positions are registered using the PC software or teaching pendant.

(Example of 2-axis system)

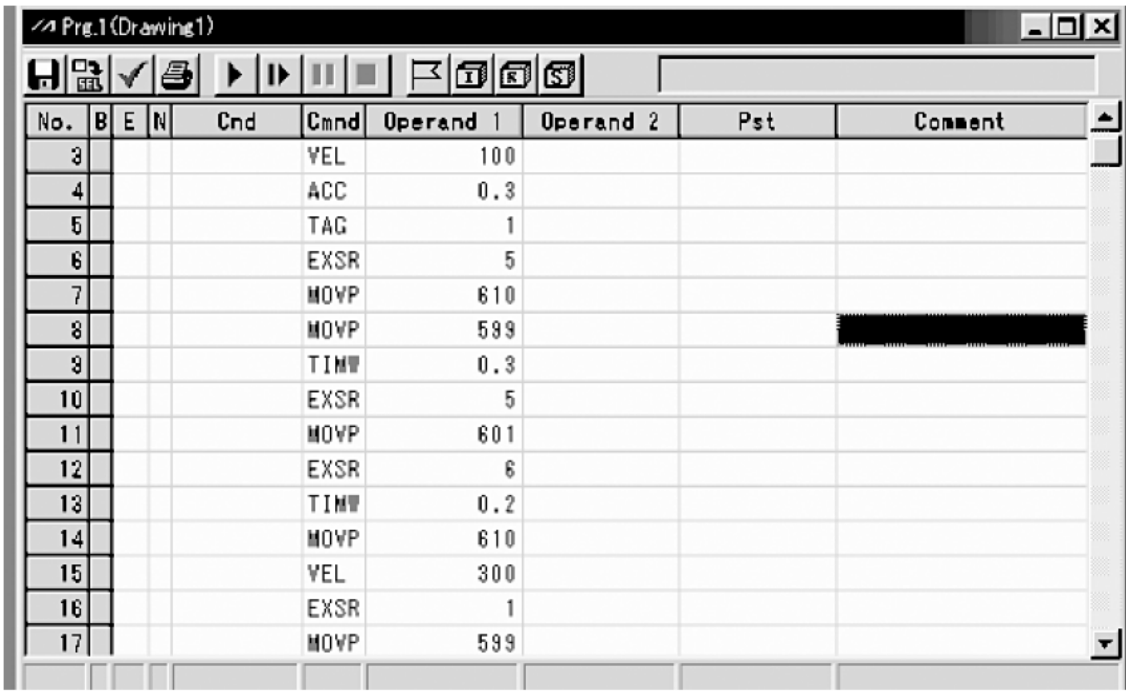
No.	Axis1	Axis2	Vel	Acc	Dcl
1	50.000	50.000			
2	100.000	30.000			
3	125.000	96.000			
4	75.000	102.000			
5	200.000	110.000			
6	150.500	116.000			
	⋮	⋮	⋮	⋮	⋮

No.: Specify a number, and the actuator will move to the position registered for the specified number in the program.
Axis 1 to Axis 2: Enter the target position of each axis for each position number.
Vel: Set a speed. The speed set in this field takes precedence over the speed specified in the program. In other words, the actuator uses the speed specified here when moving to the position specified for the corresponding position number.
Acc: Set an acceleration. The acceleration set in this field takes precedence over the acceleration specified in the program or one set by the applicable parameter.
Dcl: Set a deceleration. The deceleration set in this field takes precedence over the deceleration specified by the program or one set by the applicable parameter.

2. Programming Format

Program Edit Screen (PC Software)

The ASEL controllers support programs consisting of up to 2,000 steps.
Programs are edited using the PC software or teaching pendant.



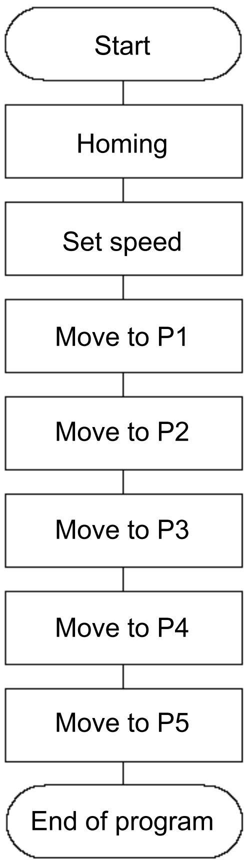
No.:	Step number
B:	Set a breakpoint (this field becomes editable during online edit). Click the “B” field in the line where you want to set a breakpoint. Once a breakpoint has been set, “B” is shown in the line. * Breakpoint --- A breakpoint is set in a step where you want to stop the program temporarily while the program is run from the PC software.
E:	Enter a desired extension condition (A, O, LD, AB or OB).
N:	Specify “N” to indicate negation of the input condition.
Cnd:	Enter an input condition.
Cmnd:	Enter a SEL command.
Operand 1:	Enter operand 1.
Operand 2:	Enter operand 2.
Pst:	Enter an output (operand 3).
Comment:	Enter a comment, if necessary (using up to 18 single-byte characters).

3. Positioning to Five Positions

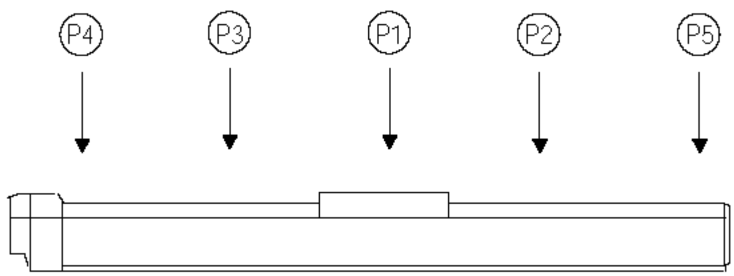
Description

Move the actuator to positions 1 through 5 at a speed of 100 mm/sec after homing.
Use of only 1 axis is assumed.

Flowchart



- Homing must be performed and a speed must be set, before the actuator can be operated.
- The actuator moves to the position data coordinates specified by the respective move commands.
- With the absolute specification, homing (HOME command) is not required.



Application program

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					HOME	1			Home Axis 1
2					VEL	100			Set velocity- mm/s
3					MOVL	1			Move to point 1
4					MOVL	2			Move to point 2
5					MOVL	3			Move to point 3
6					MOVL	4			Move to point 4
7					MOVL	5			Move to point 5
8					EXIT				End Program

Position data

No.	Axis1
1	100.000
2	150.000
3	50.000
4	0.000
5	200.000
6	
7	
8	
9	

4. How to Use TAG and GOTO

Description

Use GOTO and TAG commands to repeat the same operation within the program or to jump to a desired step if a condition is satisfied. A TAG command can be written in a step either before or after a GOTO command.

Example of Use 1

Repeat the same operation.

```
        :
        TAG    1
        MOVP   1  }
Repeated. MOVP 2  }  Steps to be repeated
        GOTO   1
```

Example of Use 2

Skip steps.

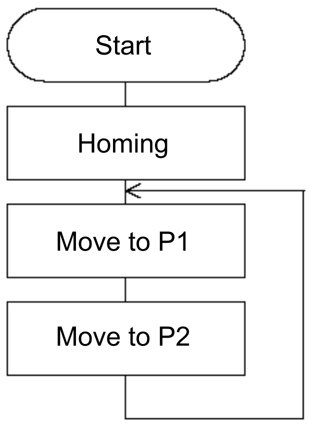
```
        :
        GOTO   2
Jump.   MOVP   1  }
        MOVP   2  }  Steps to be ignored
        TAG    2
```

5. Moving Back and Forth between Two Points

Description

Moves back and forth between two points.

Flowchart



- The actuator moves back and forth between P1 and P2 indefinitely.
- Use of only 1 axis is assumed.
- Enter TAG in the first of the steps to be repeated, and enter GOTO in the last of the steps to be repeated.

Application program

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					HOME	1			Home Axis 1
2					VEL	100			Set velocity- mm/s
3					TAG	1			Set loop marker 1
4					MOVL	1			Move to point 1
5					MOVL	2			Move to point 2
6					GOTO	1			Loop to TAG 1
7									

Position data

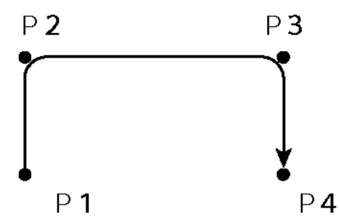
No.	Axis1
1	100.000
2	150.000
3	
4	
5	
6	

6. Path Operation

Description

Move continuously through four arbitrary points without stopping (PATH movement).

The actuator moves along the path shown at right, without stopping at P2 and P3.
Compared with MOVP and MOVL, this command does not require the actuator to position exactly at P2 and P3, and thus the movement tact time can be reduced.
Assume the following command is executed when the actuator is stopped at P1:
PATH 2 4
The actuator will move from P1 to P4 by passing points near P2 and P3. (The passing points can be brought closer to the specified positions by increasing the acceleration.)



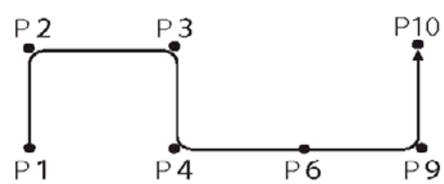
Even if "PATH 2 3" and "PATH 3 4" are input successively, the actuator will still move in the same way as when "PATH 2 4" is input.

If "PATH 4 1" is executed while the actuator is stopped at P4, the actuator will move along the same path in the opposite direction (P4 → P3 → P2 → P1).

Continuous movement through positions is possible even when the specified positions are not continuous.

    PATH 1 4
    PATH 6 6 (Discontinuous position)
    PATH 9 10

As shown above, specify each discontinuous position number, or position No. 6 in this case, as both the start position number and end position number in a PATH command. The axis will move through P1, P2, P3, P4, P6, P9 and P10 in this order.



7. Output Control during Path Movement

Description

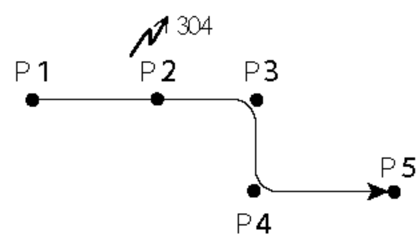
In spray operation, etc., output control may be required while the actuator is moving. The ASEL controller can output signals while the actuator is moving with a PATH command.

How to Use

Before executing a PATH command, declare a POTP command to specify signal output during movement. If a given output or global flag is specified in the output field of the PATH command, the output or flag specified in the output field will turn ON as the actuator approaches, via path movement, the position specified in the PATH command.

Example of Use 1

The actuator moves from P1 to P5 along the positions shown at right, without stopping. As the actuator approaches P2, output port 304 turns ON.



Cmnd	Operand 1	Operand 2	Pst
VEL	100		
POTP	1		
PATH	1	1	
PATH	2	2	304
PATH	3	5	

← A declaration command to specify signal output during path movement.
← 304 turns ON when the actuator approaches P2 specified in this step.

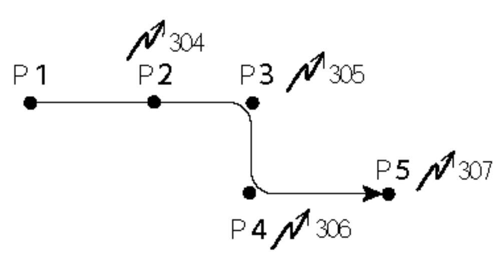
Outputs and flags can only be turned ON. The output or flag that was turned ON during path operation must be turned OFF (using a BTOF command) after the operation is completed.

Example of Use 2

Outputs 304 to 307 can be turned ON sequentially at the respective points of P2 to P5.

Cmnd	Operand 1	Operand 2	Pst
VEL	100		
POTP	1		
PATH	1	1	
PATH	2	5	304

← A declaration command to specify signal output during path movement.
← 304 to 307 turn ON sequentially at P2 to P5 specified in this step.



8. Circle/Arc Operation

Description

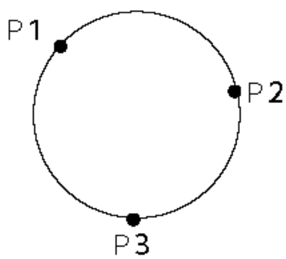
The actuator moves along a two-dimensional circle or arc.

How to Use

To specify a circle, specify three points the actuator will pass. To specify an arc, specify the starting point, passing point and end point.

Example of Use 1

Circle



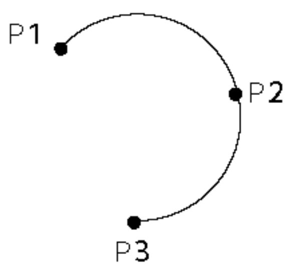
• Specify "CIR2 2 3" after the actuator has moved to P1.
• If "CIR2 2 3" is specified in the figure shown at left, the actuator will move along this circle clockwise.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			VEL	100		
			MOVP	1		
			CIR2	2	3	

• To cause the actuator to move counterclockwise, specify "CIR2 3 2."

Example of Use 2

Arc



• Specify "ARC2 2 3" after the actuator has moved to P1.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			VEL	100		
			MOVP	1		
			ARC2	2	3	

9. Home Return Completion Output

Description

Output a signal to confirm completion of homing (incremental specification).
With the ASEL controller, a home return completion signal can be output using an I/O parameter.
However, the following explains how to output a home return completion signal within a program using a general-purpose output.
Once turned ON, a general-purpose output will remain ON even after the current program ends or other program is started. (It will turn OFF upon emergency stop, etc., but the ON status can be maintained using an I/O parameter (I/O parameter Nos. 70 and 71).)

Example of Use

a. Output a home return completion signal.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	
			HOME	11			Execute homing.
			BTON	303			General-purpose output (arbitrary)

b. Use a home return completion signal to make sure the actuator will not perform homing if it has already been performed.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	
	N	303	HOME	11			Execute homing if output 303 is OFF.
			BTON	303			Home return completion output

c. Use the output field instead of a BTON command.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	
	N	303	HOME	11		303	Execute the same processing performed with the above two steps.

Reference

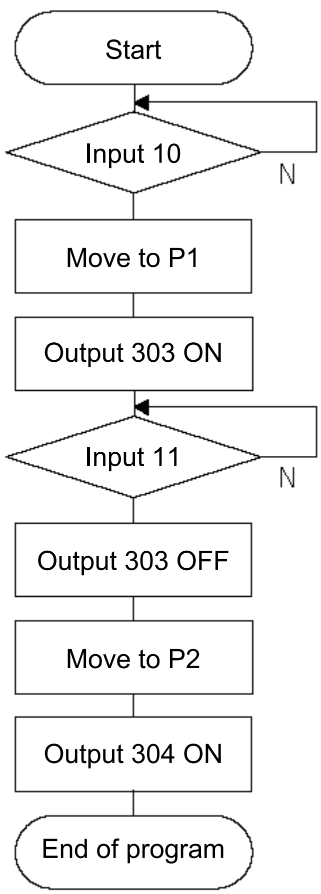
Output port No. 304 can be used as a home return completion output (dedicated output) by setting I/O parameter No. 50 to “13.”

10. Axis Movement by Input Waiting and Completion Output

Description

How to perform input waiting and output a processing completion signal is explained.

Flowchart



Example of Use

The actuator waits until input port 10 turns ON, and then moves to P1.
The actuator waits until input port 11 turns ON, and then moves to P2.
A movement completion signal is output from 303 upon reaching P1, and from 304 upon reaching P2.

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			Set velocity- mm/s
			WTON	10			Wait on input 10
			MOVP	1			Move to point 1
			BTON	310			Turn ON output 310
			WTON	11			Wait on input 11
			BTOF	310			Turn OFF outpt 310
			MOVP	2			Move to point 2
			BTON	311			Turn ON output 310
			EXIT				End Program

11. Changing the Moving Speed

Description

Change the moving speed.

How to Use

With the ASEL controller, the speed can be set using the following two methods:
a: Use a VEL command within the application program
b: Use a speed setting in the position data table

Example of Use

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			MOVP	1		
			VEL	1000		
			MOVP	2		
			MOVP	3		
			VEL	50		
			MOVP	4		

Position data

No.	Axis1	Vel	Acc	Dcl
1	100.000	100		
2	200.000	500		
3	300.000			
4	400.000			

Moving speeds in the above program

Position at 100 mm --- The actuator moves at 100 mm/sec.
Position at 200 mm --- The actuator moves at 500 mm/sec.
Position at 300 mm --- The actuator moves at 1000 mm/sec.
Position at 400 mm --- The actuator moves at 50 mm/sec.

If a speed is specified in the position data table, this speed takes precedence over the speed specified in the application program, as shown above. In general, speeds are set in the application program using VEL.

Vel in Point Data Table and PATH Command

The speed can be changed without stopping the actuator, by using a PATH command and Vel in the position data table. (Refer to the next page.)

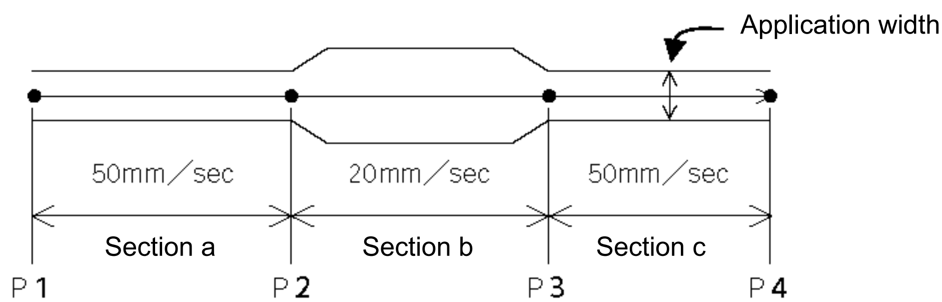
12. Changing the Speed during Operation

Description

Use a PATH command to change the speed while the actuator is moving.
For example, this command is useful in a paint dispensing application where the application volume changes in the middle.

Example of Use

The actuator moves through linear sections a, b and c at 50 mm/sec, 20 mm/sec and 50 mm/sec, respectively, without stopping (PATH movement).



Position data

No.	Axis1	Vel	Acc	Dcl
1	0.000	50		
2	100.000	50		
3	200.000	20		
4	300.000	50		

Application program

“PATH 1 4” is the only movement command required.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			PATH	1	4	

Reference

The speed can also be changed from other program using a CHVL (speed change) command (in the multi-tasking mode).

13. Local/Global Variables and Flags

Description

The internal variables and flags used in the SEL language are classified into local and global types.
The data range used commonly by all programs is called the global range, while the data range used only by each program is called the local range. When multi-tasking programs are run simultaneously, the global range must be used to synchronize the programs and allow cross-referencing of variables among the programs.

Example of Use

Program handshake

Program A

Cmnd	Operand 1
MOVL	1
BTON	600
WTON	601
MOVL	3

Program B

Cmnd	Operand 1
WTON	600
MOVL	2
BTON	601

Use of global flags with the above two programs permits handshake between the programs, and the actuator moves per “MOVL 1” in program A, moves per “MOVL 2” in program B, and then move per “MOVL 3” in program A, for example.

Backup in Battery

If the ASEL controller has a built-in battery (optional), variables and flags used in the programs are retained. For both variables and flags, only those in the global range will be retained after the controller power is turned off.
The variables and flags in the local range are cleared when the program is started (the variables are reset to “0,” while the flags turn OFF).

14. How to Use Subroutines

Description

A subroutine is a group of steps that are called and executed several times within a program. Subroutines are used to reduce the number of program steps and make the program easy to read. Up to 99 subroutines can be used in one program. Up to 15 subroutine calls can be nested.

How to Use

Declare/call subroutines using the following commands:
EXSR: Call a subroutine
BGSR: Declare the start of a subroutine (start of a group of steps)
EDSR: Declare the end of a subroutine (end of a group of steps)

Example of Use

Cmnd	Operand 1
VEL	100
MOVL	1
BTON	303
WTON	20
BTOF	303
MOVL	2
BTON	303
WTON	20
BTOF	303
MOVL	3
BTON	303
WTON	10
BTOF	303
EXIT	

The same tasks are consolidated into a single location.

Cmnd	Operand 1
VEL	100
MOVL	1
EXSR	1
MOVL	2
EXSR	1
MOVL	3
EXSR	1
EXIT	
BGSR	1
BTON	303
WTON	10
BTOF	303
EDSR	

Subroutine

Caution

Jumping from within a subroutine to a TAG position outside the subroutine using a GOTO command is prohibited.

15. Pausing the Operation

Description

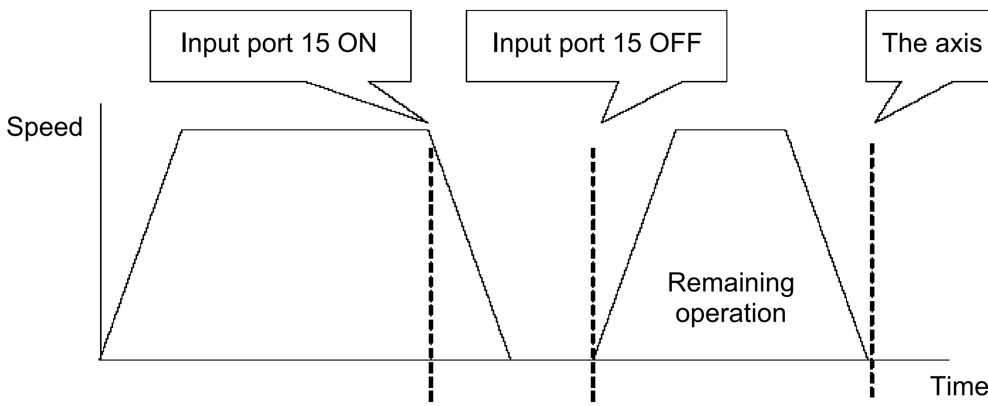
Use a declaration command HOLD to pause the moving axis temporarily via external input.

How to Use

A pause interruption operation can be executed to a moving axis (to decelerate the axis to a stop) by declaring a HOLD command within the program.
While HOLD is input, the actuator pauses (decelerates to a stop, if currently moving) against all moving commands in the same program.

Example of Use

HOLD 15 A declaration to execute pause if general-purpose input 15 turns ON.



Application

You can specify a global flag, instead of an input port, in Operand 1 of the HOLD command.
Use of a global flag allows the actuator to be paused from other program.
The input signal pattern and stop action can be selected using Operand 2.

0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)
2 = Contact b (Decelerates to a stop, and then servo OFF ⇒ The drive power is not cut off.)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			HOLD	20	2		SVOF when input 20

Caution

If the actuator is paused during homing, it will start the homing sequence from the beginning upon restart.

16. Canceling the Operation 1 (CANC)

Description

Use a declaration command CANC to decelerate the moving axis to a stop and cancel the remaining operation.

How to Use

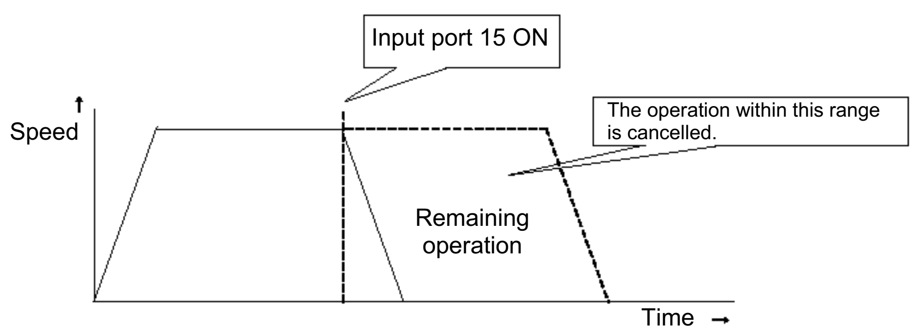
While CAN is input, all movement commands in the same program are cancelled.

Example of Use

CANC command

```
CANC   15        Cancel the movement commands if input port 15 turns ON (declaration).
   :
MOVP    1
MOVP    2
   :
WTON   14
   :
```

* Declare this command in a step before the movement commands you want to cancel.
* While CANC is input, all operation commands are cancelled sequentially, while tasks other than operation commands (such as I/O processing and calculation processing) are executed sequentially.



Caution

Since execution of this command makes it no longer possible to specify which program step is currently executed, it is recommended that a WTON command be used to create an input wait step.

Application

A desired input signal pattern can be selected for a CANC command using Operand 2.

0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			CANC	20	1		Halt when input 20

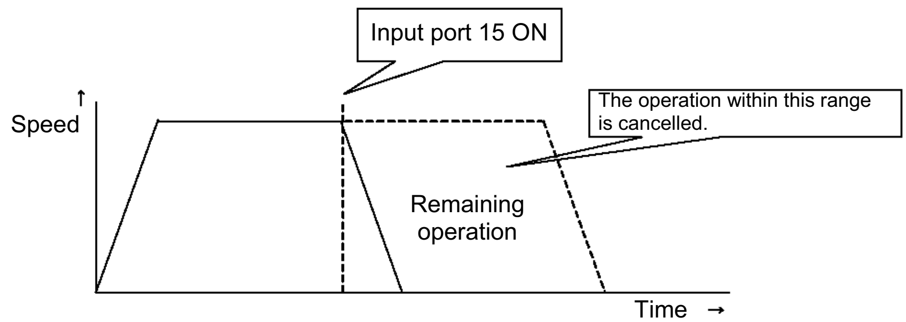
17. Canceling the Operation 2 (STOP)

Description

Decelerate the moving axis to a stop and cancel the remaining operation. (STOP)

How to Use

Execute a STOP command from other program to forcibly stop the operation (in the multi-tasking mode).
Specify the axis you want to stop using an axis pattern.



Example of Use 1

STOP command

Main program

```
EXPG    n    The stop program starts.
  :
MOVL    1
MOVL    2
  :
```

Stop control program

```
WTON    15    Wait for stop input.
STOP    11    Axes 1 and 2 stop.
```

If “STOP 11” is executed while “MOVL 1” is being executed, “MOVL 1” will be cancelled and the actuator will continue its operation from “MOVL 2.”

Example of Use 2

Main program

```
EXPG    n    The stop program starts.
  :
MOVP    1
MOVP    2
  :
```

Stop control program

```
WTON    15    Wait for stop input.
STOP    10    Axis 2 stops.
```

If “STOP 10” is executed while “MOVL 1” is being executed, only the axis 2 part of “MOVL 1” will be cancelled. Both axes 1 and 2 will operate under “MOVL 2.”

Caution

If a STOP command is executed during a CP operation (interpolation operation) initiated by MOVL, etc., the operations of all axes will be cancelled regardless of the axis pattern specified in the STOP command.

18. Movement by Position Number Specification

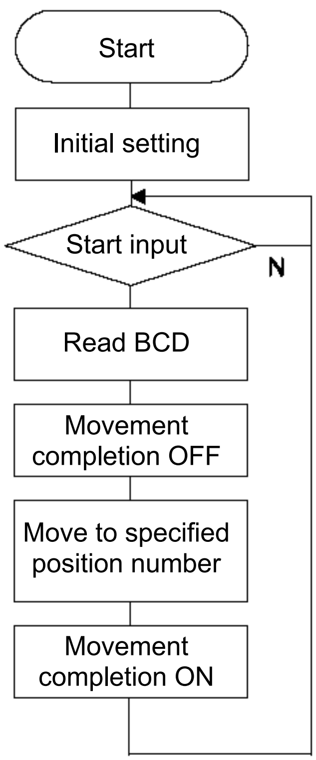
Description

Load externally input BCD codes as position numbers to execute movements.

Example of Use

Use an INB command to load a position number as a BCD code from an input port.
A position number can be specified using a value consisting of up to three digits.

Flowchart



Input assignment

Port	Description
1	Start input
2	Position specification 1
3	Position specification 2
4	Position specification 4
5	Position specification 8
6	Position specification 10
7	Position specification 20
8	Position specification 40
9	Position specification 80
10	Position specification 100
11	Position specification 200
12	Position specification 400
13	Position specification 800

Output

303 Movement completion

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			HOME	11			Home axis 1 & 2
			VEL	100			Set velocity- mm/s
			TAG	1			Set loop marker 1
			WTON	1			Wait on start inpt
			INB	15	3		Read position #
			BTOF	303			Mov cmplt sgnl OFF
			MOVL	*99			Move to position
			BTON	303			Move cmplt sgnl ON
			GOTO	1			Jump to marker 1 ^

19. Movement by External Position Data Input

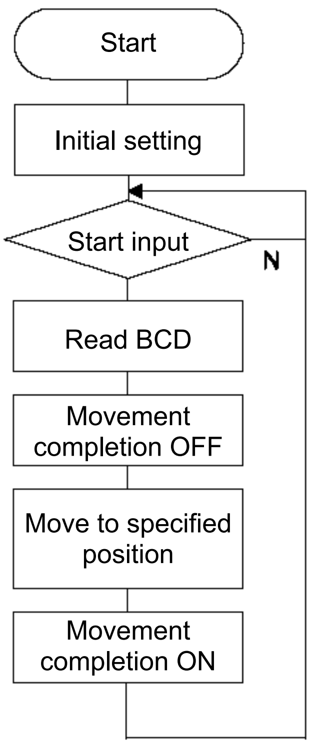
Description

Receive target position data as absolute values from a host device to execute movements.

Example of Use

Use an INB command to load position data as a BCD code from an input port.
Each BCD value should consist of four digits, with the last digit indicating a decimal place.
The moving axis is axis 1.
Example: If a BCD of “1234” is received, the axis will move to the position at 123.4 mm.
Note: When using input port Nos. 16 and 17, do so after changing them to general-purpose inputs.

Flowchart



Input assignment

Port	Description
1	Start input
2	0.1mm
3	0.2mm
4	0.4mm
5	0.8mm
6	1mm
7	2mm
8	4mm
9	8mm
10	10mm
11	20mm
12	40mm
13	80mm
14	100mm
15	200mm
16	400mm
17	800mm

Output

303 Movement completion

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			HOME	11			Home axis 1 & 2
			VEL	100			Set velocity- mm/s
			TAG	1			Set loop marker 1
			WTON	1			Wait on start inpt
			INB	15	4		Read position #
			LET	199	*99		Indirect refernce
			DIV	199	10		Div by 10 resolutn
			PPUT	1	1000		Put 1K for axis 1
			BTOF	303			Mov cmplt sigl OFF
			MOVL	1000			Move to entry pos.
			BTON	303			Mov cmplt signl ON
			GOTO	1			Jump to marker 1 ^

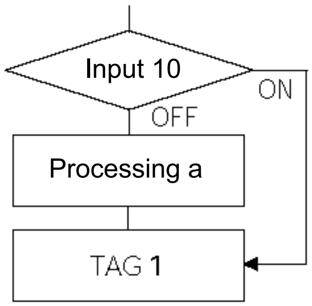
20. Conditional Jump

Description

Select the destination to jump to via GOTO using the external input, output and/or internal flag statuses as a condition.
The controller waits for multiple inputs, and performs processing according to the received input(s).

Example of Use 1

If input 10 turns ON, the actuator will jump to TAG 1. If it turns OFF, the actuator will proceed to the next processing.



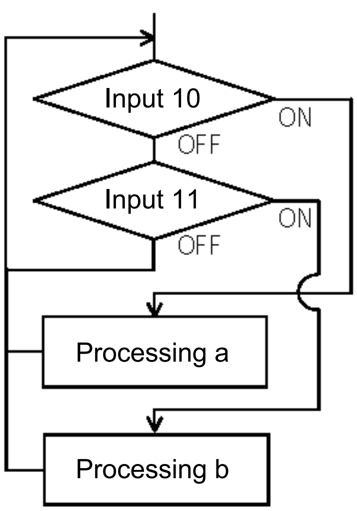
E	N	Cnd	Cmnd	Operand 1
		10	GOTO	1
		Processing a		
			TAG	1
		Processing b		

Execute GOTO 1 if input 10 turns ON.

* If input 10 turns ON, the actuator will skip processing a and perform processing b.
If input 10 turns OFF, the actuator will perform processing a, and then perform processing b.

Example of Use 2

The controller waits for an input signal to be received at input port 10 or 11. If an input signal is received at input 10, the actuator will perform processing a. If an input signal is received at input 11, it will perform processing b.



E	N	Cnd	Cmnd	Operand 1
			TAG	1
		10	GOTO	2
		11	GOTO	3
			GOTO	1
			TAG	2
		Processing a		
			GOTO	1
			TAG	3
		Processing b		
			GOTO	1

No input.
Input 10 turns ON.
Input 11 turns ON.

If both inputs 10 and 11 turn ON, the actuator will perform processing a.

21. Waiting Multiple Inputs

Description

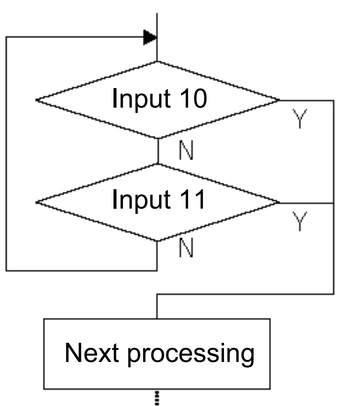
The controller waits for multiple different inputs and performs processing upon reception of any of these inputs.

Point

A WTON command permits processing only when the specified input is received. The controller cannot wait for multiple inputs.

Example of Use

Inputs 10 and 11 are monitored, and the actuator will proceed to the next step when either input is received (OR logic).



Program a

E	N	Cnd	Cmnd	Operand 1
			TAG	1
		10		
O		11	GOTO	2
			GOTO	1
			TAG	2

Next processing

Program b

E	N	Cnd	Cmnd	Operand 1
			TAG	1
	N	10		
A	N	11	GOTO	1

Next processing

* Both programs a and b perform the same processing.

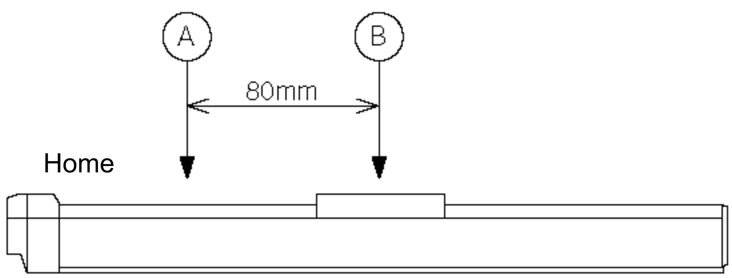
As shown in the sample, the controller waits for input without using a WTON command.
This method can also be used when multiple input conditions must be combined.

22. How to Use Offset

Description

With an OFST command, an offset can be specified for position data when you want to shift (offset) all teaching points by several millimeters because the actuator was not installed exactly in the specified position or for other reasons.
An OFST command can also be used to perform pitch feed. (Refer to 24, "Constant-pitch Feed.")

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			Set velocity- mm/s
			MOVP	1			Move to point 1
			OFST	1	80		Offset axis 1 80mm
			MOVP	1			Move to point 1



Caution

Once an offset has been set, the offset applies to all movement commands executed thereafter. To cancel the offset, execute an offset command again by specifying "0" mm. An offset does not apply to other programs (even in the multi-tasking mode). If a given offset must be applied to all programs, it must be set for all programs individually.

23. Executing an Operation N times

Description

Execute a specific operation n times.

Example of Use

The actuator moves back and forth between P1 and P2 ten times, and then the program ends.
Use a CPEQ command to compare the number of times the movement has been actually repeated, against 10.
It is assumed that homing has been completed.

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			Set velocity- mm/s
			LET	1	0		Initlz counter 1
			TAG	1			Set loop marker 1
			MOVP	1			Move to point 1
			MOVP	2			Move to point 2
			ADD	1	1		Incrmt cntr by 1
			CPEQ	1	10	900	Repeat 10 times
	N	900	GOTO	1			Loop if not done
			EXIT				Else end program

Reference

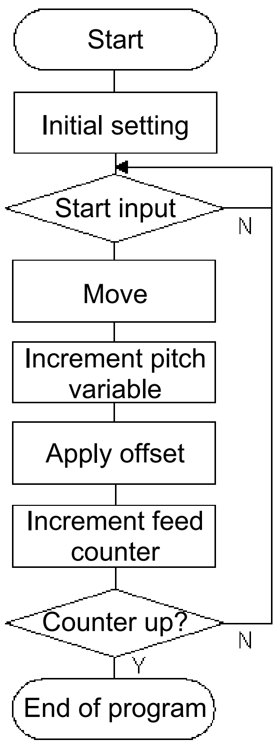
The same operation can also be performed using a DWEQ command.

24. Constant-pitch Feed

Description

Feed the actuator by a specified pitch n times from a reference point.
The pitch and number of repetitions are specified by variables in advance.

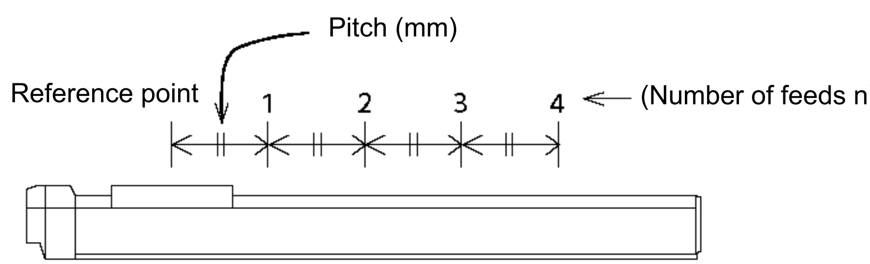
Flowchart



Example of Use

Use an OFST command to perform pitch feed.
The number of times the actuator has been fed is counted by a counter variable.
The X-axis is fed in the positive direction.

Point
An OFST command applies to movement commands.
Executing an OFST command alone does not move the axis.



Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			LET	1	4		Variable feed #= 4
			LET	100	80		Var. pitch = 80mm
			LET	2	0		Clear counter 2
			LET	101	0		Initializ var. 101
			HOME	1			Home axis 1
			VEL	100			Set velocity- mm/s
			TAG	1			Set loop marker 1
			WTON	1			Wait on start inpt
			MOVP	1			Move to point 1
			ADD	101	*100		Add pitch to offst
			OFST	1	*101		Process x offset
			ADD	2	1		Add 1 to counter 2
			CPGT	2	*1	900	Confirm feed cmplt
	N	900	GOTO	1			Repeat if needed
			EXIT				End Program

Reference

Pitch feed can also be performed using a MVPI or MVLI command.

25. Jogging

Description

The slider moves forward or backward while an input is ON or OFF.
Instead of an input, an output or global flag can be used as a cue.
The slider will move directly to the next step if the specified input does not satisfy the condition when the command is executed.
Regardless of the input status, the slider will stop upon reaching the soft limit, and the command in the next step will be executed.

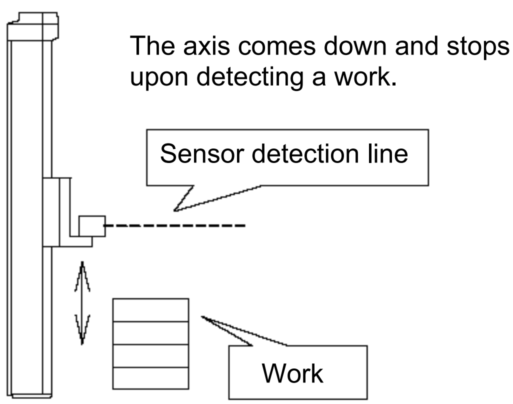
How to Use

• Explanation of commands

JFWN	1	1	Axis 1 moves forward while input 1 is ON.
JFWF	1	2	Axis 1 moves forward while input 2 is OFF.
JBWN	10	3	Axis 2 moves backward while input 3 is ON.
JBWF	10	4	Axis 2 moves backward while input 4 is OFF.

Example of Use 1

• Stop the axis when a sensor input is received.



:

VEL	50		Specify a low speed.
JFWF	1	10	Move until a sensor input (10) is received.
EXIT			The program ends.

Example of Use 2

• Cause the actuator to jog just like in teaching pendant operation (2 axes are operated).

Application program

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			TAG	1		
			JFWN	1	1	
			JBWN	1	2	
			JFWN	10	3	
			JBWN	10	4	
	N	24	GOTO	1		
			EXIT			

Reference

HOLD, STOP and CANC commands remain valid while the actuators are jogging.

26. Switching Programs

Description

Use EXPG/ABPG commands to switch programs using a program.

Example of Use 1

Start program 2 once the processing of program 1 is completed, and then end program 1.

Program 1	Program 2
:	:
EXPG 2	:
EXIT	

Example of Use 2

Start a program via an external signal, and then end the other program.

Program 1	Program 2
ABPG 2	ABPG 1
:	:

If program 2 is started while program 1 is running, program 1 will be aborted.
If program 1 is started while program 2 is running, program 2 will be aborted.

Application

If a program number is specified in operand 2, the programs from the one corresponding to the program number in operand 1 to the other corresponding to the program number in operand 2 can be started (EXPG) or ended (ABPG) simultaneously.

Caution

- The ASEL controller supports multi-tasking. Up to 8 programs can be run at the same time. To use other programs when the controller is already running 8 programs, switch programs by closing a program or programs that are not required.
- If an ABPG command was executed to end a program while the program was executing a movement command, the actuator immediately decelerates to a stop.

27. Aborting a Program

Description

Abort a program currently running.
Execute an ABPG command (command to abort other program) from other program in the multi-tasking mode.

Caution

* If the target program was executing a movement command, the actuator immediately decelerates to a stop and the program ends.

Example of Use

Main program (Prg. 1)

EXPG	n	The abort control program starts.
WTON	10	
MOVP	1	
BTON	303	
:		
:		

Abort control program (Prg. n)

WTON	10	Wait for an abort input.
ABPG	1	Prg. 1 is aborted.
EXIT		The program ends.

* If ABPG was executed while the actuator was moving via a MOVP command, the actuator immediately decelerates to a stop and the program ends.

Part 3 Positioner Mode

In the positioner mode, position data is input in the MANU mode and positioning operation based on input data is performed in the AUTO mode (the controller modes are switched using the AUTO/MANU switch).
If the controller mode is changed to MANU while positioning is performed in the AUTO mode, the controller will maintain the servo ON or OFF status that was effective prior to the mode change. The output conditions of ready/alarm status and absolute-data/system battery error status will be retained. All other outputs will be turned OFF.
When the controller is returned to the AUTO mode in this condition, the outputs will also return to their original conditions.

Chapter 1 Modes and Signal Assignments

The positioner mode provides five sub-modes associated with different PIO (parallel I/O) patterns. Select a mode appropriate for your specific purpose.
To select a desired mode, set a number between 1 and 4 or 16 in other parameter No. 25, "Operation mode type."

1. Feature of Each Mode

Value set in parameter No. 25	Feature of each mode
1	Standard mode Positioning to up to 1,500 positions can be performed. Push-motion operation is also supported.
2	Product switching mode Product numbers can be set in addition to position numbers. A position number can be changed for each product under the same position number. Push-motion operation is also supported.
3	2-axis independent mode Operations of two axes (start/stop) can be controlled separately.
4	Teaching mode Positions to be registered can be taught externally.
16	DS-S-C1 compatible mode This mode reflects the operation of the DS-S-C1 controller by adopting compatible pin assignments. Replacement without any modification is possible.

2. Number of Positions Supported in Each Mode

Mode	Number of positions
Standard mode	Maximum 1,500 positions
Product switching mode	Total 1,500 positions for all products (The same number of position data sets is used for each product.)
2-axis independent mode	13 input bits are divided into position-number input bits for axis 1 and position-number input bits for axis 2.
Teaching mode	Maximum 1,500 positions
DS-S-C1 compatible mode	Maximum 1,500 positions

Note) Two sets of position data are needed for push-motion operation. (Push-motion operation can be performed only in the standard mode and product switching mode.)

3. Quick Mode Function Reference Table

		Other parameter No. 25				
		1	2	3	4	16
I/O	Function	Standard mode	Product switching mode	2-axis independent mode	Teaching mode	DS-S-C1 compatible mode
Input	Push-motion operation	O	O	x	x	x
	Error reset	O	O	O	O	x
	CPU reset	x	x	x	x	O
	Home return	O	O	O	Note 1	Note 2
	Servo ON	O	O	O	O	x
	Cancellation	O	O	O	x	O
	Interpolation	O	O	x	x	O
	Jog	x	x	x	O	x
Output	Home return complete	O	O	O	O	x
	Servo ON output	O	O	O	O	x
	System battery error	O	O	x Note 3	O	O
	Absolute battery error	O	O	x Note 4	O	O

Note 1) In the teaching mode, home return will be performed when the start signal is input after specifying a desired position number in a condition where home return is not yet complete.
Note 2) In the DS-S-C1 compatible mode, home return will be performed when the start signal is input after specifying position No. 0.
Note 3) In the 2-axis independent mode, a system-battery voltage low warning will not be output. In this mode, it is recommended not to back up the position data and error list using the battery (not to use the optional system-memory backup battery).
Note 4) In the 2-axis independent mode, an absolute-data backup battery low warning will not be output. If your system operates in this mode, use incremental actuators.

4. Interface List of All PIO Patterns

Pin No.	Category	Port No.	Positioner mode					Cable color
			Standard mode	Product switching mode	2-axis independent mode	Teaching mode	DS-S-C1 compatible mode	
1A	P24		24-V input					1-Brown
1B	Input	16	Position input 10	Input 10	Position input 7	Axis 1 jog-	Position No. 1000 input	1-Red
2A		17	Position input 11	Input 11	Position input 8	Axis 2 jog+	-	1-Orange
2B		18	Position input 12	Input 12	Position input 9	Axis 2 jog-	-	1-Yellow
3A		19	Position input 13	Input 13	Position input 10	Inching (0.01 mm)	-	1-Green
3B		20	-	Input 14	Position input 11	Inching (0.1 mm)	-	1-Blue
4A		21	-	Input 15	Position input 12	Inching (0.5 mm)	-	1-Purple
4B		22	-	Input 16	Position input 13	Inching (1 mm)	-	1-Gray
5A		23	Error reset	Error reset	Error reset	Error reset	CPU reset	1-White
5B		0	Start	Start	Axis 1 start	Start	Start	1-Black
6A		1	Home return	Home return	Home return	Servo ON	Pause	2-Brown
6B		2	Servo ON	Servo ON	Axis 1 servo ON	*Pause	Cancellation	2-Red
7A		3	Push motion	Push motion	*Axis 1 pause	Position input 1	Interpolation setting	2-Orange
7B		4	*Pause	*Pause	*Axis 1 cancellation	Position input 2	Position No. 1 input	2-Yellow
8A		5	*Cancellation	*Cancellation	Axis 2 start	Position input 3	Position No. 2 input	2-Green
8B		6	Interpolation	Interpolation	Axis 2 home return	Position input 4	Position No. 4 input	2-Blue
9A		7	Position input 1	Input 1	Axis 2 servo ON	Position input 5	Position No. 8 input	2-Purple
9B		8	Position input 2	Input 2	*Axis 2 pause	Position input 6	Position No. 10 input	2-Gray
10A		9	Position input 3	Input 3	*Axis 2 cancellation	Position input 7	Position No. 20 input	2-White
10B		10	Position input 4	Input 4	Position input 1	Position input 8	Position No. 40 input	2-Black
11A		11	Position input 5	Input 5	Position input 2	Position input 9	Position No. 80 input	3-Brown
11B		12	Position input 6	Input 6	Position input 3	Position input 10	Position No. 100 input	3-Red
12A		13	Position input 7	Input 7	Position input 4	Position input 11	Position No. 200 input	3-Orange
12B		14	Position input 8	Input 8	Position input 5	Teaching mode specification	Position No. 400 input	3-Yellow
13A		15	Position input 9	Input 9	Position input 6	Axis 1 jog+	Position No. 800 input	3-Green
13B	Output	300	*Alarm	*Alarm	*Alarm	*Alarm	Alarm	3-Blue
14A		301	Ready	Ready	Ready	Ready	Ready	3-Purple
14B		302	Positioning complete	Positioning complete	Axis 1 positioning complete	Positioning complete	Positioning complete	3-Gray
15A		303	Home return complete	Home return complete	Axis 1 home return complete	Home return complete	-	3-White
15B		304	Servo ON output	Servo ON output	Axis 1 servo ON	Servo ON output	-	3-Black
16A		305	Push motion complete	Push motion complete	Axis 2 positioning complete		-	4-Brown
16B		306	System battery error	System battery error	Axis 2 home return complete	System battery error	System battery error	4-Red
17A		307	Absolute battery error	Absolute battery error	Axis 2 servo ON	Absolute battery error	Absolute battery error	4-Orange
17B	N		0-V input					4-Yellow

*: Contact B (always ON)

Chapter 2 Standard Mode

The standard mode provides a PIO pattern of greatest general utility among all positioner modes accessible in the ASEL controller.

1. I/O Interface List

Pin No.	Category	Port No.	Signal name	Signal symbol	Function overview	Cable color
1A	P24		External power supply 24 V	P24		1-Brown
1B	Input	016	Position input 10	PC10	(Same as position inputs 1 through 9)	1-Red
2A		017	Position input 11	PC11		1-Orange
2B		018	Position input 12	PC12		1-Yellow
3A		019	Position input 13	PC13		1-Green
3B		020	-			1-Blue
4A		021	-			1-Purple
4B		022	-			1-Gray
5A		023	Error reset	RES	Present alarms will be reset at the leading edge of this signal.	1-White
5B		000	Start	CSTR	The actuator will start moving at the leading edge of this signal.	1-Black
6A		001	Home return	HOME	The actuator will start home-return operation at the leading edge of this signal.	2-Brown
6B		002	Servo ON	SON	The servo will remain on while this signal is ON, and remain off while this signal is OFF.	2-Red
7A		003	Push motion	PUSH	The actuator will start linear interpolation operation if the start input signal is turned ON while this signal is ON.	2-Orange
7B		004	*Pause	*STP	The actuator can be moved when this signal is ON, and will decelerate to a stop when the signal turns OFF.	2-Yellow
8A		005	*Cancellation	*CANC	The remaining travel distance will be cancelled if this signal turns OFF.	2-Green
8B		006	Interpolation	LINE	With the 2-axis specification, linear interpolation operation will start when the start input signal is turned ON while this signal is ON.	2-Blue
9A		007	Position input 1	PC1	Input the position number corresponding to the position you want to move the actuator to. Be sure to specify a position input by no later than 6 msec before the start input signal turns ON. Position numbers are input as binary codes (factory setting). The input mode can be changed to BCD by changing the setting of other parameter No. 71. (PC1 through 4 indicate the one's place, PC5 through 8 indicate ten's place, PC9 through 12 indicate the hundred's place, and PC13 indicates the thousand's place.)	2-Purple
9B		008	Position input 2	PC2		2-Gray
10A		009	Position input 3	PC3		2-White
10B		010	Position input 4	PC4		2-Black
11A		011	Position input 5	PC5		3-Brown
11B		012	Position input 6	PC6		3-Red
12A		013	Position input 7	PC7		3-Orange
12B		014	Position input 8	PC8		3-Yellow
13A		015	Position input 9	PC9		3-Green
13B	Output	300	*Alarm	*ALM	This signal remains ON if the controller is normal. It will turn OFF if an alarm occurs.	3-Blue
14A		301	Ready	RDY	This signal will turn ON when the controller becomes ready.	3-Purple
14B		302	Positioning complete	PEND	This signal will turn ON once the actuator has moved to the target position and entered the positioning band.	3-Gray
15A		303	Home return complete	HEND	This signal is OFF when the power is input, and will turn ON when home return is completed.	3-White
15B		304	Servo ON output	SVON	This signal will turn ON when the servo is turned on, and turn OFF when the servo is turned off.	3-Black
16A		305	Push motion complete	PSED	This signal will turn ON when the push-motion operation is completed successfully, and turn OFF if the work is mixed.	4-Brown
16B		306	System battery error	SSER	This signal will turn ON when the voltage of the system-memory backup battery drops to the voltage-low warning level.	4-Red
17A		307	Absolute battery error	ABER	This signal will turn ON when the voltage of the absolute-data backup battery drops to the voltage-low warning level.	4-Orange
17B	N		External power supply 0 V	N		4-Yellow

*: Contact B (always ON)

2. Parameters

To use the controller in the standard mode, set other parameter No. 25 to "1."
Position numbers are specified as binary codes according to the factory setting. To change the input mode to BCD, set a value "other than 0" in other parameter No. 25.

	No.	Parameter	Function
Other	25	Operation mode type	1: Standard mode
	71	Positioner mode parameter 1	Position-number input mode specification (0: Binary, ≠ 0: BCD) * Default value: 0 (Binary)

3. Details of Each Input Signal

■ Start (CSTR)
When the OFF → ON leading edge of this signal is detected, the controller will load the target point number specified by the 13-bit binary code consisting of PC1 through PC13, and perform positioning to the target position specified by the corresponding position data.
Before movement is started, the target position, speed and other operation data must be set in the position table using a PC or teaching pendant.
If this signal is input when no single home-return operation has been performed after the power was input (= when the HEND output signal is OFF), "C6F, Home-return incomplete error" will generate.

■ Command position number (PC1 through PC13)
When a movement command is executed upon the OFF → ON edge of the start signal, the controller will load the command position number specified by the 13-bit binary code consisting of signals PC1 through PC13.
The weight of each bit is as follows: 2^0 for PC1, 2^1 for PC2, 2^2 for PC3, 2^3 for PC4, 2^4 for PC5, ..., and 2^{10} for PC11. By combining these bits, any position number between 0 and 1500 (maximum) can be specified.
The input mode can be changed to BCD by changing the setting of other parameter No. 71, as follows:
Other parameter No. 71 = 1 (BCD input)
(Default setting of other parameter No. 71 = 0 (Binary input))
In the BCD input mode, PC1 through 4 indicate the one's place, PC5 through 8 indicate ten's place, PC9 through 12 indicate the hundred's place, and PC13 indicates the thousand's place.

■ Pause (*STP)
If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be held, which means that when the signal turns ON again, the actuator will resume movement of the remaining travel distance.
To cancel the movement command altogether after turning OFF the pause signal, turn OFF the cancellation signal while this signal is OFF to cancel the remaining travel distance.
The pause signal can be used for the following purposes:
[1] As a sensor to detect entry into a specified area around the system or for other lower-level safety measures to stop the axis while the servo is on
[2] To prevent contact with other equipment
[3] For positioning based on sensor or LS signal detection
(Note) When this signal is input during home return, the movement command will be held if the actuator has not yet contacted the mechanical end. If the signal is input after the actuator has reversed upon contacting the mechanical end, home return will be performed again.

■ Cancellation (*CANC)
If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop. The remaining travel distance will be cancelled and the movement will not resume even when the signal turns ON thereafter.

■ Home return (HOME)
The actuator will start home-return operation upon detection of the OFF → ON edge of this signal.
Once the home return is complete, the HEND signal will be output. This signal can be input as many times as desired after completion of the initial home return.
(Note) An actuator of incremental specification must always perform home return after the power is turned on.

■ Servo ON (SON)
The servo remains on while this signal is ON.
To operate the actuator using the start input/home return input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. (Only the commands will be ignored, and no error will generate.)
(Note) When this signal turns OFF while the actuator is moving, the actuator will not decelerate to a stop. It will complete the movement to the target position, after which the servo will turn off.

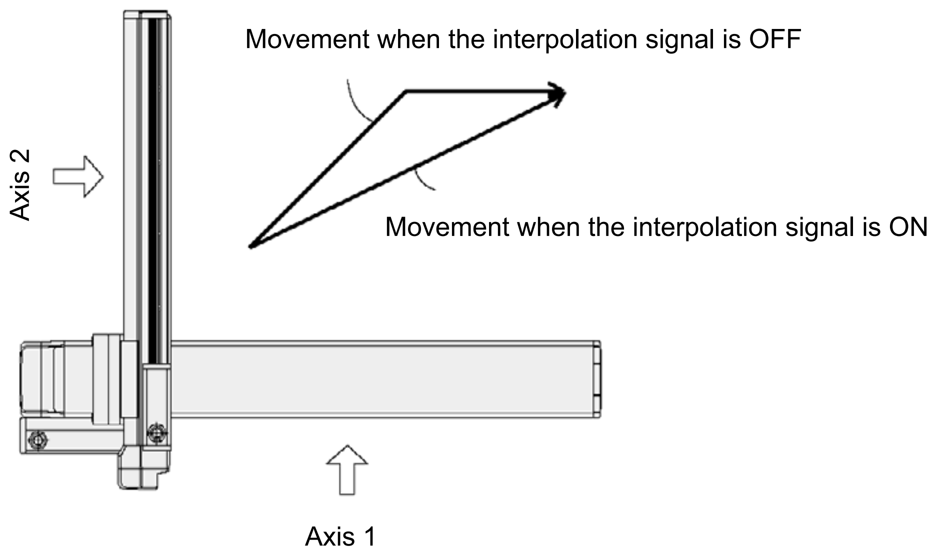
⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
Put the slider or rod away from the mechanical end when turning the servo ON.

■ Error reset (RES)
This signal is used to reset the alarm output signal (*ALM) that has been generated due to an error.
If an error occurred, check the content of the error and then turn this signal ON.
The error will be reset upon detection of the leading edge of the signal.
(Note) Errors of cold start and higher level cannot be reset using this signal. The power must be reconnected to reset these errors. For details, refer to Appendix, "Error Level Management."

■ Push motion (PUSH)
The actuator will perform push-motion operation if the position signal and start signal are input while this signal is ON. To perform push-motion operation, turn ON the push-motion input signal before turning the start input signal ON.
A push-motion operation command is specified using two successive position data points.
If the "start" input signal is turned ON while the "push-motion" input signal is ON for position No. n, the position data corresponding to position No. n and position No. n+1 will indicate the following items:
The position data for position No. n indicates the target position.
The position data for position No. n+1 indicates the push width.
The acceleration data for position No. n+1, multiplied by 100, indicates the current-limiting value during push-motion operation.
The speed data for position No. n+1 indicates the push speed.
Example: The position data for position No. 1, as specified in the table below, is used for push-motion operation.
Target position: 100 mm, Push width: 30 mm, Current-limiting value: 50%
Acceleration/deceleration until the push width before the target position: 0.2 G
Push speed: 25 mm/sec

No	Axis1	Axis2	Vel	Acc	Dcl
1	100.000		100	0.20	0.20
2	30.000		25	0.50	

■ Interpolation (LINE)
With the 2-axis specification, input of the position signal and start signal while this signal is ON will cause the two axes to perform interpolation operation (the two axes will start simultaneously and arrive at the target position simultaneously).
To perform interpolation operation, turn ON the interpolation input signal before turning ON the start input signal.



4. Details of Each Output Signal

■ Positioning complete (PEND)
This signal indicates that the actuator reached the target position and the positioning has completed.
After the power was input and the servo has turned on, this signal will turn ON if the position deviation is within the in-position band when the controller becomes ready.
Thereafter, this signal will turn OFF when the start signal is turned ON to execute a movement command.
The signal will turn ON if the position deviation from the target position is within the in-position band after the start signal has turned OFF.
Once this signal turns ON, it will not turn OFF even after the position deviation subsequently exceeds the in-position band.
(Note) If the start signal is ON, this signal will not turn ON even when the position deviation from the target position falls within the in-position band. The signal will turn ON after the start signal turns OFF.
Even if the motor is stopped, this signal will remain OFF if a pause signal is input or the servo is off.

■ Home return complete (HEND)
This signal is OFF when the power is input, and will turn ON when the home-return operation initiated by input of the home-return signal is completed.
Once this signal turns ON, it will not turn OFF until the input power is cut off or the home-return signal is input again.

■ Alarm (*ALM)
This signal remains ON while the controller is normal, and will turn OFF if an alarm occurs.
This signal will turn OFF when an error of operation-cancellation level or higher generates.
Program the PLC so that it will monitor this signal and implement appropriate safety measures to protect the entire system when the signal turns OFF.
For details on alarms, refer to Appendix “◎ Error Level Management” and “◎ Error List.”

■ Ready (RDY)
This signal will turn ON when the initialization has completed successfully after the main power was input, and the controller enters the mode where it can control the actuator.
This signal will turn OFF when an error of cold level or higher generates.
Use this signal as a condition to start control on the PLC side.

■ Servo ON output (SVON)
This signal will turn ON when the servo turns on. Issue a movement command after the servo ON output signal has turned ON.

⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
Put the slider or rod away from the mechanical end when turning the servo ON.

■ System battery error
This signal will turn ON when the voltage of the optional system-memory backup battery drops to a specified level.

■ Absolute battery error
On a controller of absolute specification, this signal will turn ON when the voltage of the absolute-data backup battery drops to a specified level.

5. Timing Chart

5.1 Recognition of I/O Signals

An input time constant is set for the input signals of this controller to prevent malfunction due to chattering, noise, etc.
Except for certain signals, the input signal will switch if the new signal level has remained for at least 6 [msec].
For example, when an input signal is turned ON, the controller will recognize that the signal is ON after elapse of 6 [msec]. The same applies when the signal is turned OFF. (Fig. 1)

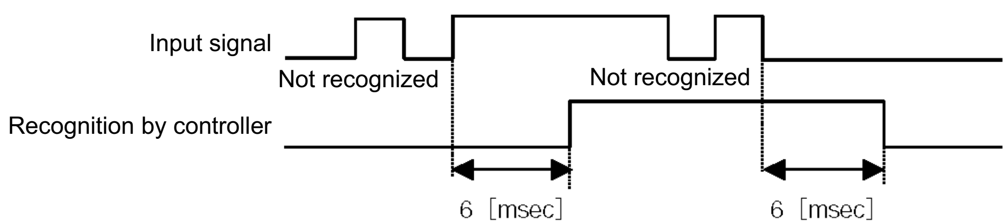
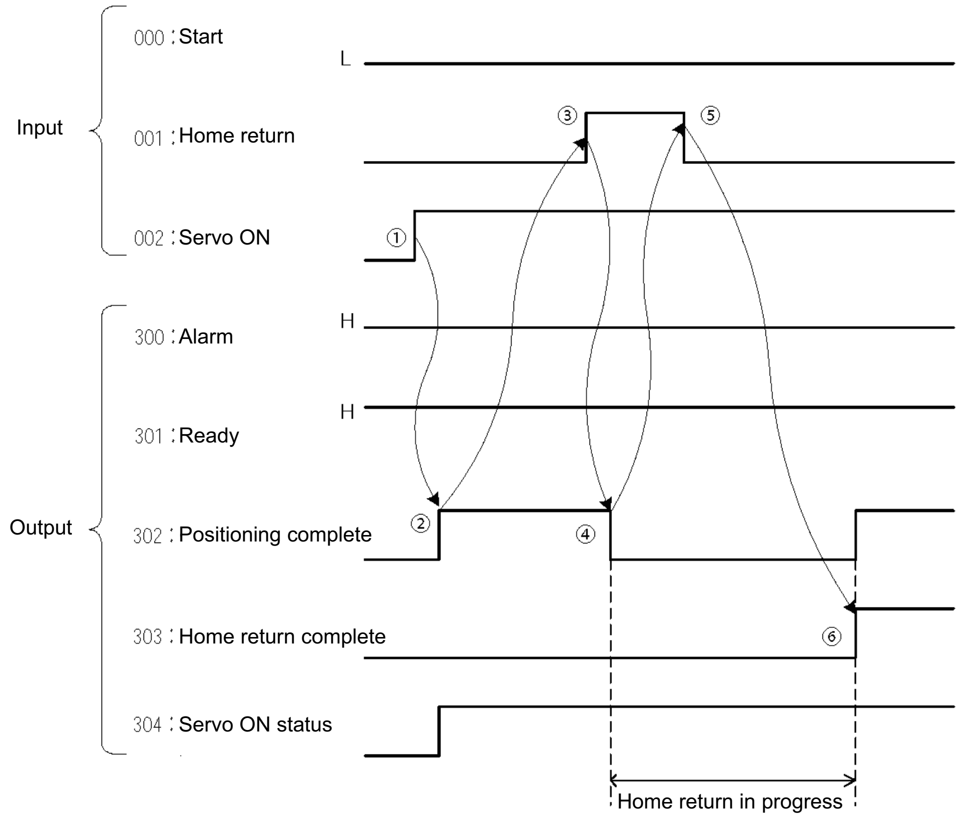


Fig. 1 Recognition of Input Signal

5.2 Home Return

Timings associated with home-return operation are illustrated below.



Timing Chart of Home-return Operation (Standard Positioner Mode)

Perform home-return operation by following the procedure explained below.
* Before commencing the procedure, confirm that the ready output signal and alarm output signal are ON.

[1] Turn ON the servo ON input signal.
[2] Confirm that the servo-ON status output signal is ON.
[3] Turn ON the home-return input signal.
[4] Confirm that the positioning complete output signal is OFF.
[5] Turn OFF the home-return input signal.
[6] Confirm that the home-return complete output signal is ON. Home return is now completed.

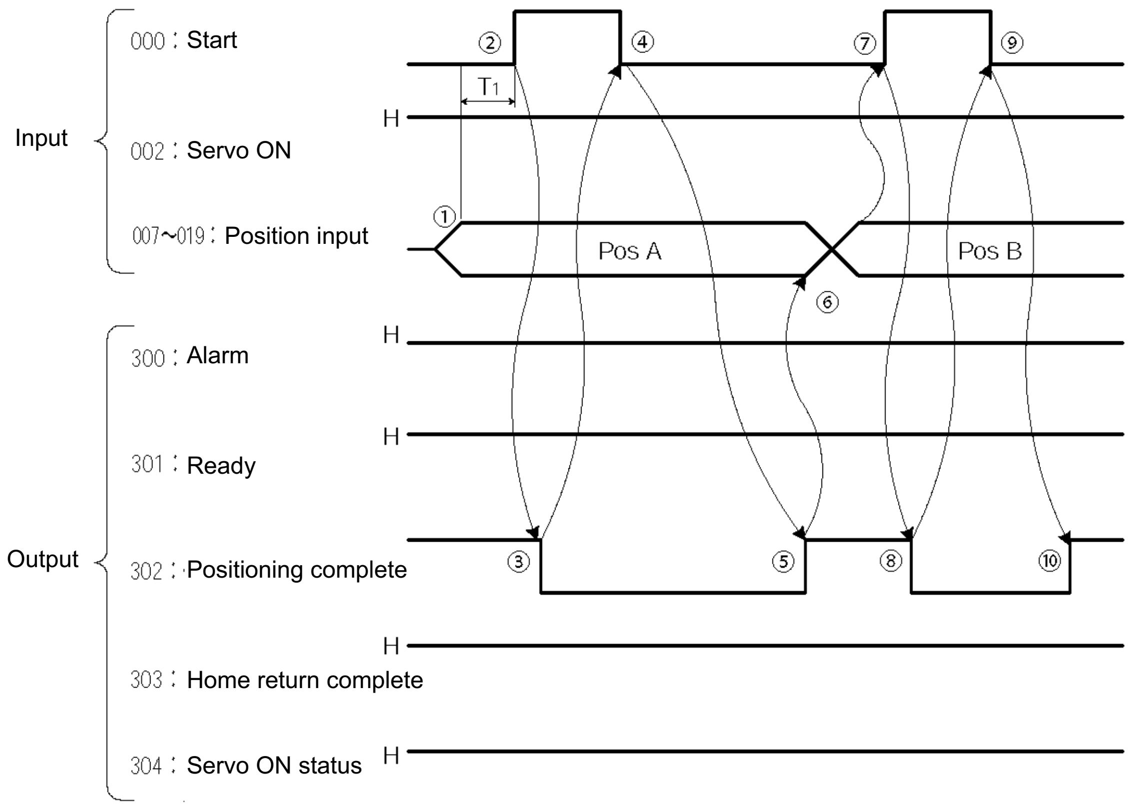
*Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while home return is in progress.
To initiate home return using the home-return signal input, the servo ON input signal must be ON. These operation commands will not be accepted if the servo ON input signal is OFF. Note, however, that only the commands will be ignored and no error will generate.

> ⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

With the 2-axis specification, the controller has been configured at the factory so that the two axes will start home return simultaneously.
You can cause either axis to start home return earlier than the other axis by changing the applicable parameter setting. Specifically, change the setting in axis-specific parameter No. 13, “SIO/PIO home-return order” so that the parameter value for the axis number corresponding to the axis for which you want to complete home return first, will become smaller than the parameter value for the other axis number.
Example) Cause axis 1 to perform home return after axis 2 has completed home return, set “1” for axis 1 and “0” for axis 2 in axis-specific parameter No. 13.

5.3 Movements through Positions

Timings of how the actuator moves through positions are illustrated below.



Timing Chart of Movement through Positions (Standard Positioner Mode)
Ti: At least 6 msec

Operate the actuator to move through positions by following the procedure explained below.

* Confirm beforehand that the positioning complete output signal, home-return complete output signal and servo-ON status output signal are all ON.

[1] Change the previous position number input to a different position number.
[2] Turn ON the start input signal.
[3] Confirm that the positioning complete output signal is OFF.
[4] Turn OFF the start input signal.
[5] Confirm that the positioning complete output signal is ON.
Repeat steps [1] through [5] sequentially.

* Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while the actuator are moving through the specified positions.

* To perform push-motion or interpolation operation, turn ON the applicable input signal before turning ON the start input signal. Turn the operation signal OFF after the start input signal has turned OFF.
* While the actuator is moving to the target position, only the pause or cancellation input is accepted. The servo cannot be turned off even if the servo ON input signal is turned OFF. (The servo can be turned off only when the positioning complete output signal is ON.)
* While the start input signal is ON, the positioning complete output signal will not turn ON even after the actuator physically completes moving to the target position. Therefore, always turn OFF the start input signal ([4]) to detect the completion of positioning.

* As for the positioning complete output signal and push-motion complete output signal, they will not be output until the start signal turns OFF (based on the I/O control handshake rules).

* For the actuator to operate upon start signal input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. Note, however, that only the commands will be ignored and no error will generate.

Chapter 3 Product Switching Mode

In addition to position numbers, product numbers can also be specified in this mode. Sixteen bits of inputs 1 through 16 are divided into position number inputs and product number inputs.
In other words, the actuator can be moved to different positions for different products by specifying the same position number.

1. I/O Interface List

Pin No.	Category	Port No.	Signal name	Signal symbol	Function overview	Cable color
1A	P24		External power supply 24 V	P24		1-Brown
1B	Input	016	Input 10	PC10	(Same as inputs 1 through 9)	1-Red
2A		017	Input 11	PC11		1-Orange
2B		018	Input 12	PC12		1-Yellow
3A		019	Input 13	PC13		1-Green
3B		020	Input 14	PC14		1-Blue
4A		021	Input 15	PC15		1-Purple
4B		022	Input 16	PC16		1-Gray
5A		023	Error reset	RES	Present alarms will be reset at the leading edge of this signal.	1-White
5B		000	Start	CSTR	The actuator will start moving at the leading edge of this signal.	1-Black
6A		001	Home return	HOME	The actuator will start home-return operation at the leading edge of this signal.	2-Brown
6B		002	Servo ON	SON	The servo will remain on while this signal is ON, and remain off while this signal is OFF.	2-Red
7A		003	Push motion	PUSH	The actuator will start linear interpolation operation if the start input signal is turned ON while this signal is ON.	2-Orange
7B		004	*Pause	*STP	The actuator can be moved when this signal is ON, and will decelerate to a stop when the signal turns OFF.	2-Yellow
8A		005	*Cancellation	*CANC	The remaining travel distance will be cancelled if this signal turns OFF.	2-Green
8B		006	Interpolation	LINE	With the 2-axis specification, linear interpolation operation will start when the start input signal is turned ON while this signal is ON.	2-Blue
9A		007	Input 1	PC1	These input signals specify position numbers and product numbers. Sixteen bits of inputs 1 through 16 are divided into position number inputs and product number inputs. Be sure to specify an input by no later than 6 msec before the start signal turns ON. Position numbers and product numbers are input as binary codes (factory setting). The input mode can be changed to BCD by changing the setting of other parameter No. 71.	2-Purple
9B		008	Input 2	PC2		2-Gray
10A		009	Input 3	PC3		2-White
10B		010	Input 4	PC4		2-Black
11A		011	Input 5	PC5		3-Brown
11B		012	Input 6	PC6		3-Red
12A		013	Input 7	PC7		3-Orange
12B		014	Input 8	PC8		3-Yellow
13A		015	Input 9	PC9		3-Green
13B	Output	300	*Alarm	*ALM	This signal remains ON if the controller is normal. It will turn OFF if an alarm occurs.	3-Blue
14A		301	Ready	RDY	This signal will turn ON when the controller becomes ready.	3-Purple
14B		302	Positioning complete	PEND	This signal will turn ON once the actuator has moved to the target position and entered the positioning band.	3-Gray
15A		303	Home return complete	HEND	This signal is OFF when the power is input, and will turn ON when home return is completed.	3-White
15B		304	Servo ON output	SVON	This signal will turn ON when the servo is turned on, and turn OFF when the servo is turned off.	3-Black
16A		305	Push motion complete	PSED	This signal will turn ON when the push-motion operation is completed successfully, and turn OFF if the work is mixed.	4-Brown
16B		306	System battery error	SSER	This signal will turn ON when the voltage of the system-memory backup battery drops to the voltage-low warning level.	4-Red
17A		307	Absolute battery error	ABER	This signal will turn ON when the voltage of the absolute-data backup battery drops to the voltage-low warning level.	4-Orange
17B	N		External power supply 0 V	N		4-Yellow

*: Contact B (always ON)

2. Parameters

The following parameters must be set in the product switching mode.

Table: Parameter Settings in Product Switching Mode

Type	No.	Parameter	Function
Other	25	Operation mode type	2: Product switching mode
	71	Positioner mode parameter 1	Position-number input mode specification (0: Binary, ≠ 0: BCD) * Default value: 0 (Binary)
	72	Positioner mode parameter 2	Number of position-number input bits Binary: Number of bits – 1 through 15 bits BCD: Number of BCD digits – 1 through 3 digits
	73	Positioner mode parameter 3	Number of positions per product

When the above parameters are set, the actual position movement commands will apply based on the following formula:
"(Product number input – 1) x Number of positions per product + Position number input"

For example, assume that the parameters are set as follows:
Other parameter No. 71 = 0 (Binary) "Position-number input mode specification"
Other parameter No. 72 = 6 "Number of position-number input bits"
Other parameter No. 73 = 50 "Number of positions per product"
Each position number is assigned to six bits of inputs 1 through 6 (007 through 012), as a binary code, and position Nos. 1 through 63 can be specified.
Each product number is assigned to 10 bits of inputs 7 through 6 (013 through 022), as a binary code, and 30 types can be specified (the number of types is limited to 30, because the maximum number of position data is 1500). If any greater value is set that brings the number of position data to more than 1500, a "point number error" will generate.

* If the value of position number input exceeds the number of positions per product, the controller will recognize that "1" has been set as the position number.

(Note) The result of "Number of position-number input bits" + "Number of product-number input bits" must not exceed 16 (bits).

3. Details of Each Input Signal

■ Start (CSTR)

Movement to the position corresponding to the position data of the specified product will start upon detection of the OFF → ON leading edge of this signal. Product numbers and position numbers are specified by the 16-bit binary code consisting of inputs 1 through 16.
Before movement is started, the target position, speed and acceleration/deceleration must be set as position data. Use a PC (software) or teaching pendant to set position data.
If this signal is input when no single home-return operation has been performed after the power was input (= when the HEND output signal is OFF), “C6F, Home-return incomplete error” will generate.

■ Inputs 1 through 16 (PC1 through 16)

Sixteen bits of inputs 1 through 16 are divided into position-number input bits and product-number input bits.
Example) Assume that the parameters are set as follows:
  Other parameter No. 71 = 0 (Binary) “Position-number input mode specification”
  Other parameter No. 72 = 6 “Number of position-number input bits”
  Other parameter No. 73 = 50 “Number of positions per product”
Each position number input is assigned to six bits of inputs 1 through 6 (007 through 012), as a binary code.
Each product number input is assigned to 10 bits of inputs 7 through 16 (013 through 022), as a binary code
Position numbers and product numbers are specified as shown in the table below, based on the ON/OFF levels of inputs 1 through 16.

		Product				Position number input					
		Product 1	Product 2	Product 3	Product 4	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
Position number (when set)		1	51	101	151	0	0	0	0	0	1
		2	52	102	152	0	0	0	0	1	0
		3	53	103	153	0	0	0	0	1	1
		4	54	104	154	0	0	0	1	0	0
		:	:	:	:	:	:	:	:	:	:
		:	:	:	:	:	:	:	:	:	:
		49	99	149	199	1	1	0	0	0	1
		50	100	150	200	1	1	0	0	1	0
Product number input	Input 7	1	0	1	0						
	Input 8	0	1	1	0						
	Input 9	0	0	0	1						
	Input 10	0	0	0	0						
	Input 11	0	0	0	0						
	Input 12	0	0	0	0						
	Input 13	0	0	0	0						
	Input 14	0	0	0	0						
	Input 15	0	0	0	0						
	Input 16	0	0	0	0						

Fifty position numbers (Nos. 1 through 50) can be specified for each product.
Position No. 49 for product 2 (set as No. 99 within the entire data) is specified as follows.

Input 16	Input 15	Input 14	Input 13	Input 12	Input 11	Input 10	Input 9	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

The input mode can be changed to BCD by changing the setting of other parameter No. 71.
Assume the following settings:
Other parameter No. 71, "Position-number input method specification" = 1 (BCD)
Other parameter No. 72, "Number of position-number input bits" = 8
(In the BCD input mode, one digit consists of four bits. In other words, bits are input in units of four.)
Other parameter No. 73, "Number of positions per product" = 50
Each position number is assigned to eight bits of inputs 1 through 8 (007 through 014), as a two-digit BCD code.
Each product number is assigned to eight bits of inputs 9 through 16 (015 through 022), as a two-digit BCD code.
As for the position number, specify the one's place in inputs 1 through 4, and ten's place in inputs 5 through 8.
As for the product number, specify the one's place in inputs 9 through 12, and ten's place in inputs 13 through 16.

■ Pause (*STP)

If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be held, which means that when the signal turns ON again, the actuator will resume movement of the remaining travel distance.
To cancel the movement command altogether after turning OFF the pause signal, turn OFF the cancellation signal while this signal is OFF to cancel the remaining travel distance.
The pause signal can be used for the following purposes:
[1] As a sensor to detect entry into a specified area around the system or for other lower-level safety measures to stop the axis while the servo is on
[2] To prevent contact with other equipment
[3] For positioning based on sensor or LS signal detection
(Note) When this signal is input during home return, the movement command will be held if the actuator has not yet contacted the mechanical end. If the signal is input after the actuator has reversed upon contacting the mechanical end, home return will be performed again.

■ Cancellation (*CANC)

If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop. The remaining travel distance will be cancelled and the movement will not resume even when the signal turns ON thereafter.

■ Home return (HOME)

The actuator will start home-return operation upon detection of the OFF → ON edge of this signal.
Once the home return is complete, the HEND signal will be output. This signal can be input as many times as desired after completion of the initial home return.
(Note) An actuator of incremental specification must always perform home return after the power is turned on.

■ Servo ON (SON)

The servo remains on while this signal is ON.
Use this signal if servo ON/OFF control is required as part of the safety circuit for the entire system to be provided on the PLC side.
To operate the actuator using the start input/home return input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. (Only the commands will be ignored, and no error will generate.)
(Note) When this signal turns OFF while the actuator is moving, the actuator will not decelerate to a stop. It will complete the movement to the target position, after which the servo will turn off.

⚠ Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
Put the slider or rod away from the mechanical end when turning the servo ON.

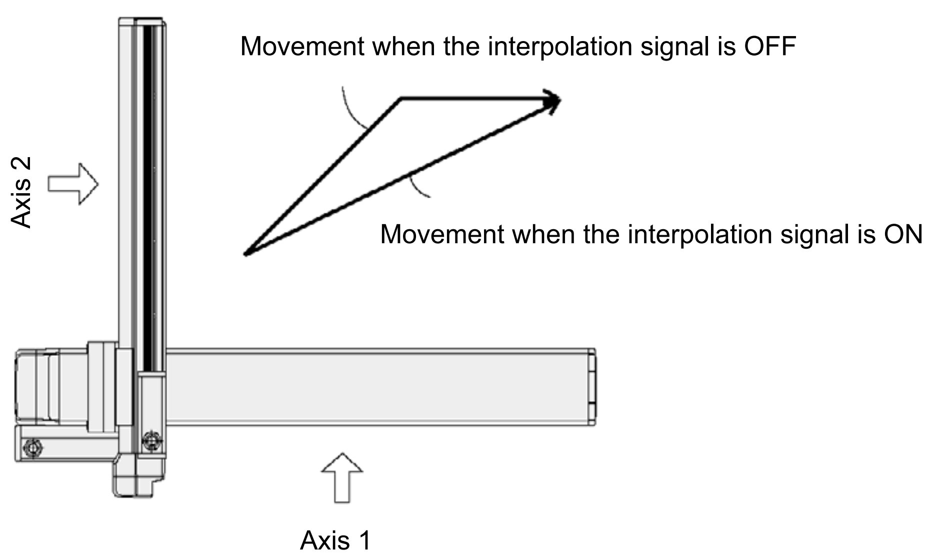
■ Error reset (RES)
This signal is used to reset the alarm output signal (*ALM) that has been generated due to an error.
If an error occurred, check the content of the error and then turn this signal ON.
The error will be reset upon detection of the leading edge of the signal.
(Note) Errors of cold start and higher level cannot be reset using this signal. The power must be reconnected to reset these errors. For details, refer to Appendix, "Error Level Management."

■ Push motion (PUSH)
The actuator will perform push-motion operation if the position signal and start signal are input while this signal is ON. To perform push-motion operation, turn ON the push-motion input signal before turning the start input signal ON.
A push-motion operation command is specified using two successive position data points.
If the "start" input signal is turned ON while the "push-motion" input signal is ON for position No. n, the position data corresponding to position No. n and position No. n+1 will indicate the following items:
The position data for position No. n indicates the target position.
The position data for position No. n+1 indicates the push width.
The speed data for position No. n+1 indicates the push speed.
The acceleration data for position No. n+1, multiplied by 100, indicates the current-limiting value during push-motion operation.
Example: The position data for position No. 1, as specified in the table below, is used for push-motion operation.

No	Axis1	Axis2	Vel	Acc	Dcl
1	100.000		100	0.20	0.20
2	30.000		25	0.50	

The actuator moves at a speed of 100 mm/sec, acceleration of 0.2 G and deceleration of 0.2 G, until 30 mm before a target position of 100 mm. Thereafter, the actuator performs push-motion operation to the target position at a speed of 25 mm/sec and current-limiting value of 50%.

■ Interpolation (LINE)
With the 2-axis specification, input of the position signal and start signal while this signal is ON will cause the two axes to perform interpolation operation (the two axes will start simultaneously and arrive at the target position simultaneously).
To perform interpolation operation, turn ON the interpolation input signal before turning ON the start input signal.



4. Details of Each Output Signal

■ Positioning complete (PEND)

This signal indicates that the actuator reached the target position and the positioning has completed.
After the power was input and the servo has turned on, this signal will turn ON if the position deviation is within the in-position band when the controller becomes ready.
Thereafter, this signal will turn OFF when the start signal is turned ON to execute a movement command.
The signal will turn ON if the position deviation from the target position is within the in-position band after the start signal has turned OFF.
Once this signal turns ON, it will not turn OFF even after the position deviation subsequently exceeds the in-position band.

(Note) If the start signal is ON, this signal will not turn ON even when the position deviation from the target position falls within the in-position band. The signal will turn ON after the start signal turns OFF.
Even if the motor is stopped, this signal will remain OFF if a pause signal is input or the servo is off.

■ Home return complete (HEND)

This signal is OFF when the power is input, and will turn ON when the home-return operation initiated by input of the home-return signal is completed.
Once this signal turns ON, it will not turn OFF until the input power is cut off or the home-return signal is input again.

■ Alarm (*ALM)

This signal remains ON while the controller is normal, and will turn OFF if an alarm occurs.
This signal will turn OFF when an error of operation-cancellation level or higher generates.
Program the PLC so that it will monitor this signal and implement appropriate safety measures to protect the entire system when the signal turns OFF.
For details on alarms, refer to Appendix "◎ Error Level Management" and "◎ Error List."

■ Ready (RDY)

This signal will turn ON when the initialization has completed successfully after the main power was input, and the controller enters the mode where it can control the actuator.
This signal will turn OFF when an error of cold level or higher generates.
Use this signal as a condition to start control on the PLC side.

■ Servo ON output (SVON)

This signal will turn ON when the servo turns on. Issue a movement command after the servo ON output signal has turned ON.

■ System battery error

This signal will turn ON when the voltage of the optional system-memory backup battery drops to a specified level.

■ Absolute battery error

On a controller of absolute specification, this signal will turn ON when the voltage of the absolute-data backup battery drops to a specified level.

5. Timing Chart

5.1 Recognition of I/O Signals

An input time constant is set for the input signals of this controller to prevent malfunction due to chattering, noise, etc.
Except for certain signals, the input signal will switch if the new signal level has remained for at least 6 [msec].
For example, when an input signal is turned ON, the controller will recognize that the signal is ON after elapse of 6 [msec]. The same applies when the signal is turned OFF. (Fig. 1)

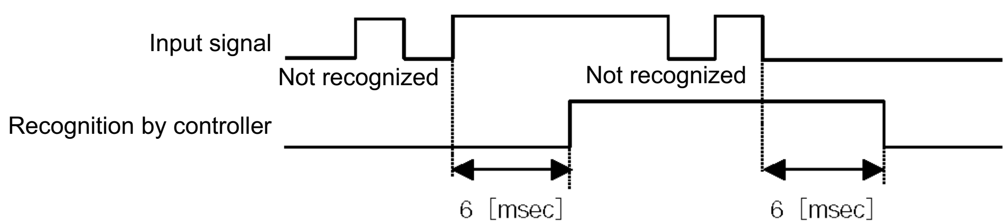
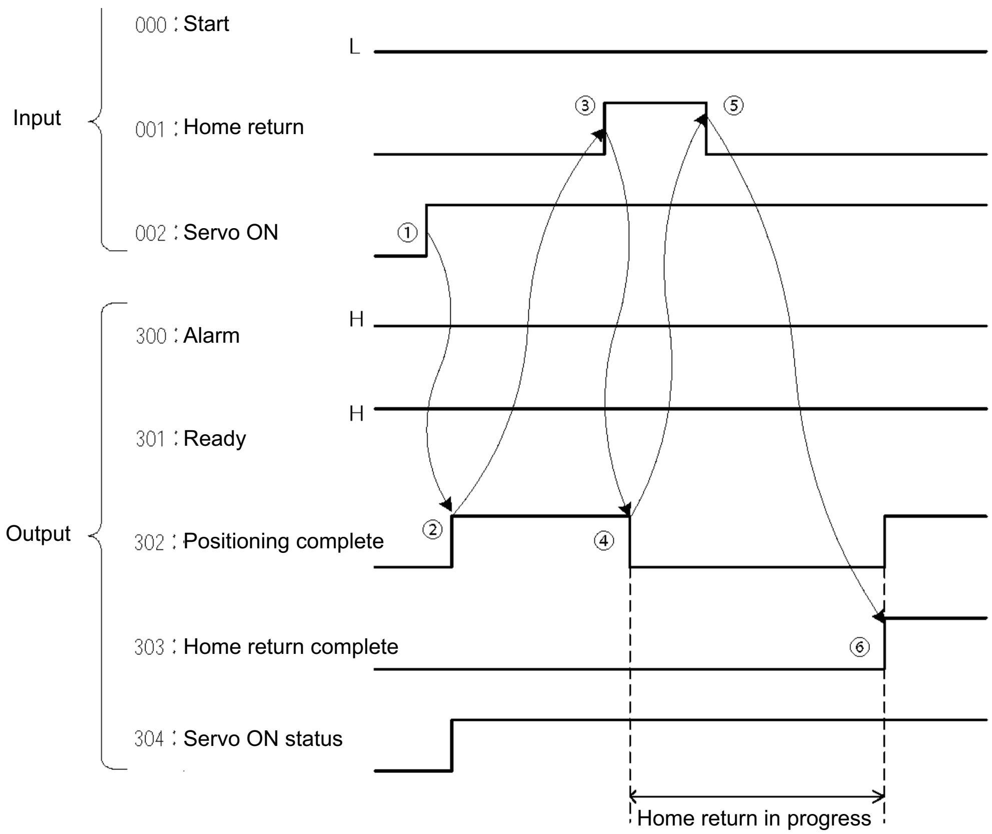


Fig. 1 Recognition of Input Signal

5.2 Home Return

Timings associated with home-return operation are illustrated below.



Timing Chart of Home-return Operation (Standard Positioner Mode)

Perform home-return operation by following the procedure explained below.

* Before commencing the procedure, confirm that the ready output signal and alarm output signal are ON.

[1] Turn ON the servo ON input signal.
[2] Confirm that the servo-ON status output signal is ON.
[3] Turn ON the home-return input signal.
[4] Confirm that the positioning complete output signal is OFF.
[5] Turn OFF the home-return input signal.
[6] Confirm that the home-return complete output signal is ON. Home return is now completed.

*Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while home return is in progress.

To initiate home return using the home-return signal input, the servo ON input signal must be ON. These operation commands will not be accepted if the servo ON input signal is OFF. Note, however, that only the commands will be ignored and no error will generate.

Warning :	Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error. Put the slider or rod away from the mechanical end when turning the servo ON.

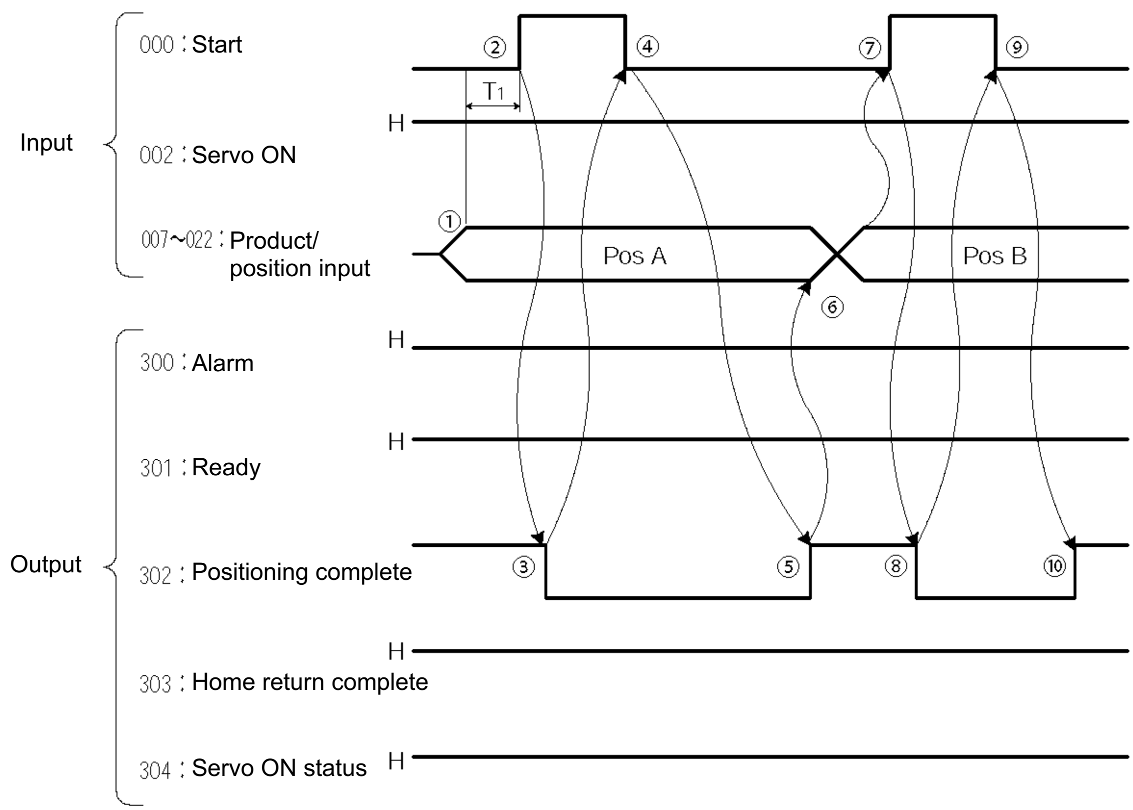
With the 2-axis specification, the controller has been configured at the factory so that the two axes will start home return simultaneously.

You can cause either axis to start home return earlier than the other axis by changing the applicable parameter setting. Specifically, change the setting in axis-specific parameter No. 13, “SIO/PIO home-return order” so that the parameter value for the axis number corresponding to the axis for which you want to complete home return first, will become smaller than the parameter value for the other axis number.

Example) Cause axis 1 to perform home return after axis 2 has completed home return, set “1” for axis 1 and “0” for axis 2 in axis-specific parameter No. 13.

5.3 Movements through Positions

Timings of how the actuator moves through positions are illustrated below.



Timing Chart of Movement through Positions (Standard Positioner Mode)
Ti: At least 6 msec

Operate the actuator to move through positions by following the procedure explained below.

* Confirm beforehand that the positioning complete output signal, home-return complete output signal and servo-ON status output signal are all ON.
  [1] Change the previous product/position number inputs to different product/position numbers.
  [2] Turn ON the start input signal.
  [3] Confirm that the positioning complete output signal is OFF.
  [4] Turn OFF the start input signal.
  [5] Confirm that the positioning complete output signal is ON.
  Repeat steps [1] through [5] sequentially.
* Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while the actuator are moving through the specified positions.

* To perform push-motion or interpolation operation, turn ON the applicable input signal before turning ON the start input signal. Turn the operation signal OFF after the start input signal has turned OFF.
* While the actuator is moving to the target position, only the pause or cancellation input is accepted. The servo cannot be turned off even if the servo ON input signal is turned OFF. (The servo can be turned off only when the positioning complete output signal is ON.)
* While the start input signal is ON, the positioning complete output signal will not turn ON even after the actuator physically completes moving to the target position. Therefore, always turn OFF the start input signal ([4]) to detect the completion of positioning.

* As for the positioning complete output signal and push-motion complete output signal, they will not be output until the start signal turns OFF (based on the I/O control handshake rules).

* For the actuator to operate upon start signal input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. Note, however, that only the commands will be ignored and no error will generate.

Chapter 4 2-axis Independent Mode

With the 2-axis specification, each axis can be controlled separately in this mode. A set of signals, such as the start input signal and positioning complete output signal, are provided for each axis.
Although the position number specification applies commonly to both axes, 13 bits of position inputs 1 through 13 (PC1 through 13) are divided into position-number specification bits for axis 1 and position-number specification bits for axis 2.

1. I/O Interface List

Pin No.	Category	Port No.	Signal name	Signal symbol	Function overview	Cable color
1A	P24		External power supply 24 V	P24		1-Brown
1B	Input	016	Position input 7	PC7	(Same as position inputs 1 through 9)	1-Red
2A		017	Position input 8	PC8		1-Orange
2B		018	Position input 9	PC9		1-Yellow
3A		019	Position input 10	PC10		1-Green
3B		020	Position input 11	PC11		1-Blue
4A		021	Position input 12	PC12		1-Purple
4B		022	Position input 13	PC13		1-Gray
5A		023	Error reset	RES	Present alarms will be reset at the leading edge of this signal.	1-White
5B		000	Axis 1 start	CSTR1	Axis 1 will start moving at the leading edge of this signal.	1-Black
6A		001	Axis 1 home return	HOME1	Axis 1 will start home-return operation at the leading edge of this signal.	2-Brown
6B		002	Axis 1 servo ON	SON1	The servo for axis 1 will remain on while this signal is ON, and remain off while this signal is OFF.	2-Red
7A		003	*Axis 1 pause	*STP1	Axis 1 can be moved when this signal turns ON, and will decelerate to a stop when the signal turns OFF.	2-Orange
7B		004	*Axis 1 cancellation	*CANC	The remaining travel distance of axis 1 will be cancelled if this signal turns OFF.	2-Yellow
8A		005	Axis 2 start	CSTR2	Axis 2 will start moving at the leading edge of this signal.	2-Green
8B		006	Axis 2 home return	HOME2	Axis 2 will start home-return operation at the leading edge of this signal.	2-Blue
9A		007	Axis 2 servo ON	SON2	The servo for axis 2 will remain on while this signal is ON, and remain off while this signal is OFF.	2-Purple
9B		008	*Axis 2 pause	*STP2	Axis 2 can be moved when this signal turns ON, and will decelerate to a stop when the signal turns OFF.	2-Gray
10A		009	*Axis 2 cancellation	*CANC2	The remaining travel distance of axis 2 will be cancelled if this signal turns OFF.	2-White
10B		010	Position input 1	PC1	Thirteen bits of position inputs 1 through 13 are divided into position-number specification bits for axis 1 and position-number specification bits for axis 2.	2-Black
11A		011	Position input 2	PC2		3-Brown
11B		012	Position input 3	PC3		3-Red
12A		013	Position input 4	PC4		3-Orange
12B		014	Position input 5	PC5		3-Yellow
13A		015	Position input 6	PC6		3-Green
13B	Output	300	*Alarm	*ALM	This signal remains ON if the controller is normal. It will turn OFF if an alarm occurs.	3-Blue
14A		301	Ready	RDY	This signal will turn ON when the controller becomes ready.	3-Purple
14B		302	Axis 1 positioning complete	PEND1	This signal will turn ON once axis 1 has moved to the target position and entered the positioning band.	3-Gray
15A		303	Axis 1 home-return complete	HEND1	This signal is OFF when the power to axis 1 is input, and will turn ON when home return is completed.	3-White
15B		304	Axis 1 servo ON	SVON1	This signal will turn ON when the servo for axis 1 is turned on, and turn OFF when the servo is turned off.	3-Black
16A		305	Axis 2 positioning complete	PEND2	This signal will turn ON once axis 2 has moved to the target position and entered the positioning band.	4-Brown
16B		306	Axis 2 home-return complete	HEND2	This signal is OFF when the power to axis 2 is input, and will turn ON when home return is completed.	4-Red
17A		307	Axis 2 servo ON	SVON2	This signal will turn ON when the servo for axis 2 is turned on, and turn OFF when the servo is turned off.	4-Orange
17B	N		External power supply 0 V	N		4-Yellow

*: Contact B (always ON)

2. Parameters

The following parameters must be set in the 2-axis independent mode.

Type	No.	Parameter	Function
Other	25	Operation mode type	3: 2-axis independent mode
	71	Positioner mode parameter 1	Position-number input mode specification (0: Binary, ≠ 0: BCD) * Default value: 0 (Binary)
	72	Positioner mode parameter 2	Specification of number of position-number input bits for axis 1 Binary: Number of bits – 1 through 12 bits BCD: Number of BCD digits – 1 or 2 digits

Specify the number of position-number input bits for axis 1 in other parameter No. 72, “Positioner mode parameter 2.” Specify how many bits will be assigned to axis 1, from among the 13 bits of position inputs 1 through 13. The remainder of the bits will be assigned to axis 2.
By specifying binary or BCD in the “position-number input mode specification” parameter, the setting unit of this parameter will change between bit and BCD digit.

Example) Assume that the parameters are set as follows:
Other parameter No. 71 = 0 (Binary) “Position-number input mode specification”
Other parameter No. 72 = 7 “Specification of position-number input bits for axis 1”

Each position number input for axis 1 is assigned to seven bits of inputs 1 through 7 (010 through 016), as a binary code, and position Nos. 1 through 127 can be specified.
Each position number input for axis 2 is assigned to the remaining six bits of inputs 8 through 13 (017 through 022), as a binary code, and position Nos. 1 through 63 can be specified.

3. Details of Each Input Signal

■ Position inputs 1 through 13 (PC1 through 13)
Thirteen bits of PC1 through 13 are divided into position-number specification bits for axis 1 and position-number specification bits for axis 2.
Example) Assume that the parameters are set as follows:
Other parameter No. 71 = 0 (Binary) "Position-number input mode specification"
Other parameter No. 72 = 7 "Specification of position-number input bits for axis 1"

Each position number input for axis 1 is assigned to seven bits of PC1 through 7 (010 through 016), as a binary code, and position Nos. 1 through 127 can be specified.
Each position number input for axis 2 is assigned to the remaining six bits of PC8 through 13 (017 through 022), as a binary code, and position Nos. 1 through 63 can be specified.
Position numbers for respective axes are specified as shown in the table below, based on the ON/OFF levels of PC1 through 13.

Position number specification for axis 2						Position No.	Position number specification for axis 1						
PC13	PC12	PC11	PC10	PC9	PC8		PC7	PC6	PC5	PC4	PC3	PC2	PC1
0	0	0	0	0	1	1	0	0	0	0	0	0	1
0	0	0	0	1	0	2	0	0	0	0	0	1	0
0	0	0	0	1	1	3	0	0	0	0	0	1	1
:	:	:	:	:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	0	62	0	1	1	1	1	0	0
1	1	1	1	1	1	63	0	1	1	1	1	1	1
						:	:	:	:	:	:	:	:
						126	1	1	1	1	1	1	0
						127	1	1	1	1	1	1	1

Also, the input mode can be changed to BCD by changing the setting of other parameter No. 71.
In the BCD input mode, one digit consists of four bits. Since there are 13 position input bits, the total number of digits assigned to the two axes will become 3.

Assume that the parameters are set as follows:
Other parameter No. 71 = 1 (BCD) "Position-number input mode specification"
Other parameter No. 72 = 8 "Specification of position-number input bits for axis 1"
(Bits are input in units of four.)
Each position number input for axis 1 is assigned to eight bits of PC1 through 8 (010 through 017), as a two-digit BCD code (position Nos. 1 to 99 can be specified). Specify the one's place in PC1 through 4, and ten's place in PC5 through 8.
Each position number input for axis 2 is assigned to five bits (actually four bits) of PC9 through 13 (011 through 022), as a one-digit BCD code (position Nos. 1 to 9 can be specified).

■ Axis 1 start (CSTR1)
Axis 1 will start moving to the position corresponding to the specified position data for axis 1 upon detection of the OFF → ON leading edge of this signal. Position numbers are specified using, among the 13 bits of PC1 through 13, the number of bits set in other parameter No. 72. Position numbers are specified as binary codes according to the factory setting.
Before movement is started, the target position, speed and acceleration/deceleration must be set as position data. Use a PC (software) or teaching pendant to set position data.
If this signal is input when no single home-return operation has been performed after the power was input (= when the HEND output signal is OFF), "C6F, Home-return incomplete error" will generate.

■ Axis 2 start (CSTR2)
Axis 2 will start moving to the position corresponding to the specified position data for axis 2 upon detection of the OFF → ON leading edge of this signal. Position numbers are specified using, among the 13 bits of PC1 through 13, the remainder of the bits excluding those used for axis 1. Other specifications are the same as those explained under "Start 1 (CSTR1)."

■ Axis 1 pause (*STP1)
If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be held, which means that when the signal turns ON again, the actuator will resume movement of the remaining travel distance.
To cancel the movement command altogether after turning OFF the pause signal, turn OFF the CANC1 while this signal is OFF to cancel the remaining travel distance.
The pause signal can be used for the following purposes:
[1] As a sensor to detect entry into a specified area around the system or for other lower-level safety measures to stop the axis while the servo is on
[2] To prevent contact with other equipment
[3] For positioning based on sensor or LS signal detection
(Note) When this signal is input during home return, the movement command will be held if the actuator has not yet contacted the mechanical end. If the signal is input after the actuator has reversed upon contacting the mechanical end, home return will be performed again.

■ Axis 2 pause (*STP2)
If this signal turns OFF while axis 2 is moving, the controller will cause the actuator to decelerate to a stop.
For other items, the same explanation under "Axis 1 pause (*STP1)" applies, except that CANC2 is used as the signal to cancel movement commands.

■ Axis 1 cancellation (*CANC1)
If this signal turns OFF while axis 1 is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be cancelled, which means that even when the signal turns ON again, the actuator will not resume movement.

■ Axis 2 cancellation (*CANC2)
If this signal turns OFF while axis 2 is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be cancelled, which means that even when the signal turns ON again, the actuator will not resume movement.

■ Axis 1 home return (HOME1)
Axis 1 will stat home-return operation upon detection of the OFF → ON edge of this signal.
Once the home return is complete, the HEND1 signal will be output. This signal can be input as many times as desired after completion of the initial home return.
(Note) An actuator of incremental specification must always perform home return after the power is turned on.

■ Axis 2 home return (HOME2)
Axis 2 will start home-return operation upon detection of the OFF → ON edge of this signal.
Once the home return is complete, the HEND2 signal will be output.
For other items, the same explanation under "Axis 1 home return (HOME1)" applies.

■ Axis 1 servo ON (SON1)
The servo for axis 1 will remain ON while this signal is ON.
To operate the actuator using the start input/home return input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. (Only the commands will be ignored, and no error will generate.)
(Note) When this signal turns OFF while the actuator is moving, the actuator will not decelerate to a stop. It will complete the movement to the target position, after which the servo will turn off.

⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error. Put the slider or rod away from the mechanical end when turning the servo ON.

■ Axis 2 servo ON (SON2)
The axis 2 servo remains ON while this signal is ON.
For other items, the same explanation under "Axis 1 servo ON (SON1)" applies.

■ Error reset (RES)
[1] This signal is used to reset the alarm output signal (*ALM) that has been generated due to an error.
If an error occurred, check the content of the error and then turn this signal ON.
The error will be reset upon detection of the leading edge of the signal.
(Note) Depending on the nature of error, some errors cannot be reset using this signal. For details, refer to 10, "Troubleshooting."
Errors of cold start and higher level cannot be reset using this signal. The power must be reconnected to reset these errors. For details, refer to Appendix, "Error Level Management."

4. Details of Each Output Signal

■ Axis 1 positioning complete (PEND1)
This signal indicates that axis 1 reached the target position and the positioning has completed.
Use it together with the aforementioned MOVE signal to determine the positioning completion status on the PLC side.
After the power was input and the servo has turned on, this signal will turn ON if the position deviation is within the in-position band when the controller becomes ready.
Thereafter, this signal will turn OFF when the start signal is turned ON to execute a movement command.
The signal will turn ON if the position deviation from the target position is within the in-position band after the start signal has turned OFF.
Once this signal turns ON, it will not turn OFF even after the position deviation subsequently exceeds the in-position band.
(Note) If the start signal is ON, this signal will not turn ON even when the position deviation from the target position falls within the in-position band. The signal will turn ON after the start signal turns OFF.
Even if the motor is stopped, this signal will remain OFF if a pause signal is input or the servo is off.

■ Axis 2 positioning complete (PEND2)
This signal indicates that axis 2 reached the target position and the positioning has completed.
For other items, the same explanation under "Axis 1 positioning complete (PEND1)" applies.

■ Axis 1 home return complete (HEND1)
This signal is OFF while the power is input. It will turn ON at the following timings:
[1] The home-return operation has completed in connection with the first movement command issued with the start signal.
[2] The home-return operation has completed following an input of the home return signal.
Once this signal turns ON, it will not turn OFF until the input power is cut off or the axis 1 home return signal (HOME1) is input again.

■ Axis 2 home return complete (HEND2)
This signal is OFF while the power is input. It will turn ON at the following timings:
[1] The home-return operation has completed in connection with the first axis 2 movement command issued with the start signal.
[2] The home-return operation of axis 2 has completed following an input of the axis 2 home return signal (HOME2).
Once this signal turns ON, it will not turn OFF until the input power is cut off or the axis 2 home return signal (HOME2) is input again.

■ Alarm (*ALM)
This signal remains ON while the controller is normal, and will turn OFF if an alarm occurs.
This signal will turn OFF when an error of operation-cancellation level or higher generates.
Program the PLC so that it will monitor this signal and implement appropriate safety measures to protect the entire system when the signal turns OFF.
For details on alarms, refer to Appendix "◎ Error Level Management" and "◎ Error List."

■ Ready (RDY)
This signal will turn ON when the initialization has completed successfully after the main power was input, and the controller enters the mode where it can control the actuator.
This signal will turn OFF when an error of cold level or higher generates.
Use this signal as a condition to start control on the PLC side.

■ Servo ON output 1 (SVON1)
This signal will turn ON when the servo for axis 1 turns on. Issue a movement command after the servo ON output signal has turned ON.

■ Servo ON output 2 (SVON2)
This signal will turn ON when the servo for axis 2 turns on. Issue a movement command after the servo ON output signal has turned ON.

5. Timing Chart

5.1 Recognition of I/O Signals

An input time constant is set for the input signals of this controller to prevent malfunction due to chattering, noise, etc.
Except for certain signals, the input signal will switch if the new signal level has remained for at least 6 [msec].
For example, when an input signal is turned ON, the controller will recognize that the signal is ON after elapse of 6 [msec]. The same applies when the signal is turned OFF. (Fig. 1)

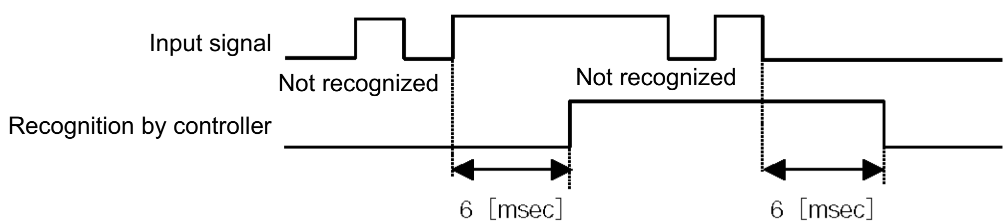
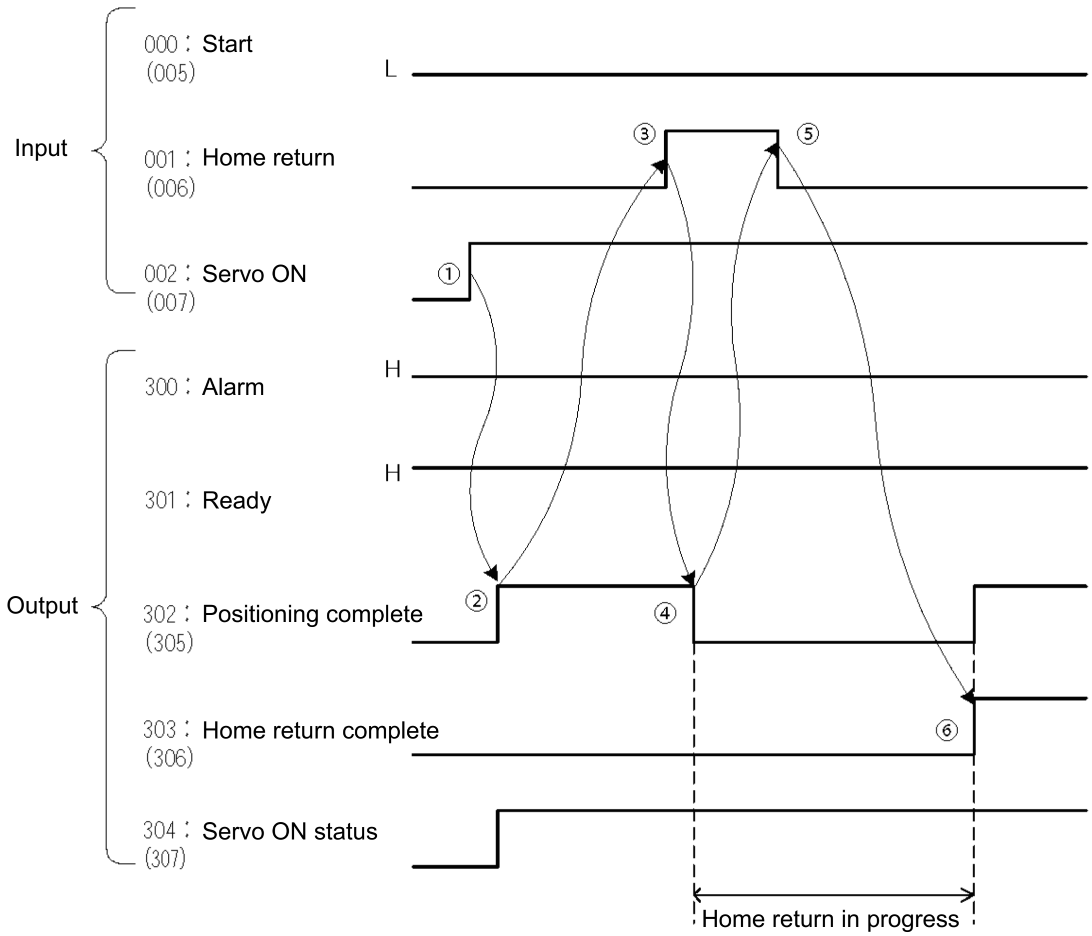


Fig. 1 Recognition of Input Signal

5.2 Home Return

Timings associated with home-return operation are illustrated below. The figures in parentheses indicate port numbers for axis 2.



Timing Chart of Home-return Operation (Standard Positioner Mode)

Perform home-return operation by following the procedure explained below.
* Before commencing the procedure, confirm that the ready output signal and alarm output signal are OFF.

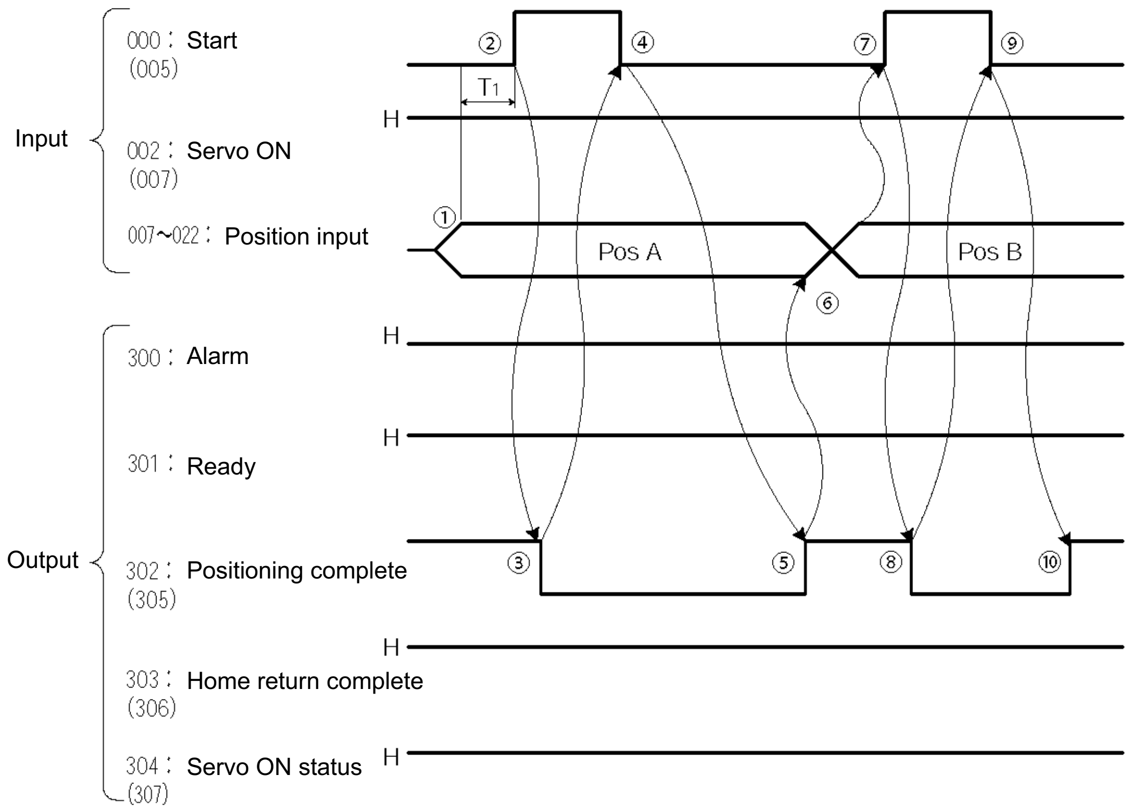
[1] Turn ON the servo ON input signal.
[2] Confirm that the servo-ON status output signal is ON.
[3] Turn ON the home-return input signal.
[4] Confirm that the positioning complete output signal is OFF.
[5] Turn OFF the home-return input signal.
[6] Confirm that the home-return complete output signal is ON. Home return is now completed.

*Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while home return is in progress.
To initiate home return using the home-return signal input, the servo ON input signal must be ON. These operation commands will not be accepted if the servo ON input signal is OFF. Note, however, that only the commands will be ignored and no error will generate.

> ⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

5.3 Movements through Positions

Timings of how the actuator moves through positions are illustrated below. The figures in parentheses indicate port numbers for axis 2.



Timing Chart of Movement through Positions (Standard Positioner Mode)
Ti: At least 6 msec

Operate the actuator to move through positions by following the procedure explained below.

* Confirm beforehand that the positioning complete output signal, home-return complete output signal and servo-ON status output signal are all ON.
  - [1] Change the previous position number input (BCD input) to a different position number .
  - [2] Turn ON the start input signal.
  - [3] Confirm that the positioning complete output signal is OFF.
  - [4] Turn OFF the start input signal.
  - [5] Confirm that the positioning complete output signal is ON.
    Repeat steps [1] through [5] sequentially.
* Pause and *cancellation inputs are contact-B input signals (always ON), so keep these signals ON while the actuator are moving through the specified positions.

* While the actuator is moving to the target position, only the pause or cancellation input is accepted. The servo cannot be turned off even if the servo ON input signal is turned OFF. (The servo can be turned off only when the positioning complete output signal is ON.)
* While the start input signal is ON, the positioning complete output signal will not turn ON even after the actuator physically completes moving to the target position. Therefore, always turn OFF the start input signal ([4]) to detect the completion of positioning.

* As for the positioning complete output signal and push-motion complete output signal, they will not be output until the start signal turns OFF (based on the I/O control handshake rules).

* For the actuator to operate upon start signal input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. Note, however, that only the commands will be ignored and no error will generate.

Chapter 5 Teaching Mode

In addition to normal positioning operation, jogging, inching and teaching can be performed in this mode. A dedicated input is used to switch to the teaching mode, where the actuator can be moved using I/Os and the achieved position can be written to the position data table.

Caution:	Position data input via teaching will be lost when the power is turned off. To retain the position data, one of the following measures must be taken: • Install the optional system-memory backup battery to back up the position data. To do this, the setting of other parameter No. 20 must be changed to “2.” Note, however, that the position data may still be lost if the battery voltage drops. (The battery should be replaced after approx. five years.) If the battery is replaced as soon as a voltage-low warning generates, the data will be retained. Once a voltage-low error generates, the data will be lost. Use the host PLC, etc., to monitor for a system-memory backup error output. • Write the position data to the flash memory using a teaching pendant or PC (software).

1. I/O Interface List

Pin No.	Category	Port No.	Signal name	Signal symbol	Function overview	Cable color
1A	P24		External power supply 24 V	P24		1-Brown
1B	Input	016	Axis 1 jog-	JOG1-	Axis 1 will move in the negative direction while this signal is ON.	1-Red
2A		017	Axis 2 jog+	JOG2+	Axis 2 will move in the positive direction while this signal is ON.	1-Orange
2B		018	Axis 2 jog-	KPG2-	Axis 2 will move in the negative direction while this signal is ON.	1-Yellow
3A		019	Inching (0.01 mm)	1C001	"0.01 mm" is specified as the inching distance.	1-Green
3B		020	Inching (0.1 mm)	1C01	"0.1 mm" is specified as the inching distance.	1-Blue
4A		021	Inching (0.5 mm)	1C05	"0.5 mm" is specified as the inching distance.	1-Purple
4B		022	Inching (1 mm)	1C1	"1 mm" is specified as the inching distance.	1-Gray
5A		023	Error reset	RES	Present alarms will be reset at the leading edge of this signal.	1-White
5B		000	Start	CSTR	The actuator will start moving at the leading edge of this signal.	1-Black
			Current position write	PWRT	The current position is written in the teaching mode.	
6A		001	Servo ON	SON	The servo will remain on while this signal is ON, and remain off while this signal is OFF.	2-Brown
6B		002	*Pause	*STP	The actuator can be moved when this signal is ON, and will decelerate to a stop when the signal turns OFF.	2-Red
7A		003	Position input 1	PC1	Input the position number corresponding to the position you want to move the actuator to. Be sure to specify a position input by no later than 6 msec before the start input signal turns ON. Position numbers are input as binary codes (factory setting). In the teaching mode, specify the position number for which the current position will be written. Position numbers are input as binary codes (factory setting).	2-Orange
7B		004	Position input 2	PC2		2-Yellow
8A		005	Position input 3	PC3		2-Green
8B		006	Position input 4	PC4		2-Blue
9A		007	Position input 5	PC5		2-Purple
9B		008	Position input 6	PC6		2-Gray
10A		009	Position input 7	PC7		2-White
10B		010	Position input 8	PC8		2-Black
11A		011	Position input 9	PC9		3-Brown
11B		012	Position input 10	PC10		3-Red
12A		013	Position input 11	PC11		3-Orange
12B		014	Teaching mode specification	MODE	ON: Teaching mode OFF: Positioner mode	3-Yellow
13A		015	Axis 1 jog+	JOG1+	Axis 1 will move in the positive direction while this signal is ON.	3-Green
13B	Output	300	*Alarm	*ALM	This signal remains ON if the controller is normal. It will turn OFF if an alarm occurs.	3-Blue
14A		301	Ready	RDY	This signal will turn ON when the controller becomes ready.	3-Purple
14B		302	Positioning complete	PEND	This signal will turn ON once the actuator has moved to the target position and entered the positioning band.	3-Gray
			Write complete	WEND	This signal will turn ON when writing of position data is completed.	
15A		303	Home return complete	HEND	This signal is OFF when the power is input, and will turn ON when home return is completed.	3-White
15B		304	Servo ON output	SVON	This signal will turn ON when the servo is turned on, and turn OFF when the servo is turned off.	3-Black
16A		305	Teaching mode output	TCMD	This signal will remain ON during the teaching mode.	4-Brown
16B		306	System battery error	SSER	This signal will turn ON when the voltage of the system-memory backup battery drops to the voltage-low warning level.	4-Red
17A		307	Absolute battery error	ABER	This signal will turn ON when the voltage of the absolute-data backup battery drops to the voltage-low warning level.	4-Orange
17B	N		External power supply 0 V	N		4-Yellow

*: Contact B (always ON)

2. Parameters

To use the controller in the teaching mode, set other parameter No. 25 to “4.”
Position numbers are specified as binary codes according to the factory setting. To change the input mode to BCD, set a value “other than 0” in other parameter No. 25.

	No.	Parameter	Function
Other	25	Operation mode type	4: Teaching mode
	71	Positioner mode parameter 1	Position-number input mode specification (0: Binary, ? 0: BCD) * Default value: 0 (Binary)

3. Details of Each Input Signal

■ Start (CSTR)
When the OFF ? ON leading edge of this signal is detected, the controller will load the target point number specified by the 13-bit binary code consisting of PC1 through PC13, and perform positioning to the target position specified by the corresponding position data.
Before movement is started, the target position, speed and acceleration/deceleration operation data must be set in the position table using a PC or teaching pendant.
If this signal is input when no single home-return operation has been performed after the power was input (= when the HEND output signal is OFF), the actuator will perform home-return operation.

■ Position inputs 1 through 11 (PC1 through PC11)
When a movement command is executed upon the OFF ? ON edge of the start signal, the controller will load the command position number specified by the 11-bit binary code consisting of signals PC1 through PC11.
The weight of each bit is as follows: 2^0 for PC1, 2^1 for PC2, 2^2 for PC3, 2^3 for PC4, ..., and 2^{10} for PC11.
By combining these bits, any position number between 1 and 1500 (maximum) can be specified.
In the teaching mode, specify the position number for which the current position will be written.
When the PWRT input signal is turned ON, the current position will be written to the position number specified by the binary code.
Also, the input mode can be changed to BCD by changing the setting of other parameter No. 71, as follows:
Other parameter No. 71 = 1 (other than 0) (BCD input)
(Default setting of other parameter No. 71 = 0 (Binary input))
In the BCD input mode, specify the one’s place in PC1 through 4, and ten’s place in PC5 through 8 (position Nos. 1 to 99 can be specified).

■ Pause (*STP)
If this signal turns OFF while the actuator is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be held, which means that when the signal turns ON again, the actuator will resume movement of the remaining travel distance.
The pause signal can be used for the following purposes:
[1] As a sensor to detect entry into a specified area around the system or for other lower-level safety measures to stop the axis while the servo is on
[2] To prevent contact with other equipment
[3] For positioning based on sensor or LS signal detection
(Note) When this signal is input during home return, the movement command will be held if the actuator has not yet contacted the mechanical end. If the signal is input after the actuator has reversed upon contacting the mechanical end, home return will be performed again.

■ Servo ON (SON)
The servo remains on while this signal is ON.
Use this signal if servo ON/OFF control is required as part of the safety circuit for the entire system to be provided on the PLC side.
To operate the actuator using the start input/jog input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. (Only the commands will be ignored, and no error will generate.)
(Note) When this signal turns OFF while the actuator is moving, the actuator will not decelerate to a stop. It will complete the movement to the target position, after which the servo will turn off.

⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
Put the slider or rod away from the mechanical end when turning the servo ON.

■ Error reset (RES)
This signal is used to reset the alarm output signal (*ALM) that has been generated due to an error.
If an error occurred, check the content of the error and then turn this signal ON.
The error will be reset upon detection of the leading edge of the signal.
(Note) Errors of cold start and higher level cannot be reset using this signal. The power must be reconnected to reset these errors. For details, refer to Appendix, “Error Level Management.”

■ Teaching mode specification (MODE)
When this signal turns ON, the normal positioning mode will change to the teaching mode. When the new mode becomes effective, the TCMD output signal will turn ON.
Program the PLC so that it will accept PWRT/JOG1+ and other operation commands after confirming that the TCMD output signal is ON.
To return the controller to the normal positioning mode, turn this signal OFF.
Program the PLC so that it will accept operation commands in the normal positioning mode after confirming that the TCMD output signal is OFF.
The controller will not return to the positioning mode right away when this signal is turned OFF while the actuator is jogging. It will not immediately stop the actuator, either. The controller will complete the movement first, and then return to the positioning mode.
Exercise caution because the actuator will start moving if this signal is turned ON when the servo is on in the positioning mode while any jog input signal (JOG1+, JOG1-, etc.) is also ON.

■ Current position write (PWRT)
This signal is effective when the aforementioned TCMD output signal is ON.
If this signal has remained on for at least 20 msec, the controller will load the position number corresponding to the binary code specified by PC1 through PC11 as currently detected, and write the current position data in the corresponding target position field of the position data table.
If any of the data fields other than the target position (such as speed, acceleration/deceleration and positioning band) is not yet defined, the default value of the applicable parameter (all-axis parameter Nos. 11, 12 or 13) will be written in that field.
When the data write is successfully completed, the WEND output signal will turn ON.
Program the PLC so that it will turn this signal OFF once the WEND signal turns ON. When this signal turns OFF, the controller will turn OFF the WEND signal.
(Note) An error will generate if position data is written before home return is completed. Position data cannot be written while the actuator is jogging.

■ Axis 1 jog (JOG1+, JOG1-)
These signals are effective when the aforementioned MODES output signal is ON.
The actuator of axis 1 will move to the + or - soft limit position upon detection of the OFF → ON leading edge of each signal.
Although the actuator will be forcibly decelerated to a stop after reaching the soft limit, no alarm will generate.
The speed and acceleration/deceleration to be used are the values set in user parameter No. 26 (PIO jog speed) and No. 9 (Default acceleration/deceleration).
If both the JOG+ and JOG- signals turn ON at the same time, the actuator will move to the direction corresponding to the signal that was input first.
The actuator will decelerate to a stop upon detection of the ON → OFF trailing edge of the signal while the actuator is moving.
(Note) Exercise due caution not to perform jogging before home return is complete, because the soft limits are still invalid and the actuator may collide with the mechanical end.

■ Inching (IN001 through 1)
These signals are used to specify the inching distance for inching operation performed in the teaching mode.
The four bits of IN001 through 1 indicate different inching distances, as follows:
IN001: 0.01 mm, IN01: 0.1 mm, IN05: 0.5 mm, IN1: 1 mm
The actuator will perform inching operation when a jog movement command is input while the bit or bits corresponding to a given inching distance is/are ON (if all four bits are OFF, the actuator will jog).
When multiple bits are turned ON, the sum of the distances represented by the applicable bits will become the inching distance.

4. Details of Each Output Signal

■ Positioning complete (PEND)

This signal indicates that the actuator reached the target position and the positioning has completed.
The signal will turn ON when the servo has turned on after the main power was input, and the controller becomes ready.
Thereafter, this signal will turn OFF when the start signal is turned ON to execute a movement command.
The signal will turn ON if the position deviation from the target position is within the in-position band after the start signal has turned OFF.
Once this signal turns ON, it will not turn OFF even after the position deviation subsequently exceeds the in-position band.

(Note) If the start signal is ON, this signal will not turn ON even when the position deviation from the target position falls within the in-position band. The signal will turn ON after the start signal turns OFF.
The signal will remain OFF while the servo is off.

■ Home return complete (HEND)

This signal is OFF when the power is input. It will turn ON upon completion of home return (if the actuator is of incremental specification).
To perform home return, specify a desired position number, and then turn ON the start input signal.
Use this signal as a condition for moving the actuator and also for writing the current position in the teaching mode.

(Note) Actuators of incremental specification must always perform home return after the power is input.
In the teaching mode, the actuator can be jogged before it completes home return, but the soft limits are still ineffective. Since coordinate values have no meaning in this condition, exercise due caution not to let the actuator contact the stroke end.

Once this signal turns ON, it will not turn OFF until the input power is cut off or the home-return signal is input again.

■ Teaching mode specification (MODES)

This signal will turn ON when the teaching mode was selected by the teaching mode input signal (turning ON the MODE signal) and the teaching mode has become effective.
Thereafter, this signal will remain ON until the MODE signal turns OFF.
Program the PLC so that it will start teaching operation after confirming that this signal has turned ON.

■ Write complete (WEND)

This signal is effective only in the teaching mode.
The signal is OFF immediately after the controller has entered the teaching mode, and will turn ON upon completion of the position data write initiated by the current position write signal.
When the current position write signal turns OFF thereafter, this signal will also turn OFF.
Program the PLC so that it will recognize completion of write operation upon turning OFF of this signal.

■ Alarm (*ALM)

This signal remains ON while the controller is normal, and will turn OFF if an alarm occurs.
This signal will turn OFF when an error of operation-cancellation level or higher generates.
Program the PLC so that it will monitor this signal and implement appropriate safety measures to protect the entire system when the signal turns OFF.
For details on alarms, refer to Appendix "◎ Error Level Management" and "◎ Error List."

■ Ready (RDY)

This signal will turn ON when the initialization has completed successfully after the main power was input, and the controller enters the mode where it can control the actuator.
This signal will turn OFF when an error of cold level or higher generates.
Use this signal as a condition to start control on the PLC side.

■ Servo ON output (SVON)

This signal will turn ON when the servo turns on. Issue a movement command after the servo ON output signal has turned ON.

■ System battery error

This signal will turn ON when the voltage of the optional system-memory backup battery drops to a specified level.

■ Absolute battery error

On a controller of absolute specification, this signal will turn ON when the voltage of the absolute-data backup battery drops to a specified level.

5. Timing Chart

5.1 Recognition of I/O Signals

An input time constant is set for the input signals of this controller to prevent malfunction due to chattering, noise, etc.

Except for certain signals, the input signal will switch if the new signal level has remained for at least 6 [msec].

For example, when an input signal is turned ON, the controller will recognize that the signal is ON after elapse of 6 [msec]. The same applies when the signal is turned OFF. (Fig. 1)

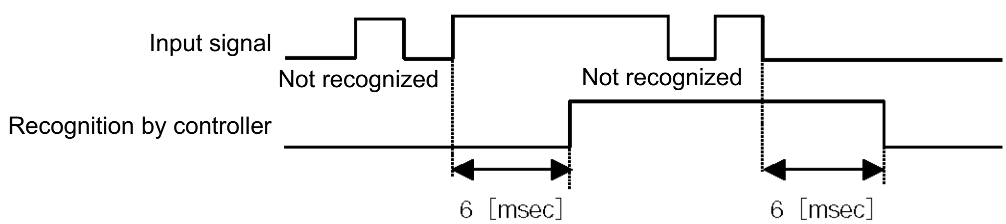
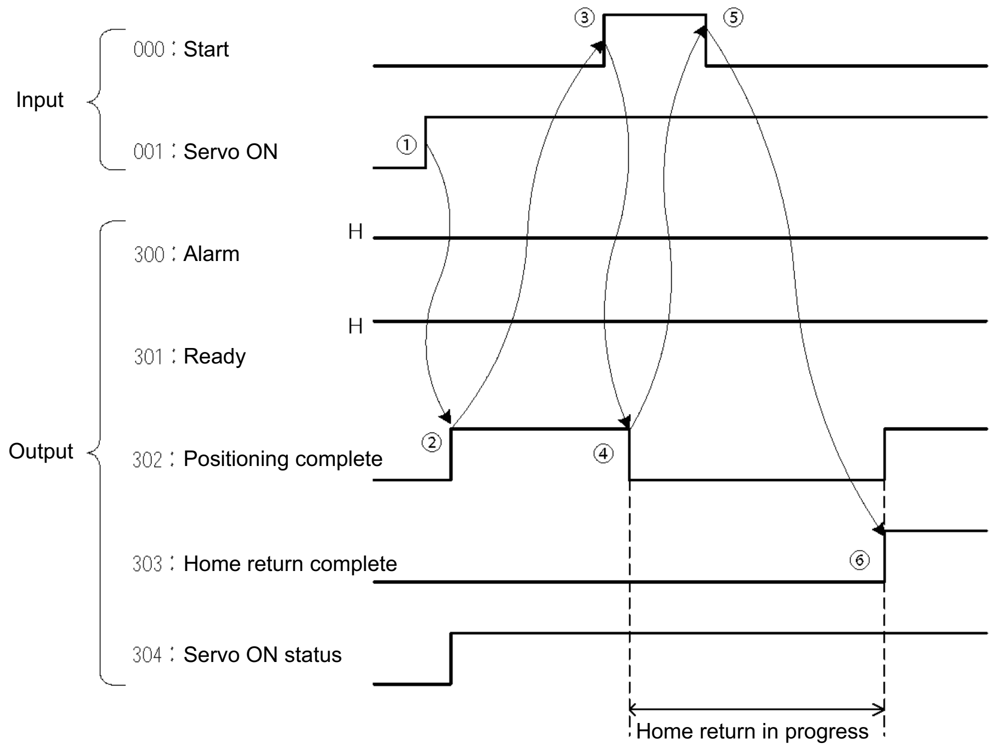


Fig. 1 Recognition of Input Signal

5.2 Home Return

In the teaching mode, no dedicated home-return input is available.
Home return will be performed when the start signal is input after specifying a desired position in a condition where home return is not yet completed.
Timings associated with home-return operation are illustrated below.



Timing Chart of Home-return Operation (Teaching Positioner Mode)

Perform home-return operation by following the procedure explained below.

* Before commencing the procedure, confirm that the ready output signal is ON, alarm output signal is OFF, and home-return complete output signal is OFF.
  - [1] Turn ON the servo ON input signal.
  - [2] Confirm that the servo-ON status output signal is ON.
  - [3] Turn ON the start input signal.
  - [4] Confirm that the positioning complete output signal is OFF.
  - [5] Turn OFF the start input signal.
  - [6] Confirm that the home-return complete output signal is ON. Home return is now completed.
* Pause input is a contact-B input signal (always ON), so keep this signal ON while home return is in progress.

To operate the actuator using the start input, the servo ON input signal must be ON. If the servo ON input signal is OFF, this operation command will not be accepted. Note, however, that only the command will be ignored and no error will generate.

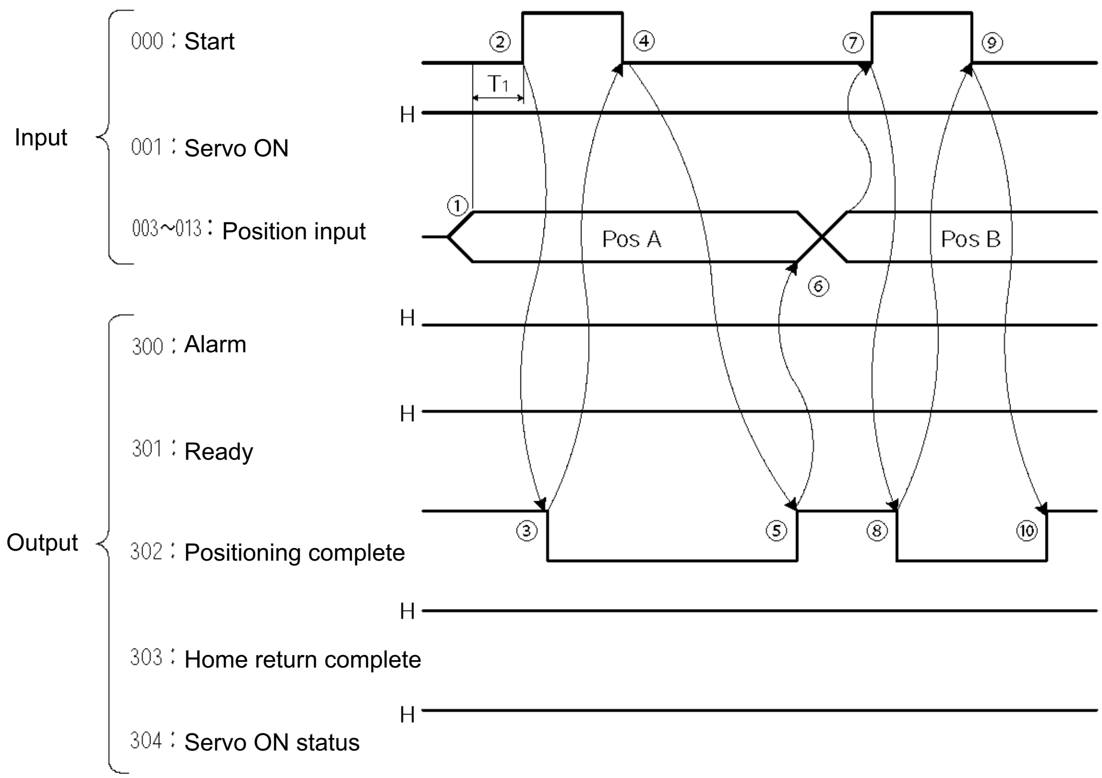
⚠Warning :	Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error. Put the slider or rod away from the mechanical end when turning the servo ON.

With the 2-axis specification, the controller has been configured at the factory so that the two axes will start home return simultaneously.
You can cause either axis to start home return earlier than the other axis by changing the applicable parameter setting. Specifically, change the setting in axis-specific parameter No. 13, “SIO/PIO home-return order” so that the parameter value for the axis number corresponding to the axis for which you want to complete home return first, will become smaller than the parameter value for the other axis number.

Example) Cause axis 1 to perform home return after axis 2 has completed home return, set “1” for axis 1 and “0” for axis 2 in axis-specific parameter No. 13.

5.3 Movements through Positions

Timings of how the actuator moves through positions are illustrated below.



Timing Chart of Movement through Positions (Standard Positioner Mode)
Ti: At least 6 msec

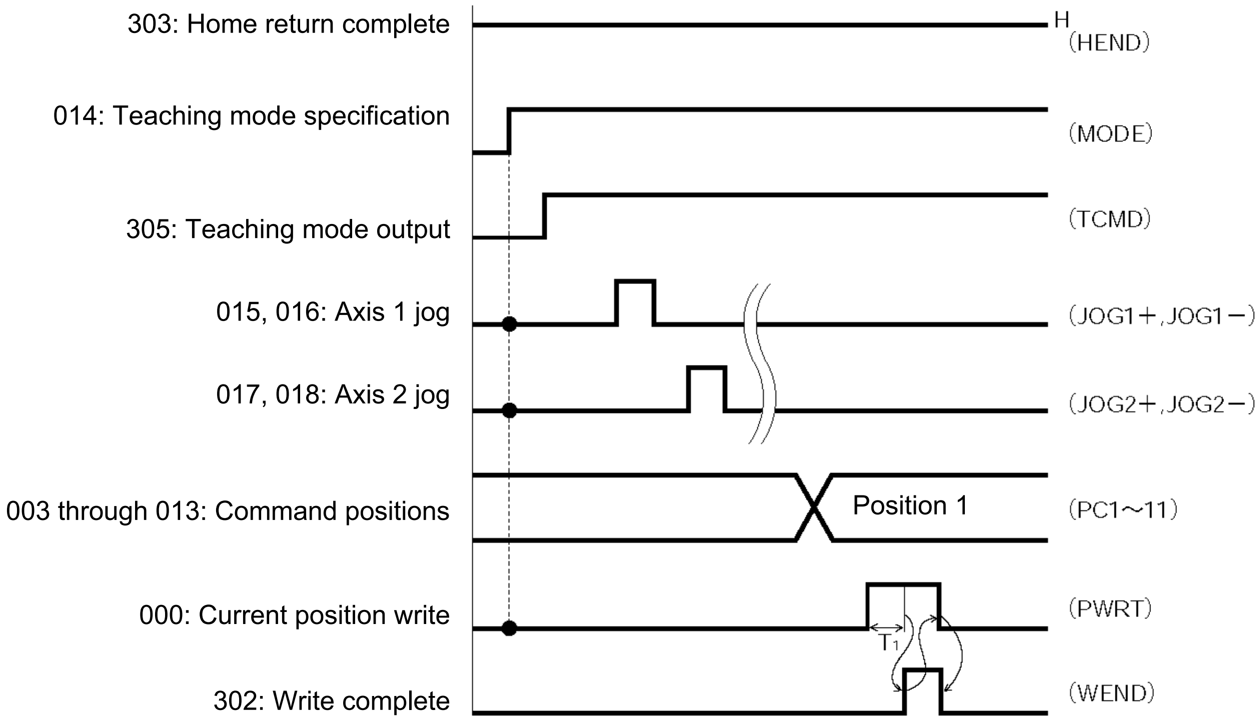
Operate the actuator to move through positions by following the procedure explained below.

* Confirm beforehand that the positioning complete output signal, home-return complete output signal and servo-ON status output signal are all ON.
  - [1] Change the previous position number input to a different position number.
  - [2] Turn ON the start input signal.
  - [3] Confirm that the positioning complete output signal is OFF.
  - [4] Turn OFF the start input signal.
  - [5] Confirm that the positioning complete output signal is ON.
    Repeat steps [1] through [5] sequentially.
* Pause input is a contact-B input signal (always ON), so keep this signal ON while home return is in progress.
* While the actuator is moving to the target position, only the operation pause signal is accepted. The servo cannot be turned off even if the servo ON input signal is turned OFF. (The servo can be turned off only when the positioning complete output signal is ON.)
* While the start input signal is ON, the positioning complete output signal will not turn ON even after the actuator physically completes moving to the target position. Therefore, always turn OFF the start input signal ([4]) to detect the completion of positioning.

* As for the positioning complete output signal and push-motion complete output signal, they will not be output until the start signal turns OFF (based on the I/O control handshake rules).

* For the actuator to operate upon start signal input, the servo ON input signal must be ON. If the servo ON input signal is OFF, these operation commands will not be accepted. Note, however, that only the commands will be ignored and no error will generate.

5.4 Timings in the Teaching Mode



T1: At least 20 msec. T1 represents the time after the position-information write input signal turns ON, until writing of the current position starts.

When the teaching mode specification (MODE) input signal is turned ON, the teaching mode (TCMD) output signal will turn ON. The controller will enter the teaching mode and jogging/teaching via PIOs will become possible.
To confirm if the controller is in the teaching mode, check if the TCMD signal is ON.
If both the jog+ and jog- input signals turn ON at the same time, the actuator will move to the position corresponding to the signal that was input first.
*Pause signal is a contact-B input signal (always ON), so keep this signal ON while teaching is in progress.
To perform inching, specify a desired inching distance (IC001 through 1) before the jog command is input.
If the current position write (PWRT) input signal has remained ON for at least 20 msec, the current actuator position will be written to the selected command position number.
Once the data write is complete, the write complete (WEND) output signal will turn ON. To confirm if the controller has finished writing data, check if the WEND signal is ON.
When the PWRT input signal turns OFF, the WEND output signal will turn OFF.

If the position table screen is open on the PC or teaching pendant, inputting a write signal from the PLC will not update the position data display. To check the acquired position data, do one of the following operations:

PC --- Click the button.

Teaching pendant --- Turn the PORT switch from OFF to ON.

Chapter 6 DS-S-C1 Compatible Mode

In this mode, the same I/O assignments used by the conventional controller model DS-S-C1 are used.
As added functions, the cancellation (CANC) input, interpolation setting input, system battery error output, and absolute battery error output are available, and the number of positions has been increased.

1. I/O Interface List

Pin No.	Category	Port No.	Signal name	Signal symbol	Function overview	Cable color
1A	P24		External power supply 24 V	P24		1-Brown
1B	Input	016	Position No. 1000 input	PC1000	(Same as PC1 through 800)	1-Red
2A		017	-			1-Orange
2B		018	-			1-Yellow
3A		019	-			1-Green
3B		020	-			1-Blue
4A		021	-			1-Purple
4B		022	-			1-Gray
5A		023	CPU reset	CPRES	The CPU will be restarted at the leading edge of this signal.	1-White
5B		000	Start	CSTR	The actuator will start moving at the leading edge of this signal.	1-Black
6A		001	Pause	STP	The actuator can be moved when this signal is ON, and will decelerate to a stop when the signal turns OFF.	2-Brown
6B		002	Cancellation	CANC	The remaining travel distance will be cancelled if this signal turns ON.	2-Red
7A		003	Interpolation setting	LINE	With the 2-axis specification, linear interpolation operation will start when the start input signal is turned ON while this signal is ON.	2-Orange
7B		004	Position No. 1 input	PC1	Input the position number corresponding to the position you want to move the actuator to. Be sure to specify a position input by no later than 6 msec before the start input signal turns ON. Position numbers are input as BCD codes. (PC1 through 8 indicate the one’s place, PC10 through 80 indicate the ten’s place, PC100 through 800 indicate the hundred’s place, and PC1000 indicates the thousand’s place.)	2-Yellow
8A		005	Position No. 2 input	PC2		2-Green
8B		006	Position No. 4 input	PC4		2-Blue
9A		007	Position No. 8 input	PC8		2-Purple
9B		008	Position No. 10 input	PC10		2-Gray
10A		009	Position No. 20 input	PC20		2-White
10B		010	Position No. 40 input	PC40		2-Black
11A		011	Position No. 80 input	PC80		3-Brown
11B		012	Position No. 100 input	PC100		3-Red
12A		013	Position No. 200 input	PC200		3-Orange
12B		014	Position No. 400 input	PC400		3-Yellow
13A		015	*Position No. 800 input	PC800		3-Green
13B	Output	300	*Alarm	ALM	This signal remains ON if the controller is normal. It will turn OFF if an alarm occurs.	3-Blue
14A		301	Ready	RDY	This signal will turn ON when the controller becomes ready.	3-Purple
14B		302	Positioning complete	PEND	This signal will turn ON once the actuator has moved to the target position and entered the positioning band.	3-Gray
15A		303	-			3-White
15B		304	-			3-Black
16A		305	-			4-Brown
16B		306	System battery error	SSER	This signal will turn ON when the voltage of the system-memory backup battery drops to the voltage-low warning level.	4-Red
17A		307	Absolute battery error	ABER	This signal will turn ON when the voltage of absolute-data backup battery drops to the voltage-low warning level.	4-Orange
17B	N		External power supply 0 V	N		4-Yellow

Caution: The power wiring polarities are reversed from those of the PNP specification applicable to the old DS-S-C1 controller. As shown above, pin Nos. 1A and 17B are connected to 24 V and 0 V, respectively, even in the PNP specification.

2. Parameters

To use the controller in the DS-S-C1 compatible mode, set other parameter No. 25 to “16.”
Other parameter No. 25 = 16, “DS-S-C1 compatible mode”

3. Details of Each Input Signal

■ Start (CSTR)

The actuator will start moving to the position corresponding to the specified position data upon detection of the OFF → ON leading edge of this signal. Position numbers are specified using a 13-bit BCD code consisting of PC1 through 1000.
Before movement is started, the target position, speed and acceleration/deceleration must be set as position data. Use a PC (software) or teaching pendant to set position data.
Turn on the power, specify position No. 0 (PC1 through 1000 are all OFF) and then turn this signal ON, and the actuator will start home return.
If a movement command is executed when no single home-return operation has been performed after the power was input, “C6F, Home-return incomplete error” will generate.

■ Position Nos. 1 through 1000 (PC1 through 1000)

When a movement command is executed upon OFF → ON of the start signal, the controller will load the command position number specified by the 13-bit BCD code consisting of PC1 through 1000.
A desired position number between 1 and 1500 can be specified. Specify the one’s place in PC1 through 8, ten’s place in PC10 through 80, hundred’s place in PC100 through 800, and thousand’s place in PC1000.

An example of position number specification based on ON/OFF levels of PC1 through 1000 is shown below.

PC 1000	PC 800	PC 400	PC 200	PC 100	PC 80	PC 40	PC 20	PC 10	PC 8	PC 4	PC 2	PC 1	Position No.
0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	0	0	0	1	1	3
0	0	0	0	0	0	0	0	1	0	0	1	0	12
0	0	1	1	0	0	1	1	1	1	0	0	0	678
1	0	0	1	0	0	0	1	1	0	1	0	0	1234

■ Pause (STP)

If this signal turns ON while the actuator is moving, the controller will cause the actuator to decelerate to a stop.
The remaining travel distance will be held, which means that when the signal turns OFF again, the actuator will resume movement of the remaining travel distance.
To cancel the movement command altogether after turning ON the pause signal, turn ON the cancellation signal while this signal is ON to cancel the remaining travel distance.
The pause signal can be used for the following purposes:

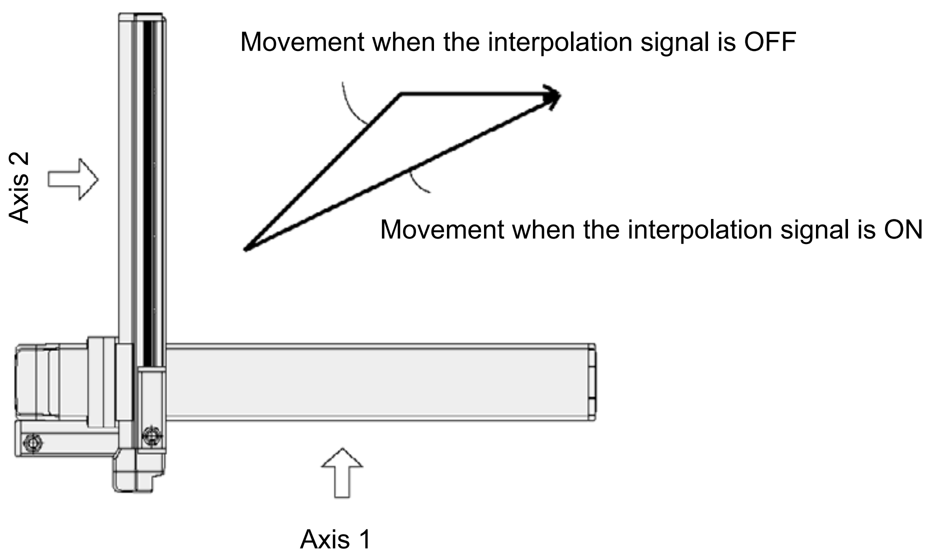
[1] As a sensor to detect entry into a specified area around the system or for other lower-level safety measures to stop the axis while the servo is on
[2] To prevent contact with other equipment
[3] For positioning based on sensor or LS signal detection

(Note) When this signal is input during home return, the movement command will be held if the actuator has not yet contacted the mechanical end. If the signal is input after the actuator has reversed upon contacting the mechanical end, home return will be performed again.

■ Cancellation (CANC)
If this signal turns ON while the actuator is moving, the controller will cause the actuator to decelerate to a stop. The remaining travel distance will be cancelled and the movement will not resume even when the signal turns OFF thereafter.

■ CPU reset (CPRES)
This input signal is used to restart the controller.
If an error occurs, identify and eliminate the cause, and then turn this signal ON.

■ Interpolation (LINE)
With the 2-axis specification, input of the position signal and start signal while this signal is ON will cause the two axes to perform interpolation operation (the two axes will start simultaneously and arrive at the target position simultaneously).
To perform interpolation operation, turn ON the interpolation input signal before turning ON the start input signal.



4. Details of Each Output Signal

■ Ready (RDY)

This signal will turn ON when the initialization has completed successfully after the main power was input, and the controller enters the mode where it can control the actuator.
This signal will turn OFF when an error of cold level or higher generates.
Use this signal as a condition to start control on the PLC side.

■ Alarm (ALM)

This signal remains OFF while the controller is normal, and will turn ON if an alarm occurs.
Program the PLC so that it will monitor this signal and implement appropriate safety measures to protect the entire system when the signal turns ON.
For details on alarms, refer to 10, "Troubleshooting."

■ Positioning complete (PEND)

This signal indicates that the actuator reached the target position and the positioning has completed.
When a movement command is executed by turning ON the start signal, this signal will turn OFF.
Thereafter, it will turn ON when the position deviation from the target position has entered the in-position band regardless of whether the start signal is ON or OFF.
Once this signal turns ON, it will not turn OFF even after the position deviation subsequently exceeds the in-position band.
(Note) Even if the motor is stopped, this signal will remain OFF if a pause signal is input or the servo is off.
This signal is OFF when the power is input. It will turn ON upon completion of home-return operation (if the actuator is of incremental specification).

■ System battery error

This signal will turn ON when the voltage of the optional system-memory backup battery drops to a specified level.

■ Absolute battery error

On a controller of absolute specification, this signal will turn ON when the voltage of the absolute-data backup battery drops to a specified level.

5. Timing Chart

5.1 Recognition of I/O Signals

An input time constant is set for the input signals of this controller to prevent malfunction due to chattering, noise, etc.
Except for certain signals, the input signal will switch if the new signal level has remained for at least 6 [msec].
For example, when an input signal is turned ON, the controller will recognize that the signal is ON after elapse of 6 [msec]. The same applies when the signal is turned OFF. (Fig. 1)

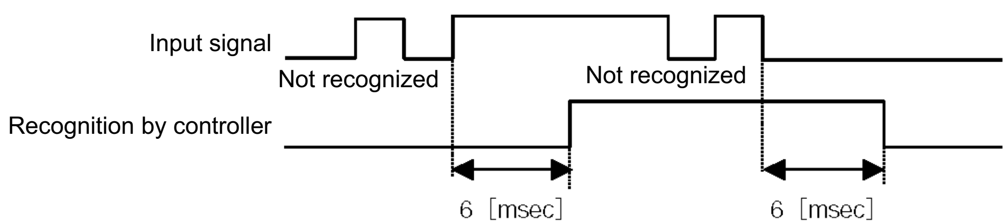
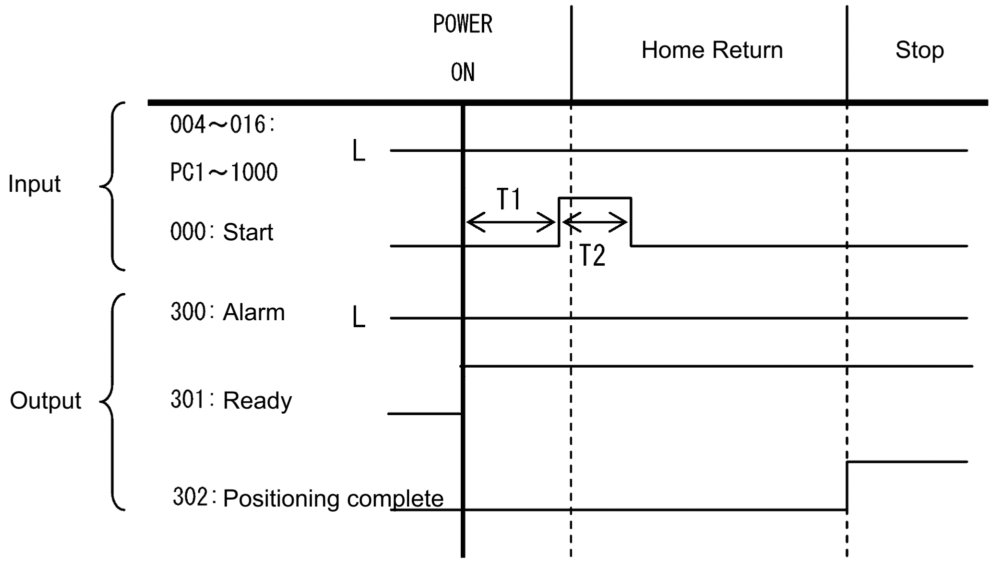


Fig. 1 Recognition of Input Signal

5.2 Home Return

In the DS-S-C1 compatible mode, no dedicated home-return input is available.
Home return will be performed when the start signal is input after specifying position No. 0. The positioning complete output signal is OFF after the power is input when home return is not yet completed.
Timings associated with home-return operation are illustrated below.



T1: Time after the ready output signal turns ON until input of the start signal becomes possible (50 msec or more)
T2: Start signal input (50 msec or more)

Timing Chart of Home-return Operation (Positioner Mode)

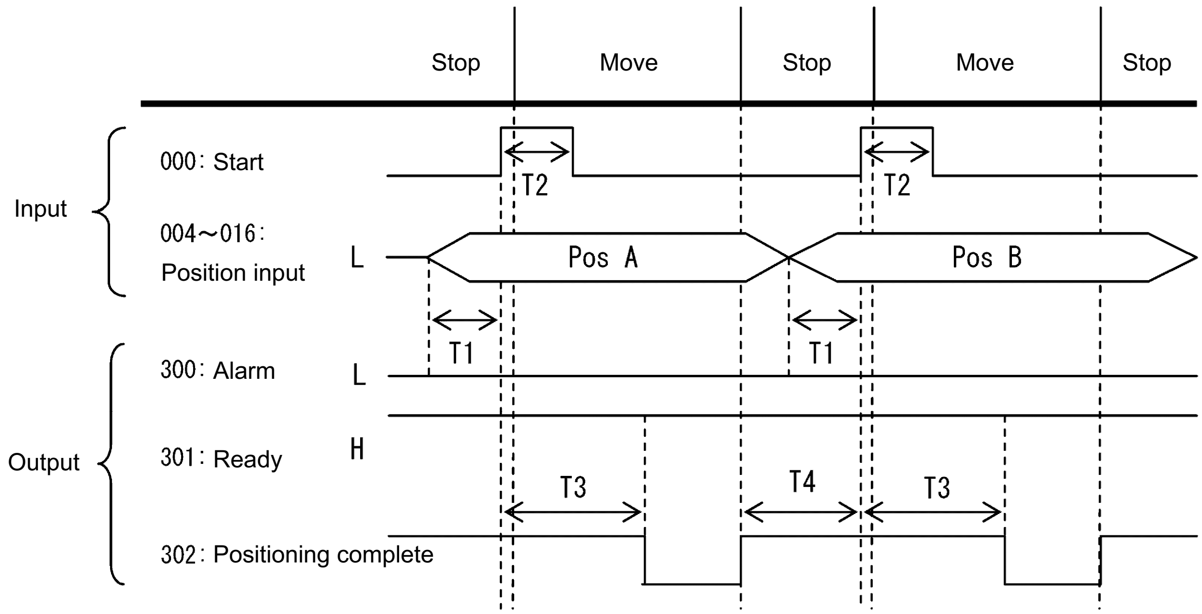
Perform home-return operation by following the procedure explained below.
* Before commencing the procedure, confirm that the ready output signal and alarm output signal are ON.
[1] Specify position No. 0 (PC1 through 1000 are all OFF).
[2] Turn ON the start input signal. (The signal should remain ON continuously for 30 msec or more (T2).)
[3] Turn OFF the start input signal.
[4] Confirm that the positioning complete output signal is ON. Home return is now completed.

> ⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

With the 2-axis specification, the controller has been configured at the factory so that the two axes will start home return simultaneously.
You can cause either axis to start home return earlier than the other axis by changing the applicable parameter setting.
Specifically, change the setting in axis-specific parameter No. 13, "SIO/PIO home-return order" so that the parameter value for the axis number corresponding to the axis for which you want to complete home return first, will become smaller than the parameter value for the other axis number.
Example) Cause axis 1 to perform home return after axis 2 has completed home return, set "1" for axis 1 and "0" for axis 2 in axis-specific parameter No. 13.

5.3 Movements through Positions

Timings of how the actuator moves through positions are illustrated below.



Timing Chart of Movement through Positions (Positioner Mode)

T1: Time after the position number signal is input until input of the start signal becomes possible (30 msec or more)
T2: Start signal input (30 msec or more)
T3: Time after the start signal turns ON until the positioning complete output signal turns OFF (60 msec or less)
T4: Time after the previous positioning complete output signal turns ON until input of the next start signal becomes possible (50 msec or more)

Operate the actuator to move through positions by following the procedure explained below.

* Confirm beforehand that the positioning complete output signal, home-return complete output signal and servo-ON status output signal are all ON.

[1] Change the previous position number input (BCD input) to a different position number.
[2] Turn ON the start input signal. (The signal should remain ON continuously for 30 msec or more (T2).)
[3] Turn OFF the start input signal.
[4] Wait for T3 after [2].
[5] Confirm that the positioning complete output signal is ON.
Repeat steps [1] through [5] sequentially.

* To perform interpolation operation, turn ON the interpolation setting input signal at least 30 msec before turning ON the start input signal. Turn OFF the interpolation signal after the start input signal has turned OFF.

* The positioning complete output signal turns ON when the actuator completes moving to the specified position, regardless of whether the start input signal is ON or OFF.

* Take note that the time after the start signal turns ON until the positioning complete output signal turns OFF is 60 msec or less, which is different from 15 msec or less with the DS-S-C1 controller.

Caution:	Unlike in other modes, the pause input and cancellation input are contact-A input signals (always OFF). The alarm output is also a contact-A output signal (always OFF) unlike in other modes.

⊙ List of Applicable Actuator Specifications

■ Slider type

Type	Stroke (mm) and maximum speed (mm/sec) *1	Load capacity Horizontal (kg)	Load capacity Vertical (kg)	Rated acceleration Horizontal (G)	Rated acceleration Vertical (G)
RCA-SA4C-I-20-10-***	50–400: 665	4	1	0.3	0.3
RCA-SA4C-I-20-5-***	50–400: 330	6	2.5	0.3	0.3
RCA-SA4C-I-20-2.5-***	50–400: 165	8	4.5	0.2	0.2
RCA-SA5C-□-20-12-***	50–450: 800; 500: 760	4	1	0.3	0.3
RCA-SA5C-□-20-6-***	50–450: 400; 500: 380	8	2	0.3	0.3
RCA-SA5C-□-20-3-***	50–450: 200; 500: 190	12	4	0.2	0.2
RCA-SA6C-□-30-12-***	50–450: 800; 500: 760; 550: 640; 600: 540	6	1.5	0.3	0.3
RCA-SA6C-□-30-6-***	50–450: 400; 500: 380; 550: 320; 600: 270	12	3	0.3	0.3
RCA-SA6C-□-30-3-***	50–450: 200; 500: 190; 550: 160; 600: 135	18	6	0.2	0.2

■ Rod type

Type	Stroke (mm) and maximum speed (mm/sec) *1	Rated thrust (N)	Maximum push force (N)	Load capacity Horizontal (kg)	Load capacity Vertical (kg)	Rated acceleration Horizontal (G)	Rated acceleration Vertical (G)
RCA-RA3C-I-20-10-***	50–200: 500	36.2	–	4	1.5	0.3	0.3
RCA-RA3C-I-20-5-***	50–200: 250	72.4	–	9	3	0.3	0.3
RCA-RA3C-I-20-2.5-***	50–200: 125	144.8	–	18	6.5	0.2	0.2
RCA-RA4C-□-20-12-***	50–300: 600	18.9	–	3	1	0.3	0.3
RCA-RA4C-□-20-6-***	50–300: 300	37.7	–	6	2	0.3	0.3
RCA-RA4C-□-20-3-***	50–300: 150	75.4	–	12	4	0.2	0.2
RCA-RA4C-□-30-12-***	50–300: 600	28.3	–	4	1.5	0.3	0.3
RCA-RA4C-□-30-6-***	50–300: 300	56.6	–	9	3	0.3	0.3
RCA-RA4C-□-30-3-***	50–300: 150	113.1	–	18	6.5	0.2	0.2

■ Arm type

Type	Stroke (mm) and maximum speed (mm/sec) *1	Thrust (N)	Load capacity Horizontal (kg)	Load capacity Vertical (kg)	Rated acceleration Horizontal (G)	Rated acceleration Vertical (G)
RCA-A4R-□-20-10-***	50–200: 300	39.2	–	2.5	–	0.2
RCA-A4R-□-20-5-***	50–200: 165	78.4	–	4.5	–	0.2
RCA-A5R-□-20-12-***	50–300: 400	33.3	–	2	–	0.2
RCA-A5R-□-20-6-***	50–300: 200	65.7	–	4	–	0.2
RCA-A6R-□-30-12-***	50–300: 400	48.4	–	3	–	0.2
RCA-A6R-□-30-6-***	50–300: 200	96.8	–	6	–	0.2

■ Dustproof/splashproof type

Type	Stroke (mm) and maximum speed (mm/sec) *1	Maximum push force (N)	Load capacity Horizontal (kg)	Load capacity Vertical (kg)	Rated acceleration Horizontal (G)	Rated acceleration Vertical (G)
RCA-RA3C-I-20-10-***	50–200: 500	–	4	1.5	0.3	0.3
RCA-RA3C-I-20-5-***	50–200: 250	–	9	3	0.3	0.3
RCA-RA3C-I-20-2.5-***	50–200: 125	–	8	6.5	0.2	0.2

*Note 1 Each band indicates applicable strokes, while the number in the band represents the maximum speed corresponding to each stroke.

Appendix

◎ Battery Backup Function

The ASEL controller uses the following two batteries.

- System-memory backup battery (optional)
  The optional battery is available for backing up position data, SEL program variables and other data.

- Absolute-data backup battery
  A separate battery is used to retain the absolute encoder's rotation data, so that the motor rotation data can be retained/refreshed when the controller power is cut off. A controller specified with an absolute-type actuator is shipped with the absolute-data backup battery.

  Each battery is explained in details.

1. System-Memory Backup Battery

The system-memory backup battery can be installed on the top face of the controller so that the data stored in the ASEL controller's SRAM will be retained even after the power is cut off.
Data to be backed up include controller parameters, SEL language variable data (global variables), position table data, and error list. The stored data will be retained even after the power is cut off.
(Use of the system-memory backup battery must be specified in the applicable controller parameter (other parameter No. 20 = 2).)

<Battery Replacement>
To replace the system-memory backup battery, disconnect the battery connector on the top face of the controller, and change the battery in the battery holder with a new battery.
It is recommended that you set a replacement schedule and replace the battery regularly.

The battery must be replaced as soon as the controller's battery voltage monitor function generates a battery voltage low alarm.

After an alarm is detected, a battery error will occur in approx. 10 days at an ambient temperature of 20°C if the power is supplied to the controller continuously. Once a battery error occurs, the data will be physically lost in approx. four days.

If the controller is not operated, the above periods should be reduced to 80% at 20°C or to 25% at 40°C.

The controller is designed so that the data will not be lost for at least 30 minutes without a battery if the controller is not detecting a battery error. Keep in mind to complete the battery replacement—taking out the current battery from the battery holder and placing a new battery in the holder—within 30 minutes.

To prevent the risk of data loss, you can use the PC software to evacuate the data in the SRAM to the flash ROM and then reload the flash ROM data to the SRAM after a new battery is installed.

The battery specifications are shown in the table below.

List of System-Memory Backup Battery Functions

Battery type	AB-5 (by IAI)	
Battery voltage	3.6 V	
Current capacity	2000 mAH	
Switching voltage at momentary power failure	(Typical) 2.81 V (2.7 V ~ 2.93 V)	System reset detection voltage
Power-source voltage drop at backup	(Typical) 0.3 V	
Detection voltage for battery voltage low alarm	(Typical) 2.65 V ± 5%	
Detection voltage for battery voltage low error	(Typical) 2.37 V ± 5%	
Time after alarm detection until error detection (reference)	10 days at 20°C based on continuous operation; 8 days if the power is not supplied. 10 days at 40°C based on continuous operation; 2.5 days if the power is not supplied.	
Minimum data retention voltage	Min. 2.0 V (Varies depending on the SRAM characteristics.)	
Time after error detection until data loss (reference)	4 days at 20°C based on continuous operation; 3 days if the power is not supplied. 4 days at 40°C based on continuous operation; 1 day if the power is not supplied.	
Data protection time during battery replacement	30 minutes (Maximum retention time when no battery is installed in the battery holder)	Data is retained by the super capacitor inside the controller.
Guide on when to replace battery	Approx. 5 years	

2. Absolute-Data Backup Battery for Absolute Encoder

If the ASEL controller is to drive/control an absolute type actuator, an absolute-data backup battery must be installed in the controller.
An absolute encoder is designed to retain rotation data and detect rotations using the power supplied from the absolute-data backup battery, even when the controller's control power is not supplied. This allows the controller to resume positioning control immediately after the controller power is restored, without performing home return.

<Backup Time>
The recommended replacement interval for the absolute-data backup battery is two years. This may be a little misleading. It means that if the battery is left at an ambient temperature of 40°C, it will retain the stored data for two years. In normal operating conditions, the battery can retain data for a longer period. As a guide, the battery will last for around four years if the controller is used at an ambient temperature of 40°C with the controller powered up 50% of the time.

<Battery Replacement>
To replace the absolute-data backup battery, disconnect the battery connector at the bottom of the controller, and change the battery in the battery holder with a new battery.
It is recommended that the battery be replaced regularly in accordance with the frequency/duration of usage.
The battery must be replaced as soon as the controller's battery voltage monitor function generates a battery voltage low alarm.
After an alarm is detected, a battery error will occur in approx. 10 days at an ambient temperature of 20°C if the power is supplied to the controller continuously. Once a battery error occurs, operations can no longer be performed unless the battery is replaced and an absolute reset is performed.
If the controller is not operated, the above periods should be reduced to 70% at 20°C or to 60% at 40°C.
The controller is designed so that the data will not be lost for at least 15 minutes without a battery if the controller is not detecting a battery error. Remember to complete the battery replacement within 15 minutes (i.e., the controller should not be without a battery for more than 15 minutes).
To prevent data loss, you can use the PC software to evacuate the data in the SRAM to the flash ROM and then reload the flash ROM data to the SRAM after a new battery is installed.

The absolute data backup battery is replaced differently depending on whether a battery error has generated or not. If an error has not been detected, only the battery needs to be replaced and an absolute reset is not required. If an error has been detected, an absolute reset will be required.

The absolute encoder backup specifications are shown in the table below.

List of Absolute Encoder Backup Functions

Battery type	AB-5 (by IAI)	
Battery voltage	3.6 V	
Current capacity	2000 mAH	
Detection voltage for battery voltage low alarm	(Typical) 3.1 V 3.0 V ~ 3.2 V	
Detection voltage for battery voltage low error	(Typical) 2.5 V 2.3 V ~ 2.7 V	
Time after alarm detection until error detection (reference)	10 days at 20°C based on continuous operation; 7 days if the power is not supplied. 10 days at 40°C based on continuous operation; 2.5 days if the power is not supplied.	
Minimum data retention voltage	Min. 2.7 V (Varies depending on the encoder characteristics.)	
Time after error detection until data loss (reference)	With an absolute encoder, an absolute reset must be performed once an error is detected.	
Data protection time during battery replacement	15 minutes (Maximum retention time when no battery is installed in the battery holder)	Data is retained by the super capacitor inside the controller.
Guide on when to replace battery	Temperature 40°C, power supplied 0% of the time	2 years
	Temperature 40°C, power supplied 50% of the time	4 years

◎ Parameter Utilization

Functions not initially available on the controller can be added, or dedicated functions can be assigned to input/output ports, by changing the values of corresponding parameters. Before changing a given parameter, always read the applicable section in the parameter list.

If you have any question regarding changing the parameters, please contact IAI's Sales Engineering Section. After changing a parameter, record the new and old parameter settings.

If you have purchased the PC software, we recommend that you back up the parameters immediately after the controller is delivered and when the system incorporating the controller is started. Since a number of customizing settings use parameters, you should back up the parameters regularly as you back up the programs.
To make the new parameters effective, write them to the flash ROM and then execute a software reset or reconnect the power.

Parameter classification

Parameters are classified into the following seven types based on what they specify:

1. I/O parameters
2. Parameters common to all axes
3. Axis-specific parameters
4. Driver parameters
5. Encoder parameters
6. I/O devices
7. Other parameters

1. Utilization Examples of I/O Parameters

I/Os include general-purpose inputs/outputs and dedicated inputs/outputs. General-purpose inputs/outputs are used by the user in SEL programs for sending/receiving ON/OFF signals to/from peripherals, among others.
Dedicated inputs are turned ON/OFF externally to activate specific functions.
Dedicated outputs turn ON or OFF in specific conditions. (Dedicated outputs cannot be turned ON/OFF in SEL programs.)

(1) I/O parameters
A desired input/output port can be specified as a dedicated input/output or general-purpose input/output. Set an appropriate input function specification value in the I/O parameter (Input/output function selection n) corresponding to the input/output port number you want to set.

The relationship of input port numbers and I/O parameter numbers is shown below.

Input port number	000	001	002	003	004	005	006	007	008	009	010	011
I/O parameter number	30	31	32	33	34	35	36	37	38	39	40	41

Input port number	012	013	014	015	016	017	018	019	020	021	022	023
I/O parameter number	42	43	44	45	251	252	253	254	255	256	257	258

Output port number	300	301	302	303	304	305	306	307
I/O parameter number	46	47	48	49	50	51	52	53

Example 1) How to set input port No. 5 as an input to forcibly release the brake for axis 1
Change the input function specification value of I/O parameter No. 35, which corresponds to input port No. 5, to “22” (Axis 1 forced brake-release input).
I/O parameter No. 35 = 22

Example 2) How to set output port No. 307 as a servo-ON status output for axis 1
Change the output function specification value of I/O parameter No. 53, which corresponds to output port No. 307, to “24” (Axis 1 servo-ON status output).
I/O parameter No. 53 = 24

Example 3) How to set input port Nos. 21 and 22 as general-purpose inputs
Change the input function specification values of I/O parameter Nos. 256 and 257, which correspond to input port Nos. 21 and 22, respectively, to “0” (General-purpose input).
I/O parameter No. 256 = 0
I/O parameter No. 257 = 0

If the above parameter changes are made from their factory settings, the start-program number specification bits will change to the five bits represented by input port Nos. 16 through 20. The range of program numbers that can be specified will become 1 to 19.

(2) Explanation of input function specification values

Input function specification value 0: General-purpose input
The applicable input can be used freely in programs as a general-purpose input.

Input function specification value 1: Program start signal (BCD) (ON edge)
The applicable signal is set as a program start signal.
Once set, the signal can start the BCD program number specified by input function setting values 9 through 15.

Input function specification value 2: Program start signal (BIN) (ON edge)
The applicable signal is set as a program start signal.
Once set, the signal can start the binary program number specified by input function setting values 9 through 15.

Input function specification value 3: Soft reset signal (ON edge)
Allow the applicable signal to restart the controller in the event of an error, etc.
Note 1: The input signal must remain ON for at least 1 second.
Note 2: The coordinate values will be cleared, so home return must be performed again.

Input function specification value 4: Servo ON
Allow the applicable signal to turn on the servo of a valid axis at its ON edge.
The signal will turn off the servo of a valid axis at its OFF edge.
Note: There must be an interval of at least 1.5 seconds between ON and OFF edges.

> ⚠Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error.
> Put the slider or rod away from the mechanical end when turning the servo ON.

Input function specification value 5: Auto-start program start signal
If an auto-start program is set, this signal can be used to start the program.
The program will start at the ON edge of this signal, while all operations and programs will be aborted at the OFF edge.

Input function specification value 6: Soft interlock for all servo axes (OFF level)
The active programs will be paused when this signal turns OFF.
(Any moving axis will decelerate to a stop.)

Input function specification value 7: Operation-pause reset signal (ON edge)
Allow the applicable signal to reset the operation pause signal set by input function selection value 8.

Input function specification value 8: Operation pause signal (OFF level)
Allow the applicable signal to pause all valid axes.
Note: The pause will be reset at the ON edge of the operation-pause reset signal (specified by input function selection 7) after turning this signal ON.

Input function specification value 9: Start-program number specification bit 1 (least significant bit)
This bit specifies the least significant bit of a program number.
Note: Start-program number specification bits x (input function setting values 9 through 15) cannot be assigned discontinuously from the least significant bit or in descending order from the least significant bit.

Input function specification value 10: Start-program number specification bit 2
This bit specifies the second bit of a program number.

Input function specification value 11: Start-program number specification bit 3
This bit specifies the third bit of a program number.

Input function specification value 12: Start-program number specification bit 4
This bit specifies the forth bit of a program number.

Input function specification value 13: Start-program number specification bit 5
This bit specifies the fifth bit of a program number.

Input function specification value 14: Start-program number specification bit 6
This bit specifies the sixth bit of a program number.

Input function specification value 15: Start-program number specification bit 7
This bit specifies the seventh bit of a program number.

Input function specification value 16: Error reset (ON edge)
This signal is used to reset errors.
Note: Only errors of operation-cancellation level or lower can be reset using this signal.

Input function specification value 17: Drive-source cutoff reset input (ON edge) (Effective when the problem factor has been removed)
This signal is used as a drive-source cutoff reset input when the emergency stop/enable switch recovery type is set to “Operation continued.”

Input function specification value 18: Home-return command signal for all valid axes (ON edge)
This signal commands home return of all valid axes.
Note: The servo ON input signal (input function specification value 4) must be turned ON first.

Input function specification value 19: Home-return command signal for all incremental axes (ON edge)
This signal commands home return of all incremental axes.
Note: The servo ON input signal (input function specification value 4) must be turned ON first.

Input function specification value 20: PC/TP-servo movement command acceptance permission input
Movements can be permitted from the PC software or teaching pendant.

Input function specification value 21: Remote-mode control input
This signal can be used to switch between the AUTO mode and MANUAL mode.
Note: Switching is enabled only when the mode switch is set to “AUTO.”

Input function specification value 22: Axis 1 forced brake release
Forcibly release the brake (axis 1).
Note: This function is effective only when the brake switch is tilted down (NOM).

Input function specification value 23: Axis 2 forced brake release
Forcibly release the brake (axis 2).
Note: This function is effective only when the brake switch is tilted down (NOM).

Input function specification value 24 ~ 27: For future expansion
Not used.

(3) Explanation of output function specification values

Output function specification value 0:	General-purpose output The applicable output can be used freely in programs as a general-purpose output.
Output function specification value 1:	Operation-cancellation level or higher error output (ON) The signal will turn ON when an error of operation-cancellation level or higher generates.
Output function specification value 2:	Operation-cancellation level or higher error output (OFF) The signal will turn OFF when an error of operation-cancellation level or higher generates.
Output function specification value 3:	Operation-cancellation level or higher error + emergency stop output (ON) This error output signal and emergency-stop output signal will turn ON when an error of operation-cancellation level or higher generates.
Output function specification value 4:	Operation-cancellation level or higher error + emergency stop output (OFF) This error output signal and emergency-stop output signal will turn OFF when an error of operation-cancellation level or higher generates.
Output function specification value 5:	READY output (PIO trigger program operation enabled) A signal will be output after the check is completed following the controller power input. The signal will turn ON only when the controller is able to perform program operation.
Output function specification value 6:	READY output (Absence of operation-cancellation level or higher error) The function is the same as that of output function specification value 5, but absence of operation-cancellation level or higher error is added as a condition.
Output function specification value 7:	READY output (Absence of cold-start level or higher error) The function is the same as that of output function specification value 5, but absence of cold-start level or higher error is added as a condition.
Output function specification value 8:	Emergency stop output (ON) The output signal will turn ON when the emergency-stop input signal turns ON. The signal will turn OFF when the emergency stop is reset.
Output function specification value 9:	Emergency stop output (OFF) The output signal will turn OFF when the emergency-stop input signal turns ON. The signal will turn ON when the emergency stop is reset.
Output function specification value 10:	AUTO mode output A signal will be output during the AUTO mode.
Output function specification value 11:	Auto operation status output A signal will be output during auto program operation.
Output function specification value 12:	All-valid-axes home (= 0) output A signal will be output when all valid axes are at the 0-mm position.

Output function specification value 13: All-valid-axes home-return complete (coordinate confirmed) output
A signal will be output when all valid axes have completed home return.

Output function specification value 14: All-valid-axes preset home coordinate output
A signal will be output when all valid axes have completed home return.
The value set by axis-specific parameter No. 12, "Home preset value" is used as the home position.

Output function specification value 15: Voltage-low warning output for system-memory backup battery
A signal will be output when the voltage of the system-memory backup battery drops to approx. 2.6 V.

Output function specification value 16: Voltage-low warning output for absolute-data backup battery
A signal will be output when the voltage of the absolute-data backup battery drops to approx. 3.2 V.
Once an abnormal voltage level is detected, the signal will remain ON until a power ON reset or software reset is performed.

Output function specification value 17: Drive-source cutoff (SDN) notification output
The output port will turn OFF when the drive source is cut off.

Output function specification value 18 ~ 23: For future expansion
Not used.

Output function specification value 24: Axis 1 servo ON output
This signal is output while the axis 1 servo is ON.

Output function specification value 25: Axis 2 servo ON output
This signal is output while the axis 2 servo is ON.

Output function specification value 26 ~ 29: For future expansion
Not used.

2. Utilization Examples of Axis-specific Parameters

The following functions can be added to, or changed from the factory-set functions, by changing the values of the corresponding axis-specific parameters. Before changing a given parameter, always read the applicable section in the parameter list.

- Change the home return direction
- About the home-return method
- Set a home preset
- Set a home offset
- Home return is desired in vertical installation
- Apply length measurement correction
- Zone output

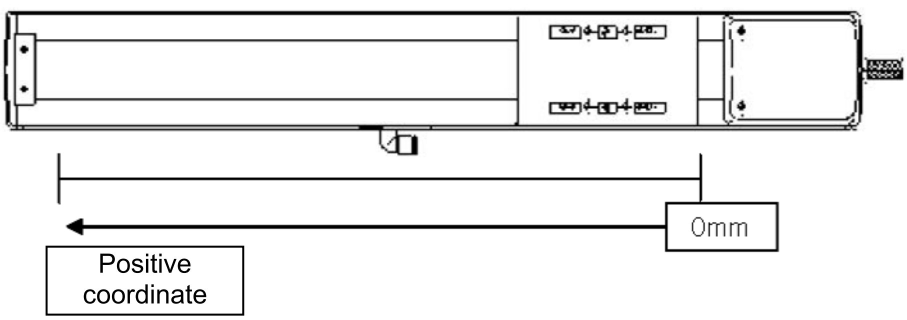
Change the home return direction
Axis-specific parameter No. 6, “Coordinate/physical-operation direction selection”

No.	Parameter name	Default value	Input range	Unit
6	Coordinate/physical-operation direction selection	1	0 ~ 1	None

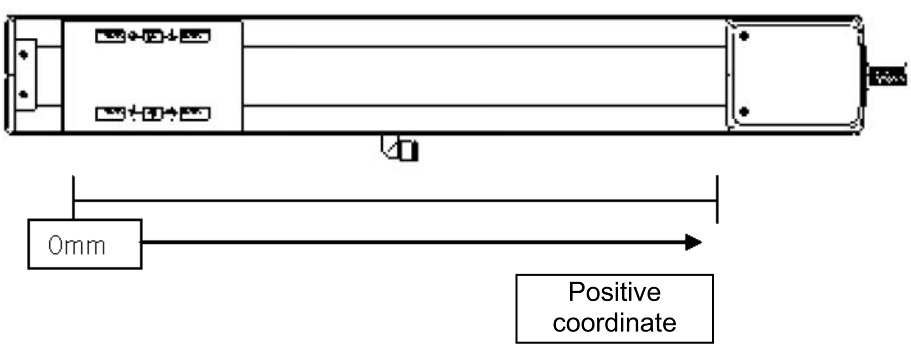
- Setting method
  A desired direction of home-return operation can be selected.

- Set value
  0: Motor CCW → Positive coordinate direction
  1: Motor CCW → Negative coordinate direction

Example 1: A linear axis whose home is at the standard position: When the parameter is set to “1”



Example 2: A linear axis whose home is at the standard position: When the parameter is set to “0”



Note: The default home return direction cannot be reversed on rod actuators simply by changing the parameter.

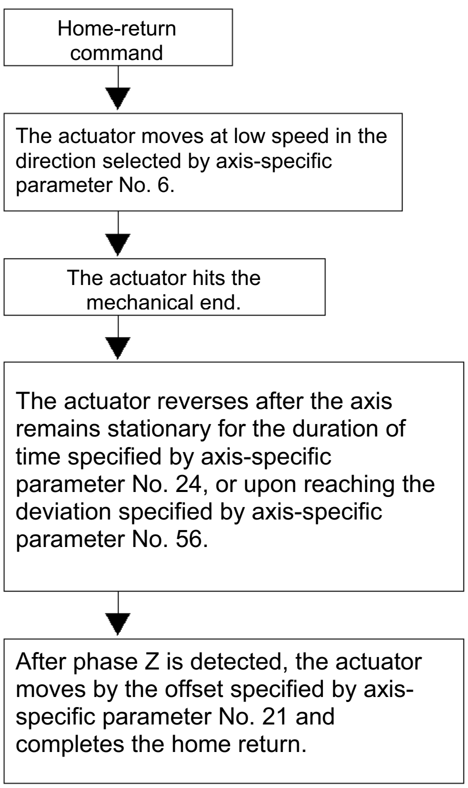
About the home-return method

Axis-specific parameter No. 10, "Home-return method"

No.	Parameter name	Default value	Input range	Unit
10	Home-return method	0	0 ~ 5	None

- Setting method
  Set a desired method to perform home return.

- Set value
  0: Search phase Z after end search
  The actuator performs normal home-return operation.



1: Current position 0 home (This setting can be specified only when an incremental encoder is used.)
The position at which the actuator is currently stopped is recognized as the home.
* Home-return operation is not performed.

2: Current position 0 home = Preset home (This setting can be specified only when an incremental encoder is used.)
Related parameter: Axis-specific parameter No. 12, "Home preset value"
The position at which the actuator is currently stopped is recognized as the home. (On the coordinate system, this position becomes the value set by the home preset parameter.)
* Home-return operation is not performed.

3 to 5: For future expansion

Set a home preset

Axis-specific parameter No. 12, “Home preset value”

No.	Parameter name	Default value	Input range	Unit
12	Home preset value	0	-99999999 ~ 99999999	0.001 mm

- Explanation of setting
  Set a value indicating where the actuator should be upon completing home return.
  (Normally, the actuator should be at 0-mm coordinate upon completing home return.)

- Set value
  Unit: 0.001 mm

Example 1: “Do not set” a home preset value

Home return complete $\rightarrow$ [0.000] mm is displayed.

Example 2: Set “3000” as a home preset value

Home return complete $\rightarrow$ [3.000] mm is displayed.

- Note
  Take note that when a home-return preset value is set, the effective stroke will also change. In particular, the stroke will decrease if the preset position is on the positive side of the default home.

Set a home offset

Axis-specific parameter No. 21, “Offset travel distance at home return”

No.	Parameter name	Default value	Input range	Unit
21	Offset travel distance at home return	1000	-99999999 ~ 99999999	0.001 mm

- Explanation of setting
  An offset can be set that will be applied after detecting phase Z (point 0) during home return.
  * If the home position has shifted after replacing the motor, jig, etc., use this parameter to adjust the home.

- Set value
  Setting unit: 0.001 mm
  Example:
  Set the offset to 0.5 mm = 500

- Note
  If the offset travel distance is near an integer multiple of the ball screw lead (such as 0, 6, 12 or 18 mm when the lead is 6 mm), the home will come directly above phase Z and thus rotation data may shift by one revolution upon absolute reset due to an “unstable” servo lock condition (a phenomenon where the coordinate values shift by one motor revolution). In this case, the position after home return will become the integer multiple of the lead length.
  * If the position after home return has become an integer multiple of the lead value, make adjustment using axis-specific parameter No. 12, “Home preset value.”

Home return is desired in vertical installation

Driver parameters No.39 “Push torque limit at home return”

No.	Parameter name	Default value	Input range	Unit
39	Push torque limit at home return	120	0 ~ 150	%

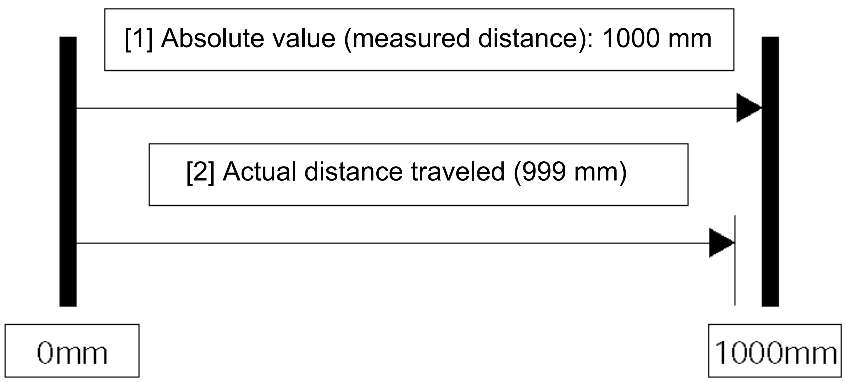
- Explanation of setting
  In case home return completes in front of the proper position because of the sliding resistance being increased in vertical installation due to how to affix the unit or the condition of load, check the value set in this parameter and adjust it if necessary.

- Set value
  The home return torque gets increased if the value is made bigger.
  Setting Reference: Set value = 120% (when direction of home return is upwards) to 80% (when direction of home return is downwards)

Apply length measurement correction
Axis-specific parameter No. 44, “Length measurement correction”

No.	Parameter name	Default value	Input range	Unit
44	Length measurement correction	0	-99999999 ~ 99999999	0.001 mm/1 M

- Explanation of setting
  Adjust the difference between the actual distance traveled and the measured distance, for the commanded travel distance.
  Example: Move the actuator from 0 mm to 1000 mm by specifying a position.



Correct the travel distance of [2] with respect to [1].
In the above example, enter “1000” because the actual distance traveled is 999 mm.
(Setting unit: 0.001/1 m)

* C10-class ball screws are subject to a margin of error of ± 0.21 mm per 300 mm.

Zone output

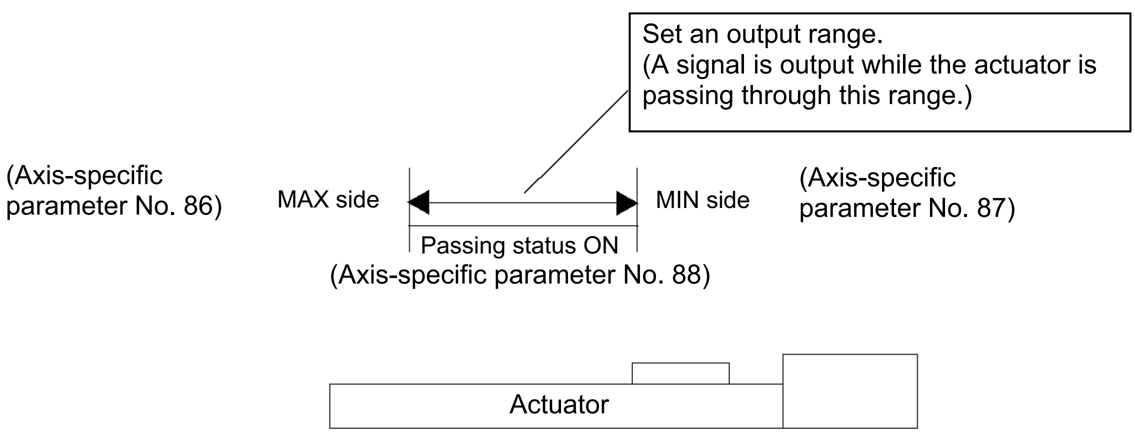
A signal can be output when the actuator has entered a desired zone specified by the user.
Three parameters must be set to specify a zone.
A zone is set for each axis.

No.	Parameter name	Default value	Input range	Unit
86	Zone 1 MAX	0	-99999999 ~ 99999999	0.001 mm
87	Zone 1 MIN	0	-99999999 ~ 99999999	0.001 mm
88	Zone 1 output number	0	0 ~ 899	None

Axis-specific parameter No. 86, “Zone 1 MAX”
Set the maximum limit of the zone, in units of 0.001 mm.
Example: To set 50 mm, set the value “50000.”

Axis-specific parameter No. 87, “Zone 1 MIN”
Set the minimum limit of the zone, in units of 0.001 mm.
Example: To set 10 mm, set the value “10000.

Axis-specific parameter No. 88, “Zone 1 output number”
Set an output port or flag number for the zone.
The output number set in this parameter cannot be used in programs.



- Note
  Set the zone so that the passing time through the zone will become at least 3 msec.

The zone output function allows four zones (zones 1 through 4) to be set for each axis.

No.	Parameter name	Default value	Input range	Unit
86	Zone 1 MAX	0	-99999999 ~ 99999999	0.001 mm
87	Zone 1 MIN	0	-99999999 ~ 99999999	0.001 mm
88	Zone 1 output number	0	0 ~ 899	None
89	Zone 2 MAX	0	-99999999 ~ 99999999	0.001 mm
90	Zone 2 MIN	0	-99999999 ~ 99999999	0.001 mm
91	Zone 2 output number	0	0 ~ 899	None
92	Zone 3 MAX	0	-99999999 ~ 99999999	0.001 mm
93	Zone 3 MIN	0	-99999999 ~ 99999999	0.001 mm
94	Zone 3 output number	0	0 ~ 899	None
95	Zone 4 MAX	0	-99999999 ~ 99999999	0.001 mm
96	Zone 4 MIN	0	-99999999 ~ 99999999	0.001 mm
97	Zone 4 output number	0	0 ~ 899	None

3. Parameter Utilization Examples (Reference)

	Description	Action	Parameter setting	Operation/outcome
1	Suppress generation of errors pertaining to the standard I/O board (so that trial operation can be performed before the board is wired, for example).	The I/O-board error monitor can be disabled to suppress error generation.	Set "0" in the I/O parameter corresponding to the I/O whose error monitor is to be disabled. Standard I/O: I/O parameter No. 10 = 0	To disable the error monitor of the standard I/O board, set "0" in I/O parameter No. 10. Note: Before operating the I/O board again, be sure to reset the parameter value to "1."
2	Implement a restart (software reset) using an external input signal.	A desired input port can be set as a restart input.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 3	The controller will be restarted when the specified port has remained ON for at least 1 second.
3	Turn on the servo using an external input signal. ⚠ Warning : Turning the servo ON near the mechanical end may disturb the magnetic pole phase detection, and may cause the magnetic pole unconfirmed error or the magnetic pole detection error. Put the slider or rod away from the mechanical end when turning the servo ON.	A desired input port can be set as a servo ON input.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 4	The servo will turn ON at the ON edge of the specified port. The servo will turn OFF at the OFF edge.
4	Start an auto-start program using an external input signal. (Under the default setting, the auto-start program will start when the power is input or the controller is restarted (by software reset) in the AUTO mode.) (The steps to start the auto-start program will increase.)	A desired input port can be set as an input for auto-program start signal.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 5 Other parameter No. 7 = 0	The program will start at the ON edge of the specified port. The program will end at the OFF edge.
5	Pause operations using an external input signal.	A desired input port can be set as a pause input. A desired input port can be set as a pause reset input.	Set the following value in each I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 7 I/O parameter "Input function specification value" = 8 Setting example) To set input port No. 5 as the pause reset input and input port No. 6 as the pause input, set "8" in I/O parameter No. 35 and "7" in I/O parameter No. 36.	Operations will pause at the OFF edge of the specified port set as the operation-pause signal input. Pause will be reset at the ON edge of the port set as the operation-pause reset signal input. (The port set as the operation-pause signal input is always ON.)
6	Reset errors (errors of operation-cancellation level or lower) using an external input signal.	A desired input port can be set as an error reset input.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 16	Errors will be reset at the ON edge of the specified port.
7	Perform home return using an external input signal.	A desired input port can be set as a home return input.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 18	Home return will be performed at the ON edge of the specified port. (The servo must be turned ON first.)

	Description	Action	Parameter setting	Operation/outcome
8	Enter program numbers as binary codes using input ports (default setting: BCD input).	Program numbers to be specified can be input as binary codes using the ports set as start-program number specification bits 1 through 7.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 2	
9	Check the level of each error currently present, using an output port.	Error levels can be checked based on the combination of the output function specification values (1 through 4, 5 through 7) and the ON/OFF levels of the applicable output ports.	Set the following value in each I/O parameter "Output function selection n" corresponding to the selected output port: I/O parameter "Output function specification value" = 2 I/O parameter "Output function specification value" = 7 (I/O parameter No. 46 and No. 47 have been set to "2" and "7," respectively, at the factory.)	Note) Factory-set parameters
10	Have emergency stop status notified via an output port.	Whether or not an emergency stop is currently actuated can be checked from the ON/OFF levels of the output ports for which function specification values of 8 and 9 are specified.	Set the following value in the I/O parameter "Output function selection n" corresponding to the selected output port: I/O parameter "Output function specification value" = 9	Note) Not set at the factory.
11	Output a signal during the AUTO mode.	A desired output port can be set as an AUTO mode output.	Set the following value in the I/O parameter "Output function selection n" corresponding to the selected output port: I/O parameter "Output function specification value" = 10	The specified port will turn ON during the AUTO mode.
12	Output a signal during auto operation.	A desired output port can be set as an auto operation status output.	Set the following value in the I/O parameter "Output function selection n" corresponding to the selected output port: I/O parameter "Output function specification value" = 11	The specified port will turn ON during auto operation.
13	How auto operation status is recognized during auto operation can be changed using the setting of other parameter No. 12.	• Auto operation status will be recognized if a program is running (regardless of the MANU or AUTO mode). • Auto operation status will be recognized if a program is running or when the controller is in the AUTO mode (regardless of whether or not a program is running). In either case, no all-operation-cancellation factor must be present. Auto operation status will be recognized based on one of the two conditions specified above.	• Other parameter No. 12 = 0 Auto operation will be recognized when a program is running. • Other parameter No. 12 = 1 Auto operation will be recognized when a program is running or the controller is in the AUTO mode. • "No all-operation-cancellation factor is present" refers to a condition in which no error of operation-cancellation level or higher is present AND no emergency stop signal is input AND no safety gate signal is input AND the deadman switch is ON (teaching pendant option).	
14	Output a signal when all valid axes are at their home.	A desired output port can be set as an all-valid-axis home position signal output. Note: Do not use a HOME command if the controller is of absolute specification.	Set the following value in the I/O parameter "Output function selection n" corresponding to the selected output port: I/O parameter "Output function specification value" = 12	The specified port will turn ON when all valid axes are at their home.

	Description	Action	Parameter setting	Operation/outcome
15	Output a signal when all valid axes have completed home return.	A desired output port can be set as an all-valid-axes home-return complete output.	Set the following value in the I/O parameter “Output function selection n” corresponding to the selected input port: I/O parameter “Output function specification value” = 13	The specified port will turn ON when all valid axes have completed home return.
16	Output a warning signal when the voltage of the system-memory backup battery became low.	A desired output port can be set as a voltage-low warning output for the system-memory backup battery.	Set the following value in the I/O parameter “Output function selection n” corresponding to the selected input port: I/O parameter “Output function specification value” = 15	The specified port will turn ON when the voltage of the system-memory backup battery became low.
17	Output a warning signal when the voltage of the absolute encoder battery became low.	A desired output port can be set as a voltage-low warning output for the absolute encoder battery.	Set the following value in the I/O parameter “Output function selection n” corresponding to the selected input port: I/O parameter “Output function specification value” = 16	The specified port will turn ON when the voltage of the absolute encoder battery became low.
18	Release the brake using an external input signal.	A desired input port can be set as a forced brake-release input.	Set the following value in each I/O parameter “Input function selection n” corresponding to the selected input port: I/O parameter “Input function specification value” = 22 (Axis 1) I/O parameter “Input function specification value” = 23 (Axis 2) Setting example) To set input port No. 12 as the brake release input for axis 2, set “23” in I/O parameter No. 42.	The brake will be forcibly released when the specified port turns ON.
19	Retain output conditions upon actuation of an emergency stop or opening of the safety gate.	The minimum and maximum output port numbers can be set to specify a range of outputs whose condition is to be retained.	I/O parameter No. 70 = Min. output port number I/O parameter No. 71 = Max. output port number Setting example) To retain the conditions of output port Nos. 303 through 307, set as follows: I/O parameter No. 70 = 303 I/O parameter No. 71 = 307	← The conditions of output port Nos. 303 through 307 will be retained when the emergency stop input turns ON or the safety gate opens.
20	Start a program when the emergency stop input turns ON or the safety gate opens. Programs that can be started in these conditions are limited to those not containing I/O processing, calculation processing or any other processing involving actuator operation (PIO processing programs).	A PIO processing program to be started in these conditions can be set. The program number of the applicable PIO processing program, and the minimum and maximum output port numbers indicating the range of processed outputs, are set by parameters.	Other parameter No. 2 = PIO processing program number Other parameter No. 70 = Min. output port number Other parameter No. 71 = Max. output port number Setting example) To start program No. 5 that processes output port Nos. 303 through 307, set as follows: Other parameter No. 2 = 5 Other parameter No. 70 = 303 Other parameter No. 71 = 307	← Program No. 5 will start when the emergency stop input turns ON or the safety gate opens. Output port Nos. 303 through 307 can be processed.

	Description	Action	Parameter setting	Operation/outcome
21	Switch between the AUTO mode and MANUAL mode using an input port.	A desired input port can be set as a mode switching input.	Set the following value in the I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 21	Set the mode switch to the "AUTO" side. The controller will switch to the AUTO mode when the specified input port turns OFF, and to the MANU mode when the port turns ON. If the mode switch is set to the "MANU" side, the controller will remain in the MANU mode regardless of the ON/OFF level of the input port.
22	Automatically restart the controller (effect a software reset) and start the auto-start program after an emergency stop has been reset.	The emergency-stop recovery type can be set to "Abort operations/programs (Software reset when the emergency stop is reset)."	Other parameter No. 10 = 3 Other parameter No. 7 = 1	After the emergency stop button has been reset, the controller will be reset (software reset will be effected) automatically and the auto-start program will start.
23	Automatically reset errors and start the auto-start program after an emergency stop has been reset.	The emergency-stop recovery type can be set to "Abort operations/programs (Error reset and auto-start program start when the emergency stop is reset)."	Other parameter No. 10 = 4 Other parameter No. 7 = 1 "17" must not be set as the "input function specification value" in the I/O parameter "Input function selection n."	After the emergency stop button has been reset, errors will be reset automatically and the auto-start program will start.
24	Continue to operate the actuator after an emergency stop has been reset (= resume actuator operation from immediately before the emergency-stop input signal turned ON). When an emergency-stop input signal is ON, all programs remain active and only programs involving actuator operation will be stopped. (When an emergency stop is actuated, all programs in which actuator operations are not specified will remain active. Programs in which actuator operations are specified will run until reaching a step in which an actuator operation command is specified.)	The emergency-stop recovery type can be set to "Operation continued." A desired port can be selected as a pause reset input. A desired port can be selected as a restart input.	Other parameter No. 10 = 2. Set the following value in each I/O parameter "Input function selection n" corresponding to the selected input port: I/O parameter "Input function specification value" = 7 I/O parameter "Input function specification value" = 3 (To ensure the specified operation cancellation method will work) Setting example) To set input port No. 5 as the pause reset input and input port No. 1 as the restart input, set "7" in I/O parameter No. 35 and "3" in I/O parameter No. 31.	After the emergency stop button has been reset, the actuator operation will resume at the ON edge of the port for which the input function specification value 7 (operation-pause reset signal) is set. To abort the remaining operation, do not allow the port for which the input function specification value 7 is set to receive an ON signal edge. Instead, turn ON for at least 1 second the port for which the input function specification value 3 (software reset signal) is set, in order to restart the controller.
25	Use the system-memory backup battery.	Install the optional system-memory backup battery.	Other parameter No. 20 = 2	When this setting is enabled, SEL global data and error list will be retained even after the main power is turned off.

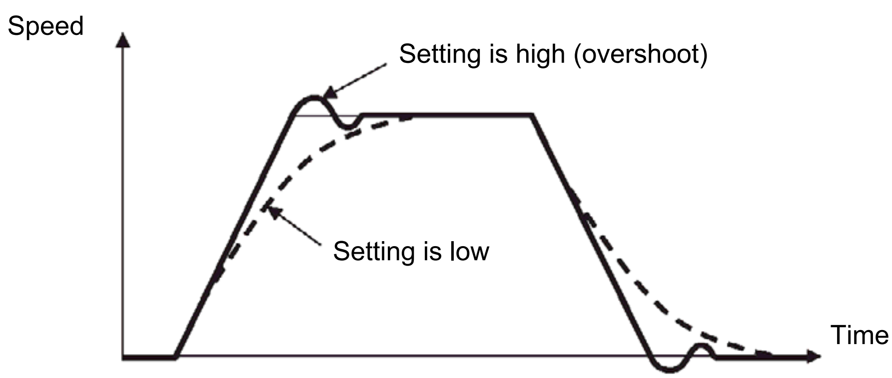
4. Servo Gain Adjustment

Since the servo has been adjusted at the factory in accordance with the standard specification of the actuator, the servo gain need not be changed in normal conditions of use.
However, vibration or noise may occur depending on how the actuator is affixed, specific load condition, and so on, and therefore the parameters relating to servo adjustment are disclosed to allow the customer to take quick actions should adjustment become necessary.
Particularly with custom models (whose ball screw lead or stroke is longer than that of the standard model), vibration/noise may occur due to external conditions.
In this case, the parameters shown below must be changed. Contact IAI for details.

- Position gain

Axis-specific parameter number	Unit	Input range	Default value (reference)
60	/sec	1 to 9999	30

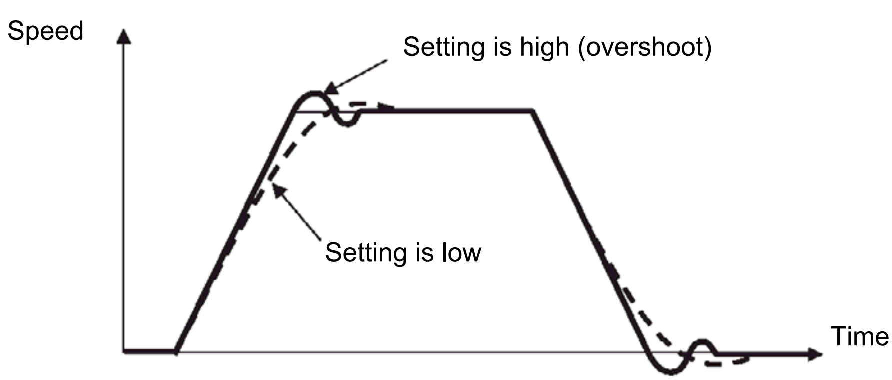
This parameter determines the level of response with respect to a position control loop.
Increasing the setting improves compliance with the position command.
However, increasing the setting too much increases the tendency of the actuator to overshoot.
If the setting is low, compliance with the position command drops and the positioning time increases as a result.



- Speed loop proportional gain

Driver parameter number	Unit	Input range	Default value (reference)
43	---	1 to 32767	500

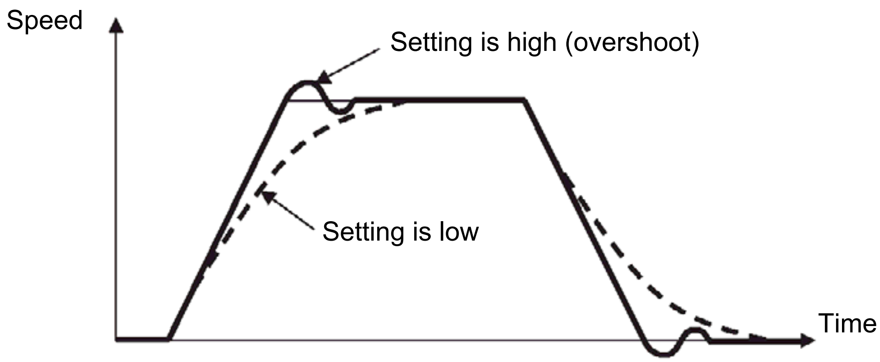
This parameter determines the level of response with respect to a speed control loop.
Increasing the setting improves compliance with the speed command (i.e., servo rigidity increases).
The greater the load inertia, the higher the setting should be.
However, increasing the setting too much increases the tendency of the actuator to overshoot or oscillate, resulting in increased mechanical vibration.



- Speed loop integral gain

Driver parameter number	Unit	Input range	Default value (reference)
44	---	1 to 3276700	1667

This parameter determines the level of response with respect to a speed control loop.
Decreasing the setting results in lower response to the speed command and decreases the reactive force upon load change. If the setting is too low, compliance with the position command drops and the positioning time increases as a result.
Increasing the setting too much increases the tendency of the actuator to overshoot or oscillate, resulting in increased mechanical vibration.



- Torque filter time constant

Driver parameter number	Unit	Input range	Default value (reference)
45	---	0 to 2500	0

This parameter determines the filter time constant applicable to the torque command.
If the mechanical resonance frequency is equal to or lower than the servo loop response frequency, the motor will vibrate.
This mechanical resonance can be suppressed by increasing the setting of this parameter.
It should be noted, however, that increasing the setting too much may affect the stability of the control system.

- Current control band number

Driver parameter number	Unit	Input range	Default value (reference)
46	---	0 to 4	4

This parameter sets the control band of the PI current control system.
Normally the default setting should not be changed.
If this parameter is changed carelessly, stability of the control system may be affected and a very dangerous situation may occur.
Changing this parameter may be effective in certain situations, such as when resonance noise occurs.
If you wish to change the setting of this parameter, please contact IAI.

◎ List of Parameters

If you have any question regarding changing the parameters, please contact IAI's Sales Engineering Section. After changing a parameter, record the new and old parameter settings.

If you have purchased the PC software, we recommend that you back up the parameters immediately after the controller is delivered and when the system incorporating the controller is started. Since a number of customizing settings use parameters, you should back up the parameters regularly as you back up the programs.
To make the new parameters effective, write them to the flash ROM and then execute a software reset or reconnect the power.
The lists below are examples of default values displayed on the PC software. The default parameter settings vary depending on the operating condition and actuators used.

The values in the "Input range" column represent input limitations on the teaching pendant or in PC software. For the actual settings, enter the values defined in the "Remarks" column.
Values other than those defined in the "Remarks" column are for future expansion, even when they are inside the input range.
Therefore, do not enter values other than those defined in the "Remarks" column.

1. I/O Parameters

1.1 I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
1	I/O port assignment type	1	0 ~ 20		0: Fixed assignment 1: Automatic assignment (Priority: Network I/F module → Standard I/O; * Ports are assigned only for the installed adjoining slots, starting from the standard I/O slot = For safety reasons)
2	Input port start number with fixed standard I/O assignments (I/O1)	000	-1 ~ 599		0 + (Multiple of 8) (Invalid if a negative value is set)
3	Output port start number with fixed standard I/O assignments (I/O1)	300	-1 ~ 599		300 + (Multiple of 8) (Invalid if a negative value is set)
4 ~ 9	For future expansion	-1	-1 ~ 599		
10	Standard I/O error monitor	1	0 ~ 5		0: Do not monitor 1: Monitor 2: Monitor (Do not monitor errors relating to 24-V I/O power source) 3: Monitor (Monitor only errors relating to 24-V I/O power source) * Some exceptions apply. * If this parameter is set to "0" (= Do not monitor) or "2" (= Do not monitor errors relating to 24-V I/O power source), a system error will not generate even when the 24-V I/O power source presents abnormality. However, the actual outputs from digital I/Os will be cut off by circuitry thereafter to protect the controller.
11 ~ 13	For future expansion	1	0 ~ 5		
14	Network system reservation	0	0 ~ 256		
15	Network system reservation	0	0 ~ 256		
16	Network system reservation	-1	-1 ~ 599		
17	Network system reservation	-1	-1 ~ 599		
18	Network system reservation	1	0 ~ 5		
19	(For expansion)	0			
20	Input filtering periods	2	1 ~ 9	msec	Input signal is recognized when the status is held for twice the period set by this parameter.
21	For future expansion (change prohibited)	0	1 ~ 9		
22	For future expansion	0	0 ~ 99999	msec	
23	For future expansion	0H	0H ~ FFFFFFFFH		
24	I/O setting bit pattern 1	10000H	0H ~ FFFFFFFFH		Bits 0 to 3: RDY OUT function selection (System IO) (0: SYSRDY (Software = PIO trigger program can be run) and hardware is normal (emergency stop has not been actuated and hardware error is not present) 1: Error of operation-cancellation level or higher is not present 2: Error of cold-start level or higher is not present) Bits 4 to 7: RDY LED function selection (0: Program can be run 1: Error of operation-cancellation level or higher is not present 2: Error of cold-start level or higher is not present) Bits 8 to 19: (For future expansion) Bits 20 to 23: ALM LED function selection (0: Error of message level or higher error is present 1: Error of operation-cancellation level or higher is present 2: Error of cold-start level or higher is present 3: Error of system-down level or higher is present)
25	I/O setting bit pattern 2	0H	0H ~ FFFFFFFFH		
26	(For expansion)	0			

I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
27	(For expansion)	0			
28	(For expansion)	0			
29	For future expansion	0	0 ~ 599		
30	Input function selection 000	1	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
31	Input function selection 001	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
32	Input function selection 002	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
33	Input function selection 003	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
34	Input function selection 004	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
35	Input function selection 005	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
36	Input function selection 006	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
37	Input function selection 007	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
38	Input function selection 008	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
39	Input function selection 009	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
40	Input function selection 010	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
41	Input function selection 011	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
42	Input function selection 012	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
43	Input function selection 013	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
44	Input function selection 014	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
45	Input function selection 015	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
46	Output function selection 300	2	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
47	Output function selection 301	7	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
48	Output function selection 302	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
49	Output function selection 303	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
50	Output function selection 304	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
51	Output function selection 305	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
52	Output function selection 306	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
53	Output function selection 307	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.

I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
54	Output function selection 308	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
55	Output function selection 309	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
56	Output function selection 310	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
57	Output function selection 311	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
58	Output function selection 312	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
59	Output function selection 313	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
60	Output function selection 314	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
61	Output function selection 315	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
62	For future expansion	0	0 ~ 299		
63	For future expansion	0	0 ~ 299		
64 ~ 67	For future expansion	0	0 ~ 299		
68	(For expansion)	0			
69	(For expansion)	0			
70	Unaffected general-purpose output area number (MIN) when all operations/programs are aborted	0	0 ~ 599		* Important: Outputs in this area must be operated under the responsibility of user programs including the "I/O processing program at operation/program abort." Outputs outside this area will be forcibly turned OFF. (Invalid if "0" is set)
71	Unaffected general-purpose output area number (MAX) when all operations/programs are aborted	0	0 ~ 599		
72	Unaffected general-purpose output area number (MIN) when all operations are paused (servo-axis soft interlock + output-port soft interlock)	300	0 ~ 599		* Important: Outputs in this area must be operated (including recovery) under the responsibility of user programs including the "I/O processing program at all operations pause." Outputs outside this area will be forcibly turned OFF, reflecting/holding the results of operations performed while all operation pause is effective (only during automatic operation). (Invalid if "0" is set)
73	Unaffected general-purpose output area number (MAX) when all operations are paused (servo-axis soft interlock + output-port soft interlock)	599	0 ~ 599		
74	Number of TP user output ports used (hand, etc.)	0	0 ~ 8		Referenced by TP. (Invalid if "0" is set)
75	TP user output port start number (hand, etc.)	0	0 ~ 599		Referenced by TP.
76	For future expansion	0	0 ~ 599		
77	For future expansion	0	0 ~ 299		
78	Axis pattern permitted to receive PC/TP servo movement command for	0	0B ~ 11111111B		
79	For future expansion	0	0 ~ 299		
80	(PC/TP SIO usage)	1	1 ~ 1		Switching of DIP switches
81	(PC/TP SIO station code)	153	153 ~ 153		Fixed to 153 (99H).
82	(PC/TP SIO reservation)	0			

I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
83	(PC/TP SIO reservation)	0			
84	(PC/TP SIO reservation)	0			
85	(PC/TP SIO reservation)	0			
86	(PC/TP SIO reservation)	0			
87	(PC/TP SIO reservation)	0			
88	(PC/TP SIO reservation)	0			
89	(PC/TP SIO reservation)	0			
90	Usage of SIO channel 0 opened to user (AUTO mode)	0	0 ~ 9		0: Open SEL program 1: Open SEL program (Connect PC/TP when both devices are closed = Used exclusively by the manufacturer) 2: IAI protocol B (Slave)
91	Station code of SIO channel 0 opened to user	153	0 ~ 255		Valid only with IAI protocol.
92	Baud rate type of SIO channel 0 opened to user	0	0 ~ 5		0: 9.6, 1: 19.2, 2: 38.4, 3: 57.6, 4: 76.8, 5: 115.2 kbps
93	Data length of SIO channel 0 opened to user	8	7 ~ 8		
94	Stop bit length of SIO channel 0 opened to user	1	1 ~ 2		
95	Parity type of SIO channel 0 opened to user	0	0 ~ 2		0: None 1: Odd 2: Even
96	Receive operation type of SIO channel 0 opened to user	0	0 ~ 1		0: Forcibly enable receive after send 1: Do not forcibly enable receive at send
97	IAI-protocol minimum response delay for SIO channel 0 opened to user	0	0 ~ 999	msec	Valid only with IAI protocol.
98	(Reservation of SIO channel 0 opened to user)	0			
99	(Reservation of SIO channel 0 opened to user)	0			
100 ~ 115	SIO system reservation	0	0H ~ FFFFFFFFH		
116	(For expansion)	0			
117	(For expansion)	0			
118	(For expansion)	0			
119	(For expansion)	0			
120	Network Attribute 1	1H	0H ~ FFFFFFFFH		Bit 0 to 3: System reservation Bit 4 to 11: Network link error check timer value (10msec) I/O Parameters No.18 = Effective only when set to 1 * When set to 0, generates system error immediately at network link error occurrence. Bit 12 to 23: System reservation Bit 24 to 27: Selection of Profibus H/L byte SWAP (0: without SWAP, 1: with SWAP)
121	Network Attribute 2	C80000H	0H ~ FFFFFFFFH		Bit 0 to 11: System reservation Bit 12 to 15: Future expansion Bit 16 to 27: Link timeout value at fieldbus initialization (100msec) (Main application version 0.36 or later)
122	Network Attribute 3	FH	0H ~ FFFFFFFFH		Bit 0 to 11: Value added for “PC·TP reconnection delay time at software reset” (sec) when fieldbus is used * Valid values 0 to 500(sec) (Main application Ver. 0.36 or later)
123	Network Attribute 4	0H	0H ~ FFFFFFFFH		
124	Network Attribute 5	0H	0H ~ FFFFFFFFH		
125	Network Attribute 6	1E32H	0H ~ FFFFFFFFH		
126	Network Attribute 7	7D007D0H	0H ~ FFFFFFFFH		
127	Network Attribute 8	5050214H	0H ~ FFFFFFFFH		
128	Network Attribute 9	0H	0H ~ FFFFFFFFH		

I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
129	Network Attribute 10	0H	0H ~ FFFFFFFFH		
130	Network system reservation	0H	Reference only (HEX)		
131	Network system reservation	0H	Reference only (HEX)		
132	Network system reservation	192	1 ~ 255		
133	Network system reservation	168	0 ~ 255		
134	Network system reservation	0	0 ~ 255		
135	Network system reservation	1	1 ~ 254		
136	Network system reservation	255	0 ~ 255		
137	Network system reservation	255	0 ~ 255		
138	Network system reservation	255	0 ~ 255		
139	Network system reservation	0	0 ~ 255		
140	Network system reservation	0	0 ~ 255		
141	Network system reservation	0	0 ~ 255		
142	Network system reservation	0	0 ~ 255		
143	Network system reservation	0	0 ~ 255		
144	Network system reservation	64511	1025 ~ 65535		
145	Network system reservation	64512	1025 ~ 65535		
146	Network system reservation	64513	1025 ~ 65535		
147	Network system reservation	64514	1025 ~ 65535		
148	Network system reservation	64515	1025 ~ 65535		
149	Network system reservation	192	0 ~ 255		
150	Network system reservation	168	0 ~ 255		
151	Network system reservation	0	0 ~ 255		
152	Network system reservation	100	0 ~ 254		
153	Network system reservation	64611	0 ~ 65535		
154	Network system reservation	192	0 ~ 255		
155	Network system reservation	168	0 ~ 255		
156	Network system reservation	0	0 ~ 255		
157	Network system reservation	100	0 ~ 254		
158	Network system reservation	64611	0 ~ 65535		
159	Network system reservation	64516	1025 ~ 65535		
160 ~ 169	(For network expansion)	0			
170 ~ 200	(For expansion)	0			
201 ~ 224	SIO system reservation	00000000H	0H ~ FFFFFFFFH		
225 ~ 250	(For expansion)	0			
251	Input function selection 016	9	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
252	Input function selection 017	10	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
253	Input function selection 018	11	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
254	Input function selection 019	12	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
255	Input function selection 020	13	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
256	Input function selection 021	14	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
257	Input function selection 022	15	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
258	Input function selection 023	3	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.

I/O Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
259	Input function selection 024	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
260	Input function selection 025	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
261	Input function selection 026	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
262	Input function selection 027	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
263	Input function selection 028	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
264	Input function selection 029	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
265	Input function selection 030	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
266	Input function selection 031	0	0 ~ 99		Input function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
267	Output function selection 316	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
268	Output function selection 317	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
269	Output function selection 318	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
270	Output function selection 319	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
271	Output function selection 320	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
272	Output function selection 321	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
273	Output function selection 322	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
274	Output function selection 323	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
275	Output function selection 324	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
276	Output function selection 325	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
277	Output function selection 326	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
278	Output function selection 327	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
279	Output function selection 328	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
280	Output function selection 329	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
281	Output function selection 330	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
282	Output function selection 331	0	0 ~ 99		Output function specification value * Refer to 1.2, "I/O Function Lists" under "I/O Parameters" for details.
283 ~ 300	(For expansion)	0			

1.2 I/O Function Lists

(1) Input Function List

Input function specification value	Function name	Remarks
0	General-purpose input	
1	Program start signal (BCD) (ON edge)	Specify a BCD program number using the ports to which start-program number specification bits x (input function specification values 9 through 15) are assigned. * To ensure starting of the program, keep these bits ON for at least 100 msec. * The following input functions cannot be assigned at the same time: • Program start signal (BCD) (input function specification value = 1) • Program start signal (BIN) (input function specification value = 2)
2	Program start signal (BIN) (ON edge)	Specify a binary program number using the ports to which start-program number specification bits x (input function specification values 9 through 15) are assigned. * To ensure starting of the program, keep these bits ON for at least 100 msec. * The following input functions cannot be assigned at the same time: • Program start signal (BCD) (input function specification value = 1) • Program start signal (BIN) (input function specification value = 2)
3	Soft reset signal (ON for 1 second)	If the emergency-stop recovery type is set to "Operation continued," enable the soft reset signal (to ensure the specified operation cancellation method will work.)
4	Servo ON	ON edge: Same as the all-valid-axes servo ON command OFF edge: Same as the all-valid-axes servo OFF command (an interval of at least 1.5 seconds is required). * The signal must be input when the actuator is not operating.
5	Auto-start program start signal	ON edge: Start the program OFF edge: Abort all operations/programs (excluding the I/O processing program at operation/program abort) * Turn ON the signal for at least 100 msec to ensure starting of the program.
6	Soft interlock for all servo axes (OFF level)	Effective when the servo OFF command is not active. Operations will be put on hold if the interlock signal is input during auto operation. Operations will be aborted if the interlock signal is input during non-auto operation.
7	Operation-pause reset signal (ON edge)	
8	Operation pause signal (OFF level)	Effective only during auto operation. * Pause is reset using the operation-pause reset signal.
9	Start-program number specification bit 1 (least significant bit)	* Start-program number specification bits x (input function setting values 9 through 15) cannot be assigned discontinuously from the LSB or in descending order from the LSB (port numbers are not considered). Program No. 1 (BIN or BCD)
10	Start-program number specification bit 2	(Same as "Input function specification value = 9") Program No. 2 (BIN or BCD)
11	Start-program number specification bit 3	(Same as "Input function specification value = 9") Program No. 4 (BIN or BCD)
12	Start-program number specification bit 4	(Same as "Input function specification value = 9") Program No. 8 (BIN or BCD)
13	Start-program number specification bit 5	(Same as "Input function specification value = 9") Program No. 16 (BIN) or 10 (BCD)
14	Start-program number specification bit 6	(Same as "Input function specification value = 9") Program No. 32 (BIN) or 20 (BCD)
15	Start-program number specification bit 7	(Same as "Input function specification value = 9") Program No. 64 (BIN) or 40 (BCD)
16	Error reset (ON edge)	
17	Drive-source cutoff reset input (ON edge) (Effective when the problem factor has been removed)	Drive-source cutoff control is not available for axes whose motor-drive power source is not installed in this controller, or axes whose drive-source cutoff circuit is not controlled by this controller.
18	Home return command signal for all valid axes (ON edge)	The servo must be turned on first (Input function specification value = 4, axis-specific parameter No. 13)
19	Home return command signal for all incremental axes (ON edge)	The servo must be turned on first (Input function specification value = 4, axis-specific parameter No. 13)
20	PC/TP-servo movement command acceptance permission input	* Caution: Ineffective once operation is started.
21	Remote-mode control input	Is the specified DI is ON or the AUTO/MANU switch is set to "MANU," the system mode will become MANU. * Debug filter is disabled on the remote-mode control input port.
22	Axis 1 forced brake-release input	When the applicable port turns ON, the brake will be unlocked forcibly (pay attention to falling load). * Brake release of the synchronized slave axis conforms to brake release of the synchronized master axis.
23	Axis 2 forced brake-release input	When the applicable port turns ON, the brake will be unlocked forcibly (pay attention to falling load). * Brake release of the synchronized slave axis conforms to brake release of the synchronized master axis.
24 ~ 27	For future expansion	

(2) Output Function List

Output function specification value	Function name	Remarks
0	General-purpose output	
1	Operation-cancellation level or higher error output (ON)	* The following output functions cannot be assigned at the same time: • Operation-cancellation level or higher alarm output (ON) (Output function specification value = 1) • Operation-cancellation level or higher alarm output (OFF) (Output function specification value = 2) • Operation-cancellation level or higher alarm + emergency stop output (ON) (Output function specification value = 3) • Operation-cancellation level or higher alarm + emergency stop output (OFF) (Output function specification value = 4)
2	Operation-cancellation level or higher error output (OFF)	(Same as "Output function specification value = 1")
3	Operation-cancellation level or higher error + emergency stop output (ON)	(Same as "Output function specification value = 1")
4	Operation-cancellation level or higher error + emergency stop output (OFF)	(Same as "Output function specification value = 1")
5	READY output (PIO trigger program operation enabled)	* The following output functions cannot be assigned at the same time: • READY output (PIO trigger program operation enabled) (Output function specification value = 5) • READY output (PIO trigger program operation enabled AND absence of operation-cancellation level or higher error) (Output function specification value = 6) • READY output (PIO trigger program operation enabled AND absence of cold-start level or higher error) (Output function specification value = 7)
6	READY output (PIO trigger program operation enabled AND absence of operation-cancellation level or higher error)	(Same as "Output function specification value = 5")
7	READY output (PIO trigger program operation enabled AND absence of cold-start level or higher error)	(Same as "Output function specification value = 5")
8	Emergency stop output (ON)	* The following output functions cannot be assigned at the same time: • Emergency stop output (ON) (Output function specification value = 8) • Emergency stop output (OFF) (Output function specification value = 9)
9	Emergency stop output (OFF)	(Same as "Output function specification value = 8")
10	AUTO mode output	
11	Auto operation status output	(Other parameter No. 12)
12	All-valid-axes home (= 0) output	* To move the absolute-encoder axis to coordinate 0 or the preset home coordinate, use a MOVP command instead of a HOME command.
13	All-valid-axes home return complete (coordinate confirmed) output	* To move the absolute-encoder axis to coordinate 0 or the preset home coordinate, use a MOVP command instead of a HOME command.
14	All-valid-axes preset home coordinate output	* To move the absolute-encoder axis to coordinate 0 or the preset home coordinate, use a MOVP command instead of a HOME command.
15	Voltage-low warning output for system-memory backup battery	
16	Voltage-low warning output for absolute-data backup battery	All axes are checked by the OR gate. Once an abnormal level has been detected, the signal will remain ON until a power ON reset or software reset is performed.
17	Drive-source cutoff (SDN) notification output	The output port will turn OFF when the drive source is cut off. (* Caution: This notification output is implemented only by software means.)
18	For future expansion	
19	For future expansion	
20 ~ 23	For future expansion	
24	Axis 1 servo-ON status output	
25	Axis 2 servo-ON status output	
26 ~ 29	For future expansion	

The following assignments are prohibited:
• Assign a specification value not included in the I/O function lists.
• Assign the same input function specification value, which is not for general-purpose input, to multiple input ports.
• Assign the same output function specification value, which is not for general-purpose output, to multiple output ports.
(For the conditions associated with each specification value, refer to the Remarks field of the applicable item.)

If a prohibited assignment is set, an error "I/O function assignment error" will generate and all input ports and output ports will become general-purpose inputs and general -purpose outputs, respectively.

* In the positioner mode, input and output function assignments are ignored. Each function will follow the corresponding specification in the positioner mode.

2. Parameters Common to All Axes

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
			~		
1	Valid axis pattern	0000B	00B ~ 11111111B		An OFF bit indicates that no driver is installed.
2	Default override	100	1 ~ 100		Used if not specified in program. (Invalid for SIO operation)
3 ~ 8	(For expansion)	0	~		
9	Physical axis pattern for which enable switch (deadman switch/safety gate) is effective	11111111B	00B ~ 11111111B		Not affected by a BASE command. (To make the enable switch effective for all axes (= it must be effective for all axes, as a rule), always specify "11111111." Only when "11111111" is set will the enable switch be included in the drive-source cutoff factor. If a value other than "11111111" is set, the drive source will not be cut off and only the servo of the specified axis will be turned off.) * All axes are specified if "Other parameter No. 11: Deadman switch/safety-gate open recovery type" is set to 1 (Reset required for recovery). * The drive-source cannot be cut off for axes whose motor-drive power unit is not housed inside this controller or whose drive-source cutoff circuit is not controlled by this controller. * If the optional (custom) specification is available, the optional (custom) specification will be given priority over the deadman-switch-enabled physical axis/drive-source cutoff specification, servo OFF specification or 7-segment display specification.
10	(For expansion)	0	0H ~ FFFFFFFFH		
11	Default acceleration	30	1 ~ 200	0.01 G	Used if not specified in position data, program or SIO message, etc.
12	Default deceleration	30	1 ~ 200	0.01 G	Used if not specified in position data, program or SIO message, etc.
13	Default speed	30	1 ~ 250	mm/s	Used if not specified in SIO message or position data, when movement is to be continued, etc.
14	Valid selection when operation point data deceleration is 0	0	0 ~ 5		0: "Deceleration = Acceleration" when the deceleration in the operation point data is "0" 1: "Deceleration = 0" when the deceleration in the operation point data is "0"
15	Maximum jog speed when home return is incomplete	30	1 ~ 250	mm/s	
16 ~ 18	(For expansion)	0	~		
19	Processing type upon stationary (non-push) torque limit over	0	0 ~ 9		0: Operation-cancellation level error (recommended) 1: Operation cancellation (SEL command outputs will turn OFF) * Driver errors resulting from overload, etc., will be given priority.
20	Maximum operating speed check timing	1	0 ~ 1		0: Check at input 1: Check at operation * If "Check at operation" is selected, the distribution speed (CP) of specified speed or the specified speed (PTP) will be compared against the maximum operating speed of each axis and clamped at the allowable speed. Accordingly, the system can achieve its maximum performance in accordance with the operation command. However, complete check cannot be performed at input (since the command/operation start position is indeterminable). In the case of CP, the distribution speed will vary depending on the operation start position. Therefore, specifying CP at an unspecified position (first point movement, etc.) will cause the speed to fluctuate depending on where the operation is started.
21	Maximum operating speed for input value check	1000	1 ~ 9999	mm/s	If "Input" is selected as the maximum speed check timing, this parameter will be used to check for input error.
22	Maximum acceleration	100	1 ~ 999	0.01 G	
23	Maximum deceleration	100	1 ~ 999	0.01 G	
24	Minimum emergency deceleration	30	1 ~ 300	0.01 G	

Parameters Common to All Axes

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
25	(Acceleration/deceleration at home return (old))	30	1 ~ 300	0.01 G	(Invalid)
26	Acceleration/deceleration specification type	0	Reference only		0: T system, 1: P, M system
27	Master axis type	0	Reference only		0: T system, 1: P system
28	Selection of inching → jog auto-switching prohibition	0	Reference only		0: Execute auto-switching (Continuous button ON timer), 1: Prohibited * Referenced by the PC/TP (no handy terminal auto-switching function)
29	All-axis setting bit pattern 1	10000H	0H ~ FFFFFFFFH		Bits 0 to 3: Selection of use of last PC/TP inching distance (0: Do not use, 1: Use) * Referenced by the PC/TP (Excluding ANSI-compatible TP) Bits 4 to 7: Overrun (servo) error level (0: Operation-cancellation level, 1: Cold-start level, 2: Operation-cancellation level at reset, thereafter cold-start level) Bits 8 to 11: "Actual-position soft limit over (servo)" error level (0: Operation-cancellation level, 1: Cold-start level, 2: Operation-cancellation level at reset, thereafter cold-start level) Bits 12 to 15: For future expansion Bits 16 to 19: Absolute-data backup battery voltage error level (0: Operation-cancellation level, 1: Message level)
30	Default division angle	150	0 ~ 1200	0.1 degree	
31	Default division distance	0	0 ~ 10000	mm	
32	Arch-trigger start-point check type	0	0 ~ 5		0: Check operation amount and actual position, 1: Check operation amount only
33	Safety speed in manual mode	250	1 ~ 250	mm/s	* This parameter is treated as a value equivalent to or below the minimum value set in "Axis-specific parameter No. 29, VLMX speed" for all valid axes.
34 ~ 100	(For expansion)	0	~		
			~		
101	For future expansion	0H	0H ~ FFFFFFFFH		
102	For future expansion	0H	0H ~ FFFFFFFFH		
103	For future expansion	0H	0H ~ FFFFFFFFH		
104	For future expansion	0H	0H ~ FFFFFFFFH		
105 ~ 120	(For expansion)	0	~		
			~		
			~		

3. Axis-Specific Parameters

No	Parameter name	Default value (Reference)	Input range	Unit	Remarks
			~		
1	Axis operation type	0	0 ~ 1		0: Linear movement axis, 1: Rotational movement axis (Angle control)
2 ~ 5	(For expansion)	0	~		
6	Coordinate/physical-operation direction selection	1	0 ~ 1		0: Motor CCW → Positive direction on the coordinate system 1: Motor CCW → Negative direction on the coordinate system
7	Soft limit +	50000	-99999999 ~ 99999999	0.001 mm	Fixed to 359.999 degrees internally in the index mode. Invalid in the infinite-stroke mode.
8	Soft limit –	0	-99999999 ~ 99999999	0.001 mm	Fixed to 0 degree internally in the index mode. Invalid in the infinite-stroke mode.
9	Soft-limit actual position margin	2000	0 ~ 9999	0.001 mm	Actual position margin in the positioning boundary critical zone in the infinite-stroke mode
10	Home-return method	0	0 ~ 5		0: Search phase Z after end search, 1: Current position 0 home (This parameter can be specified only with an incremental encoder. Pay attention to contact.), 2: Current position = Preset home (This parameter can be specified only with an incremental encoder. Pay attention to contact.)
11	Home-return end-search direction selection	0	0 ~ 1		0: Negative end of the coordinate system 1: Positive end of the coordinate system
12	Home preset value	0	-99999999 ~ 99999999	0.001 mm	(Refer to axis-specific parameter No. 76)
13	SIO/PIO home-return order	0	0 ~ 16		Executed from the smallest one.
14	Home-sensor input polarity	0	0 ~ 2		0: Do not use, 1: Contact a, 2: Contact b
15	For future expansion	0	Reference only		
16	For future expansion	0	Reference only		
17	Initial home-sensor pull-out speed at home return	10	1 ~ 100	mm/sec	
18	For future expansion	100	Reference only		
19	End search speed at home return	20	1 ~ 100	mm/sec	
20	Phase-Z search speed at home return	3	1 ~ 10	mm/sec	Exercise caution, since limitations apply depending on the read/encoder pulse count.
21	Offset travel distance at home return	1000	-99999999 ~ 99999999	0.001 mm	Offset travel distance from the ideal phase-Z position (Positive value = Applied in the direction of moving away from the end) (Refer to axis-specific parameter No. 76) * Note on absolute encoders When a value near an integer multiple of the phase-Z distance (including an offset travel distance of 0) is set in this parameter, the servo will lock above phase Z upon absolute reset. As a result, the coordinates may shift by the pulses corresponding to the phase-Z distance. Therefore, never set a value near an integer multiple of the phase-Z distance. (Provide a sufficient margin with respect to the servo amplitude.)
22	Allowable phase-Z position error check value at home return	200	0 ~ 99999999	0.001 mm	Minimum allowable distance between the end (mechanical or LS) and phase Z in a rotary encoder specification. Phase-Z search limit in a linear encoder specification.
23	Phase-Z count per encoder revolution	1	1 ~ 8		Only “1” can be set, in the case of an absolute encoder. Invalid in the case of a linear encoder.
24	Push stop check time at home return	700	1 ~ 5000	msec	Used to confirm push action during home return.
25	Push stop check time at positioning	500	1 ~ 5000	msec	Used to confirm push action during PUSH command operation.

Axis-Specific Parameters

No	Parameter name	Default value (Reference)	Input range	Unit	Remarks
26	(Phase-Z evacuation distance at absolute home return (old))	1000	0 ~ 99999	0.001 mm	Evacuation distance from the actual phase-Z position (Positive value = Applied in the direction of moving away from the end) (Phase-shift prevention margin) (Refer to axis-specific parameter No. 76)
27	Maximum motor speed	5000	Reference only		In rpm when a rotary encoder is used, or in mm/sec when a linear encoder is used (cannot be changed).
28	Maximum operating speed of each axis	1000	1 ~ 9999	mm/s	
29	VLMX speed	1000	1 ~ 9999	mm/s	During VLMX operation, the maximum operating speed of each axis or VLMX speed, whichever is lower, is used as the maximum speed of the applicable axis.
30	Servo ON check time	150	0 ~ 5000	msec	Brake equipped: Time after receiving a servo-ON start response until start of brake unlocking Brake not equipped: Time after receiving a servo ON start response until transition to an operation-enabled status
31	Offset travel speed at home return	3	1 ~ 500	mm/sec	
32	Actual distance between phase Z and end	-1	-1 ~ 99999	0.001 mm	Absolute distance from the end (mechanical or LS). Obtained automatically if the distance is a negative value. When multiple actuators are combined, it is recommended to write the flash ROM after automatic acquisition. (Refer to axis-specific parameter No. 76)
33	Ideal distance between phase Z and end	0	0 ~ 99999	0.001 mm	Absolute distance from the end (mechanical or LS). (Refer to axis-specific parameter No. 76)
34	Brake equipment specification	0	0 ~ 1		0: Not equipped, 1: Equipped
35	Brake unlock check time	150	0 ~ 3000	msec	Time after receiving a brake-unlock start response until transition to an operation-enabled status
36	Brake lock check time	300	0 ~ 1000	msec	Time after receiving a brake-lock start response until start of servo OFF
37	Encoder linear/rotary type	0	0 ~ 1		0: Rotary encoder 1: Linear encoder
38	Encoder ABS/INC type	0	0 ~ 1		0: INC, 1: ABS
39	Magnetic-pole sensor equipment specification	0	0 ~ 1		0: Not equipped, 1: Equipped
40	For future expansion (change prohibited)	0	0 ~ 1		
41	For future expansion (change prohibited)	25	1 ~ 100	DRVVR	
42	Encoder resolution	800	0 ~ 99999999	Pulse/rev, 0.001 μm/pulse	Pulses (before division)/rev, in the case of a rotary encoder. 0.001 μm/pulse (before division), in the case of a linear encoder.
43	Encoder division ratio	0	-7 ~ 7		Pulses are multiplied by ("n"th power of 1/2).
44	Length measurement correction	0	-99999999 ~ 99999999	0.001 mm/ 1M	Valid only for linear movement axes. (Coordinates other than the encoder reference Z point will change proportionally.)
45 ~ 46	(For expansion)	0			
47	Screw lead	6000	1 ~ 99999999	0.001 mm	Valid only for linear movement axes. Invalid in the case of a linear encoder.
48 ~ 49	(For expansion)	0			
50	Gear ratio numerator	1	1 ~ 99999999		Invalid in the case of a linear encoder.
51	Gear ratio denominator	1	1 ~ 99999999		Invalid in the case of a linear encoder.
52	(For expansion)	0			
53	Setting bit pattern 1 of each axis	0	0H ~ FFFFFFFFH		
54	Travel distance for push stop detection at home return	20	1 ~ 99999	0.001 mm	Used to confirm push action during home return.
55	Travel distance for push stop detection at positioning	30	1 ~ 99999	0.001 mm	Used to confirm push action during PUSH command operation.
56	Push-abort deviation ratio at home return	2000	1 ~ 99999		Deviation is compared against "Steady-state deviation of push speed + Push-speed pulse speed x Abort deviation ratio."

Axis-Specific Parameters

No	Parameter name	Default value (Reference)	Input range	Unit	Remarks
57	Push-abort deviation ratio at positioning	5000	1 ~ 99999		Deviation is compared against “Steady-state deviation of push speed + Push-speed pulse speed x Abort deviation ratio.”
58	Positioning band	100	1 ~ 9999	0.001 mm	
59	Allowable deviation error ratio (Maximum speed pulse ratio)	27	1 ~ 9999		Deviation is compared against “Steady-state deviation of maximum operating speed of each axis + Pulse speed of maximum operating speed of each axis x Allowable deviation error ratio.”
60	Position gain	30	1 ~ 9999	/s	
61	FAG	0	0 ~ 999		
62	Synchro FB gain	77	0 ~ 1000		
63	Stop special output range	1	0 ~ 9999	Pulse	Invalid if “0” is set.
64	Stop special output value	1	0 ~ 999	DRVVR	
65	Mating synchro-axis number	0	0 ~ 8		Must be input for both axes. (Of the axis pair, the axis with the smaller axis number becomes the master axis. Both axes must have the same resolution characteristics. Commands cannot be issued to the slave axis.) (Invalid if “0” is set)
66	Mode selection for rotational movement axis	0	0 ~ 5		0: Normal, 1: Index mode
67	Short-cut control selection for rotational movement axis	0	0 ~ 5		0: Do not select, 1: Select (Valid only in the index mode AND when an incremental encoder is used)
68	Mode selection for linear movement axis	0	0 ~ 5		0: Normal, 1: Infinite-stroke mode (Note: Positioning boundary applies. This setting can be specified only when an incremental encoder is used.)
69	(For expansion)	0	~		
70	For future expansion	0	Reference only		
71	For future expansion	0	Reference only		
72	DRVVR + offset	0	Reference only	DRVVR	(Change prohibited) To maintain symmetry of the positive and negative sides.
73	DRVVR – offset	0	Reference only	DRVVR	(Change prohibited) To maintain symmetry of the positive and negative sides.
74	For future expansion	0	Reference only		
75	For future expansion	0	Reference only		
76	Home-adjustment parameter set selection	1	Reference only		(Change prohibited) 0: P21 = Phase-Z evacuation distance at INC home return P12 = Ideal phase-Z position coordinate 1: P32 is read automatically even when P33 = 0. P33 = 0 indicates “actual distance.” P21 = Offset travel at home return P12 = Coordinate achieved by offset travel at home return P26 = Invalid (To facilitate adjustment)
77	Synchro S pulse	3	0 ~ 99999	Pulse	
78	Maximum takeoff command amount	0	-3000 ~ 3000	0.001 mm	Maximum lift command amount before brake unlock (Input with sign) (Suppression of momentary drop upon servo ON when a heavy object is placed) * Important: Input using the same sign as the rising coordinate direction. (0.100 mm to 0.500 mm in absolute value as a guideline) * The servo-ON check time (axis-specific parameter No. 30) must also be extended (approx. 1000 to 1500 msec) to provide a sufficient time for rise-direction torque to follow. (Valid only when installation of brake is specified.)
79	Actual takeoff check distance	5	0 ~ 3000	0.001 mm	Absolute value input
80	Maximum forced-feed range	0	0 ~ 9999	0.001 mm	For reduction of settling time. (Invalid range if “0” is set) (Approx. 1.000 mm as a guideline)
81	Minimum forced-feed range	200	0 ~ 9999	0.001 mm	
82	Medium forced-feed range	600	0 ~ 9999	0.001 mm	
83	Absolute synchro slave-axis initialization cancellation	0	0 ~ 5		Valid only with a synchro slave axis.

Axis-Specific Parameters

No	Parameter name	Default value (Reference)	Input range	Unit	Remarks
84	Maximum synchronization correction speed of synchro slave axis	5	0 ~ 100	mm/sec	Maximum travel speed for synchronization position correction of slave axis. Valid only with a synchro slave axis. * Note: Not limited by the safety speed.
85	Home-return acceleration/ deceleration	15	1 ~ 300	0.01 G	
86	Zone 1 MAX	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
87	Zone 1 MIN	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
88	Zone 1 output number	0	0 ~ 899		Physical output port or global flag (Output is invalid if “0” is input; multiple specification is invalid)
89	Zone 2 MAX	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
90	Zone 2 MIN	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
91	Zone 2 output number	0	0 ~ 899		Physical output port or global flag (Output is invalid if “0” is input; multiple specification is invalid)
92	Zone 3 MAX	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
93	Zone 3 MIN	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
94	Zone 3 output number	0	0 ~ 899		Physical output port or global flag (Output is invalid if “0” is input; multiple specification is invalid)
95	Zone 4 MAX	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
96	Zone 4 MIN	0	-99999999 ~ 99999999	0.001 mm	Valid only when MAX > MIN. * Must be inside the range for at least 3 msec.
97	Zone 4 output number	0	0 ~ 899		Physical output port or global flag (Output is invalid if “0” is input; multiple specification is invalid)
98	For future expansion	0	Reference only		
99	For future expansion	0	Reference only		
100 ~ 118	(For expansion)	0	~		
119	FSG	0	0 ~ 100		
120	FFF	10	0 ~ 100		* Change is prohibited unless instructed by the manufacturer.
121~ 170	(For expansion)	0	~		
171		0	~		
172		0	~		
173		0	~		
174		0	~		
175		0	~		
176		0	~		
~ 200	(For expansion)	0	~		

4. Driver Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
1	Type (upper) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
2	Type (middle) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
3	Type (lower) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
4	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
5	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
6	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
7	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
8	Board type (Function information)	0	Reference only		For adjustment by the manufacturer
9	Installation type word 1 (Function information)	0101H	Reference only		For adjustment by the manufacturer
10	Installation type word 2 (Function information)	0000H	Reference only		For adjustment by the manufacturer
11	(Function information)	0000H	Reference only		
12	Software version (Function information)	0000H	Reference only		For adjustment by the manufacturer
13	Maximum supported motor ID number (Function information)	0000H	Reference only		For adjustment by the manufacturer
14	Motor control data use selection (Function information)	0000H	Reference only		For adjustment by the manufacturer
15	(Function information)	0000H	Reference only		For adjustment by the manufacturer
16	(Function information)	0000H	Reference only		For adjustment by the manufacturer
17	(Function information)	0000H	Reference only		For adjustment by the manufacturer
18	(Function information)	0000H	Reference only		For adjustment by the manufacturer
19	(Function information)	0000H	Reference only		For adjustment by the manufacturer
20	(Function information)	0000H	Reference only		For adjustment by the manufacturer
21	(Function information)	0000H	Reference only		For adjustment by the manufacturer
22	(Function information)	0000H	Reference only		For adjustment by the manufacturer
23	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer
24	Configuration capacity (rated motor output) (compatible with E, priority on E) (configuration information)	0014H	Reference only	W	For adjustment by the manufacturer
25	Configuration voltage (motor voltage) (compatible with E, priority on E) (configuration information)	0018H	Reference only	V	For adjustment by the manufacturer
26	Motor/encoder configuration information (compatible with E, priority on E) (configuration information)	0005H	Reference only		For adjustment by the manufacturer
27	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer
28	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer

Driver parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
29	Motor/encoder characteristic word (compatible with E, priority on E) (configuration information)	0000H	Reference only		For adjustment by the manufacturer
30	Motor/encoder control word 1 (compatible with E, priority on E) (configuration information)	5000	Reference only	0.1 K (Kelvin = temperature unit)	For adjustment by the manufacturer
31	Motor/encoder control word 2 (compatible with E, priority on E) (configuration information)	0000H	Reference only		For adjustment by the manufacturer
32	Motor/encoder control word 3 (configuration information) (encoder cable length) [m]	2	1 ~ 30		Encoder cable length (m) If the encoder has been replaced, don't forget to change the setting of this parameter.
33	Motor/encoder control word 4 (configuration information)	14H	Reference only		For adjustment by the manufacturer
34	Motor/encoder control word 5 (configuration information)	0000H	Reference only		For adjustment by the manufacturer
35	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer
36	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer
37	(Configuration information)	0000H	Reference only		For adjustment by the manufacturer
38	Push torque limit at positioning	70	0 ~ 70	%	
39	Push torque limit at home return	120	0 ~ 150	%	The factory setting conforms to the standard specification of the actuator. The home return torque becomes higher as the value gets larger. This does not usually need a change, however, it may be necessary to make the value larger when the home return operation completes before reaching the proper position due to increase of slider resistance depending on the way to mount or the condition of load in vertical oriented mounting. Setting Reference: Set value = 120% (when direction of home return is upwards) to 80% (when direction of home return is downwards)
40	Maximum torque limit	300	10 ~ 400	%	* The maximum value that can be set varies depending on the motor, etc.
41	Dynamic brake operation specification	0	0 ~ 1		(Data for other model)
42	Software DB operation specification	0	0 ~ 1		0: Disable, 1: Enable
43	Speed loop proportional gain	500	1 ~ 32767		Proportional gain
44	Speed loop integral gain	1667	1 ~ 3276700		Integral gain
45	Torque filter time constant	0	0 ~ 2500		
46	Current control band number	4	0 ~ 4		
47	Current ON time for excited-phase signal detection step	128	0 ~ 32767	ms	
48	Excited-phase signal detection method	1	0 ~ 2		0: Current suppression method 1: Distance suppression method 2: Distance suppression method (300% excitation) (Main application version 0.10 or later)
49	Excited-phase signal detection direction	0	0 ~ 1		0: CW, 1: CCW
50	Excited-phase fixed mode: Torque-limit switching type	0	0 ~ 1		(Data for other model)
51	Excited-phase fixed mode: Torque limit	0	0 ~ 100	%	(Data for other model)
52	(For expansion)	0H	0000H ~ FFFFH		
53	Current control word 1	0H	Reference only		
54	Current control word 2	0H	Reference only		

Driver parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
55	Current control word 3	0H	Reference only		
56	Current control word 4	0H	Reference only		
57	Current control word 5	0H	Reference only		
58	Current control word 6	0H	Reference only		
59	Current control word 7	0H	0000H ~ FFFFH		
60	Current control word 8	0H	00000000H ~ FFFFFFFFH		
61 ~ 67	(For expansion)	0H	00000000H ~ FFFFFFFFH		
68 ~ 97	For future expansion	0H	Reference only		

5. Encoder Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
1	Type (upper) (Manufacturing information)	Space	Reference only		
2	Type (middle) (Manufacturing information)	Space	Reference only		
3	Type (lower) (Manufacturing information)	Space	Reference only		
4	Manufacturing data (Manufacturing information)	Space	Reference only		
5	Manufacturing data (Manufacturing information)	Space	Reference only		
6	Manufacturing data (Manufacturing information)	Space	Reference only		
7	Manufacturing data (Manufacturing information)	Space	Reference only		
8	Board type (Function information)	0	Reference only		
9	Configuration capacity (rated motor output) (compatible with X/E) (function information)	0000H	Reference only	W	For adjustment by the manufacturer
10	Configuration voltage (motor voltage) (compatible with X/E) (function information)	0000H	Reference only	V	For adjustment by the manufacturer
11	Motor/encoder configuration information (compatible with X/E) (function information)	0000H	Reference only		For adjustment by the manufacturer
12	Encoder resolution (upper word) (compatible with X/E) (function information)	0000H	Reference only		For adjustment by the manufacturer
13	Encoder resolution (lower word) (compatible with X/E) (function information)	0000H	Reference only		For adjustment by the manufacturer
14	Motor/encoder characteristic word (compatible with X/E) (function information)	0000H	Reference only		For adjustment by the manufacturer
15	Motor/encoder control word 1 (function information)	0000H	Reference only	0.1 K (Kelvin = temperature unit)	For adjustment by the manufacturer
16	Motor/encoder control word 2 (function information)	0000H	Reference only		For adjustment by the manufacturer
17	Motor/encoder control word 3 (function information)	0000H	Reference only		For adjustment by the manufacturer
18	Motor/encoder control word 4 (function information)	0001H	Reference only		For adjustment by the manufacturer
19	(Function information)	0000H	Reference only		For adjustment by the manufacturer
20	(Function information)	0000H	Reference only		For adjustment by the manufacturer
21	(Function information)	0000H	Reference only		For adjustment by the manufacturer
22	(Function information)	0000H	Reference only		For adjustment by the manufacturer
23 ~ 30	Card parameter (by board type)	0000H	Reference only		For adjustment by the manufacturer

6. I/O Devices

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
1	Type (upper) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
2	Type (middle) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
3	Type (lower) (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
4	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
5	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
6	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
7	Manufacturing data (Manufacturing information)	Space	Reference only		For adjustment by the manufacturer
8	Board type (Function information)	0	Reference only		For adjustment by the manufacturer
9	Function information 01 (by board type)	0000H	Reference only		For adjustment by the manufacturer
10	Function information 02 (by board type)	0000H	Reference only		For adjustment by the manufacturer
11	Function information 03 (by board type)	0000H	Reference only		For adjustment by the manufacturer
12	Function information 04 (by board type)	0000H	Reference only		For adjustment by the manufacturer
13	Function information 05 (by board type)	0000H	Reference only		For adjustment by the manufacturer
14	Function information 06 (by board type)	0000H	Reference only		For adjustment by the manufacturer
15	Function information 07 (by board type)	0000H	Reference only		For adjustment by the manufacturer
16	Function information 08 (by board type)	0000H	Reference only		For adjustment by the manufacturer
17	Function information 09 (by board type)	0000H	Reference only		For adjustment by the manufacturer
18	Function information 10 (by board type)	0000H	Reference only		For adjustment by the manufacturer
19	Function information 11 (by board type)	0000H	Reference only		For adjustment by the manufacturer
20	Function information 12 (by board type)	0000H	Reference only		For adjustment by the manufacturer
21	Function information 13 (by board type)	0000H	Reference only		For adjustment by the manufacturer
22	Function information 14 (by board type)	0000H	Reference only		For adjustment by the manufacturer
23 ~ 52	Device parameter (by board type)	0000H	Reference only		For adjustment by the manufacturer
53 ~ 82	Query information 01 to 30 (by board type)	0000H	Reference only		For adjustment by the manufacturer

7. Other Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
1	Auto-start program number	0	0 ~ 64		(Invalid if "0" is set)
2	I/O processing program number at operation/program abort	0	0 ~ 64		The start trigger is determined from the "I/O processing program start type at operation/program abort." (Note: This program will be started before confirming an abort of other programs.) (Invalid if "0" is set) * If the setting is valid, the number of user program tasks that can be used will decrease by 1.
3	I/O processing program number at all operation pause	0	0 ~ 64		This program will be started when an all-operation-pause command is issued due to an all-operation-pause factor. (Only when a program is running) (Invalid if "0" is set) * If the setting is valid, the number of user program tasks that can be used will decrease by 1.
4	Program abort type at error	0	0 ~ 5		0: Cancel only the program in which an error of operation-cancellation level or higher has generated. (If the error requires the drive source to be cut off or a servo-OFF or all-axis servo-OFF request to be issued, all programs other than the "I/O processing program at operation/program abort" will be cancelled.) 1: Cancel all programs other than the "I/O processing program at operation/program abort" when an error of operation-cancellation level or higher has generated.
5	I/O processing program start type at operation/program abort	0	0 ~ 5		0: When all-operation-cancellation factor has generated (Only when a program is running) 1: When all-operation-cancellation factor has generated (Always) 2: All-operation-cancellation factor + Error of operation-cancellation level or higher ("Other parameter No. 4 = 0" is considered) (Only when a program is running) 3: All-operation-cancellation factor + Error of operation-cancellation level or higher ("Other parameter No. 4 = 0" is considered) (Always)
6	PC/TP reconnection delay at software reset	14000	1 ~ 99999	msec	* The setting will become effective after the controller, PC or TP is restarted.
7	Auto program start setting	1	0 ~ 5		0: Do not start the auto-start program upon power ON reset/software reset 1: Start the auto-start program
8	(For expansion)	0			
9	For future expansion (change prohibited)	0	0 ~ 2		
10	Emergency-stop recovery type	0	0 ~ 4		0: Abort operations/programs 1: Recovery after reset 2: Operation continued (Only during automatic operation. * Operation commands from the PC/TP will be aborted on the PC/TP side.) 3: Abort operations/programs (Software reset when the emergency stop is reset. The home-return completion status of incremental-encoder axes will be reset (EG approximation swap).) 4: Abort operations/programs (Error reset (only with an error of operation-cancellation level or lower) and auto-start program start (only if AUTO mode AND other parameter No. 7 = 1 AND I/O parameter "Input function selection" ≠ 17 AND all-operation-cancellation factor is not present) when the emergency stop is reset. There must be a minimum interval of 1 second after an emergency stop is actuated before it is reset. The home-return completion status of incremental-encoder axes will be retained.
11	Enable switch (deadman/enable switch) recovery type	0	0 ~ 2		0: Abort operations/programs 1: Recovery after reset 2: Operation continued (Only during automatic operation. * Operation commands from the PC/TP will be aborted on the PC/TP side.)

Other Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
12	Automatic operation recognition type	0	0 ~ 3		0: Program is running AND all-operation-cancellation factor is not present 1: [Program is running OR in AUTO mode] AND all-operation-cancellation factor is not present
13 ~ 19	(For expansion)	0			
20	System-memory backup battery installation function type	0	0 ~ 2		0: Not installed (SEL global data/error lists cannot be recovered from the flash ROM) 1: Not installed (SEL global data/error lists can be recovered from the flash ROM) 2: Installed * When the power is turned on without battery installed, point data can be copied from the flash ROM. * Use of setting “1” will be prohibited for the time being due to limitations. * When point data is lost due to a battery error, the point data valid before the flash ROM was written can be restored → Input “0” (not installed) and transfer the setting to the controller, and then perform a software reset without writing the flash ROM. The point data last written to the flash ROM will be restored. Thereafter, reset this parameter to the original value. (No remedy is available for recovery of SEL global data/error lists.)
21	Manual mode type	0	0 ~ 5		0: Always enable edit and SIO/PIO start (Initial condition after connection = With safety speed) 1: Select edit and start (with password) (EU, etc.) 2: Always enable edit and SIO/PIO start (Initial condition after connection = Without safety speed (cancellation)) * Referenced by the PC/TP.
22	Control use region	0	0 ~ 99		0: J, 1: E, 2: EU
23	PSIZ command function type	0	0 ~ 5		0: Maximum number of point data areas 1: Number of point data used
24	Local variable number for storing SEL communication command return code	99	1 ~ 99 1001 ~ 1099		
25	Operation mode type	0	0 ~ 16		0: Program mode 1 to 16: Positioner mode
26 ~ 29	(For expansion)	0			
30	Option Password 00	0H	0H ~ FFFFFFFFH		HOME command option (Change prohibited) * Change is prohibited unless instructed by the manufacturer.
31	Option Password 01	0H	0H ~ FFFFFFFFH		Reserved (Change prohibited) * Change is prohibited unless instructed by the manufacturer.
32	Option Password 02	0H	0H ~ FFFFFFFFH		Reserved (Change prohibited) * Change is prohibited unless instructed by the manufacturer.
33 ~ 35	(For expansion)	0	0H ~ FFFFFFFFH		

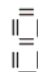
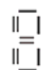
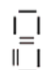
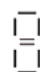
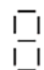
Other Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
36	PC/TP data protect setting (Program)	0H	0H ~ FFFFFFFFH		Bits 0 to 3: Protect type (0: Read/write, 1: Read only, 2: No read/write) Bits 4 to 7: Protect release method (0: Special operation) Bits 8 to 11: Protect range maximum number (1's place, BCD) Bits 12 to 15: Protect range maximum number (10's place, BCD) Bits 16 to 19: Protect range minimum number (1's place, BCD) Bits 20 to 23: Protect range minimum number (10's place, BCD) * Referenced by the PC/TP
37	PC/TP data protect setting (Position)	0H	0H ~ FFFFFFFFH		Bits 0 to 3: Protect type (0: Read/write, 1: Read only, 2: No read/write) Bits 4 to 7: Protect release method (0: Special operation) Bits 8 to 11: Protect range maximum number (10's place, BCD) Bits 12 to 15: Protect range maximum number (100's place, BCD) Bits 16 to 19: Protect range maximum number (1000's place, BCD) Bits 20 to 23: Protect range minimum number (10's place, BCD) Bits 24 to 27: Protect range minimum number (100's place, BCD) Bits 28 to 31: Protect range minimum number (1000's place, BCD) * The value in the 1's place is considered "0" for both the protect range maximum/minimum numbers. * Referenced by the PC/TP
38	PC/TP data protect setting (Symbol, parameter)	0H	0H ~ FFFFFFFFH		Bits 0 to 3: Protect type (Parameter) (0: Read/write, 1: Read only, 2: No read/write) Bits 4 to 7: Protect release method (Parameter) (0: Special operation) Bits 8 to 11: Protect type (Symbol) (0: Read/write, 1: Read only, 2: No read/write) Bits 12 to 15: Protect release method (Symbol) (0: Special operation) * Referenced by the PC/TP
39	(For future expansion)	0H	0H ~ FFFFFFFFH		

Other Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
40	EEPROM information check type	02H	0H ~ FFFFFFFFH		0: Disable checksum, 1: Enable checksum Bit 0 = (For future expansion) Bit 1 = Encoder Bits 2 to 7 = (For future expansion) 0: Do not use EEPROM, 1: Use EEPROM Bits 16 to 23 = (For future expansion)
41	Hardware information check type	0H	0H ~ FFFFFFFFH		Bits 0 = (For future expansion)
42	Hardware test type	0H	0H ~ FFFFFFFFH		Bits 0 to 2 = (For future expansion)
43	For future expansion	0H	0H ~ FFFFFFFFH		
44	(For expansion)	0			
45	Special start condition setting	0	0H ~ FFFFFFFFH		Bits 0 to 3: Enable start from PC/TP in AUTO mode = Used exclusively by the manufacturer (0: Do not enable, 1: Enable) Bits 4 to 7: PIO program start (Input port 000) Single start selection (0: Normal, 1: Single start) * In accordance with the input port for which the I/O parameter "Input function selection" has been set to "1" or "2." * When single start is selected, the next PIO program start will not be accepted as long as a program with the same program number as the one started by the last PIO program start is running. Bits 8 to 11: Permission of auto program start when all-operation-cancellation factor is present (0: Do not permit, 1: Permit) Bits 12 to 15: Permission of ON edge acceptance for PIO-program start when all-operation-cancellation factor is present (0: Do not permit, 1: Permit) * In accordance with the input port for which the I/O parameter "Input function selection" has been set to "1" or "2." * This parameter specifies an ON-edge acceptance condition. If the starting condition is not satisfied, an "Error No. A1E: Start condition non-satisfaction error" will generate.
46	Other setting bit pattern 1	2011H	0H ~ FFFFFFFFH		Bits 0 to 3: Variable-value format type in response message to real-number/variable query (0: Big endian with four upper/lower binary-converted bytes reversed, 1: Big endian) Bits 4 to 7: Decimal-place rounding selection for real-number → integer-variable assignment in LET/TRAN commands (0: Do not round, 1: Round) Bits 8 to 11: For future expansion * Change strictly prohibited unless specified by the manufacturer. Bits 12 to 15: Selection of processing to be performed when subroutine first step input condition is not specified when TPCD command = 1 (0: Do not execute, 1: Execute, 2: Error)
47 ~ 48	(For expansion)	0			

Other Parameters

No.	Parameter name	Default value (Reference)	Input range	Unit	Remarks
49	Panel 7-segment display data type	0	0 ~ 9		0: Display controller status 1: Display motor current indicator The current pattern of each axis is displayed instead of “ready status” or “program run number.” “Minimum indicator-displayed axis number” (far-right column) is specified by “Other parameter No. 50.”  0 < Motor current to rating ratio (%) ≤ 25  25 < Motor current to rating ratio (%) ≤ 50  50 < Motor current to rating ratio (%) ≤ 75  75 < Motor current to rating ratio (%) ≤ 100  100 < Motor current to rating ratio (%) ≤ 150  150 < Motor current to rating ratio (%) ≤ 200  200 < Motor current to rating ratio (%) 2: Display user information number (U001 to U999) The user information number is displayed instead of “ready status” or “program run number” only when the user information number is not “0.” “Global integer variable number for specifying user information number” is specified by “Other parameter No. 50.”
50	Auxiliary specification for panel 7-segment display data type	0	-99999999 ~ 99999999		* Refer to the Remarks field for “Other parameter No. 49.”
51	Monitoring-data buffering period	10	1 ~ 100	msec	
52 ~ 70	(For expansion)	0			
71	Positioner mode parameter 1	0	-99999999 ~ 99999999		
72	Positioner mode parameter 2	0	-99999999 ~ 99999999		
73	Positioner mode parameter 3	0	-99999999 ~ 99999999		
74	Positioner mode parameter 4	0	-99999999 ~ 99999999		
75	Positioner mode parameter 5	0	-99999999 ~ 99999999		
70 ~ 100	(For expansion)	0			

8. Manual Operation Types

The selectable operation types will vary depending on the setting of the “Manual operation type” parameter (Other parameter No. 21).

(1) PC software

[1] Setting = 0 (Always enable edit and SIO/PIO start)

Operation type	Password	Functions				
		Edit	Safety speed	Jog, move, continuous move	SIO program start	PIO program start
With safety speed	Not required.	○	○	○	○	○
Without safety speed	Not required.	○		○	○	○

[2] Setting = 1 (Select edit and start (with password))

Operation type	Password	Functions				
		Edit	Safety speed	Jog, move, continuous move	SIO program start	PIO program start
Edit and jog	Not required.	○	○	○		
SIO start and jog (safety speed)	1817 (*1)		○	○	○	
SIO start and jog	1818 (*1)			○	○	
SIO/PIO start and jog	1819 (*1)			○	○	○

(*1) PC software version 0.0.6.0 or later (“0000” in versions 0.0.0.0 through 0.0.5.x)

(2) Teaching pendant

[1] Setting = 0 (Always enable edit and SIO/PIO start)

Safety-speed enable selection	Password	Functions				
		Edit	Safety speed	Jog, move, continuous move	SIO program start	PIO program start
Enable	Not required.	○	○	○	○	○
Disable	Not required.	○		○	○	○

[2] Setting = 1 (Select edit and start (with password))

Safety-speed enable selection	Password	Functions				
		Edit	Safety speed	Jog, move, continuous move	SIO program start	PIO program start *2
Enable	Not required.	○	○	○	○	(*3)
Disable	1818 (*1)	○		○	○	(*3)

PIO start prohibition selection	Password	Functions				
		Edit	Safety speed	Jog, move, continuous move	SIO program start	PIO program start *2
Prohibit	Not required.	○	(*4)	○	○	
Enable	1819 (*1)	○	(*4)	○	○	○

(*1) Teaching pendant application version 0.02 or later (not supported by version 0.01 or earlier)
(*2) PIO program start is enabled only in modes other than the edit mode.
(*3) In accordance with the “PIO start prohibition selection” setting.
(*4) In accordance with the “Safety-speed enable” setting.

◎ Combination Table of ASEL Linear/Rotary Control Parameters

Axis-specific parameter No. 1, Axis operation type	Axis-specific parameter No. 68, Mode selection for linear movement axis	Axis-specific parameter No. 66, Mode selection for rotational movement axis	Axis-specific parameter No. 67, Short-cut control selection for rotational movement axis	Permitted encoder processing method		Expression of current position (approx.)	Axis-specific parameter No. 7, Soft limit +	Axis-specific parameter No. 8, Soft limit -	Axis-specific parameter No. 44, Length measurement correction	Axis-specific parameter No. 47, Screw lead	Axis-specific parameter No. 50, Gear ratio numerator	Axis-specific parameter No. 51, Gear ratio denominator	Input unit
				ABS	INC								
0 (Linear movement axis)	0 (Normal mode)	Invalid	Invalid	○	○	Counter range	Valid	Valid	Valid	Valid	Valid	Valid	• Distance mm • Speed mm/sec • Acceleration/ deceleration G

◎ Error Level Control

Error level	System error assignment source	Error No. (HEX)	Display (7-segment display, etc.)	Error list (Application only)	Error LED output (MAIN only)	Program run (Application only): Other parameter No. 4 = 0	Program run (Application only): Other parameter No. 4 = 1	Error reset (Application only)	Remarks
Secret level	MAIN application	800 ~ 88F		○					Special error level provided for maintenance purposes
	MAIN core	890 ~ 8AF							
	PC	8B0 ~ 8DF							
	TP	8E0 ~ 8FF							
Message level	MAIN application		○	△ (Battery and fieldbus errors will be registered in an error list.)				Enabled.	Status display, input error, etc.
	MAIN core	-							
	PC								
	PC (Update tool)								
	TP								
	MAIN application	200 ~ 24F							
	MAIN core	-							
	PC	250 ~ 29F							
	PC (Update tool)	2A0 ~ 2CF							
	TP	2D0 ~ 2FF							
	MAIN application	900 ~ 93F							
	MAIN core	940 ~ 97F							
	PC	980 ~ 9AF							
	PC (Update tool)	9B0 ~ 9BF							
	TP	9C0 ~ 9FF							
	MAIN application	A00 ~ A6F							
	MAIN core	A70 ~ A9F							
	PC	AA0 ~ ACF							
	TP	AD0 ~ AFF							
Operation-cancellation level	MAIN application		○	○		The program in which the error generated will be cancelled. (Except for axis errors, a cancellation factor is present only for the moment the error occurs.) * However, in the case of an error requiring servo OFF or all-axis servo OFF, all programs other than the “I/O processing program at operation/program abort” will be cancelled.	All programs other than the “I/O processing program at operation/program abort” will be cancelled. (Except for axis errors, a cancellation factor is present only for the moment the error occurs.)	Enabled.	Errors affecting operation. The system will attempt to reset minor errors below this level using an auto-reset function via external active command (SIO/PIO) (application only).
	MAIN core	-							
	PC								
	PC (Update tool)								
	TP								
	MAIN application	400 ~4CF							
	MAIN core	-							
	PC	4D0 ~ 4DF							
	PC (Update tool)	4E0 ~ 4EF							
	TP	4F0 ~ 4FF							

Error level	System error assignment source	Error No. (HEX)	Display (7-segment display, etc.)	Error list (Application only)	Error LED output (MAIN only)	Program run (Application only): Other parameter No. 4 = 0	Program run (Application only): Other parameter No. 4 = 1	Error reset (Application only)	Remarks
Operation-cancellation level	MAIN application	B00 ~ B9F	○	○		The program in which the error generated will be cancelled. (Except for axis errors, a cancellation factor is present only for the moment the error occurs.) * However, in the case of an error requiring servo OFF or all-axis servo OFF, all programs other than the "I/O processing program at operation/program abort" will be cancelled.	All programs other than the "I/O processing program at operation/program abort" will be cancelled. (Except for axis errors, a cancellation factor is present only for the moment the error occurs.)	Enabled.	Errors affecting operation. The system will attempt to reset minor errors below this level using an auto-reset function via external active command (SIO/PIO) (application only).
	MAIN core	BA0 ~ BBF							
	PC	BC0 ~ BDF							
	TP	BE0 ~ BFF							
	MAIN application	C00 ~ CCF							
	MAIN core	CD0 ~ CDF							
	PC	CE0 ~ CEF							
	TP	CF0 ~ CFF							
Cold-start level	MAIN application		○	○	○ (Core only)	The program in which the error generated will be cancelled. * However, in the case of an error requiring drive-source cutoff, servo OFF or all-axis servo OFF (initialization error, power error, etc.), all programs other than the "I/O processing program at operation/program abort" will be cancelled.	All programs other than the "I/O processing program at operation/program abort" will be cancelled.	Not enabled.	The controller power must be reconnected (MAIN only). (The CPU and OS will run properly.)
	MAIN core	-							
	PC								
	PC (Update tool)								
	TP								
	MAIN application	600 ~ 6CF							
	MAIN core	-							
	PC	6D0 ~ 6DF							
	PC (Update tool)	6E0 ~ 6EF							
	TP	6F0 ~ 6FF							
	MAIN application	D00 ~ D8F							
	MAIN core	D90 ~ DAF							
	PC	DB0 ~ DCF							
	PC (Update tool)	DD0 ~ DDF							
	TP	DE0 ~ DFF							
	MAIN application	E00 ~ E8F							
	MAIN core	E90 ~ EBF							
	PC	EC0 ~ EDF							
	TP	EE0 ~ EFF							
System-down level	MAIN application		○	○	○	All programs will be cancelled.		Not enabled.	The controller power must be reconnected (MAIN only). (The CPU and OS will not run.)
	MAIN core	-							
	PC								
	PC (Update tool)								
	TP								
	MAIN application	FF0 ~ FBF							
	MAIN core	FC0 ~ FCF							
	PC	FD0 ~ FDF							
	TP	FE0 ~ FEF							

Note) Secret-level errors are not actual errors. Internal statuses are registered in an error list as secret-level errors, when deemed necessary, in order to facilitate error analysis.
PC: PC software TP: Teaching pendant

◎ Error List (MAIN application) (In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
200	Encoder parameter data version mismatch warning	The version of encoder parameter data is not supported by this controller. Update the encoder parameters.
203	Drive-source cutoff relay DET (MELT) error	The drive-source cutoff relay may have fused.
206	Updating system mode error (IAI protocol)	An update command was received other than in the update mode.
207	Update file name error (IAI protocol)	The name of the update program file selected in the update mode is invalid. Select the correct file and repeat the updating procedure from the beginning.
208	Time data error	The time data is invalid. Check the data.
209	Unsupported control constant table ID error	The control constant table ID is not supported. Check the data.
20A	Control constant table change/query error	The message of the control constant table change/query command contains error. Check the message that has been sent.
20B	Control constant table write data type specification error	The specified control constant table write data type is invalid. Check the message that has been sent.
20C	Control constant table management information mismatch error	The management information regarding the control constant table is invalid. Confirm that the control constant table is supported by the controller.
20D	Flash busy reset timeout error	Error erasing/writing the flash ROM
20E	Motorola S-byte count error	The update program file is invalid. Check the file.
20F	Updating target specification error (Received by the application)	The system application received an updating target specification command. To update the program, restart the controller and repeat the updating procedure from the beginning.
210	Program-related data change/run command rejection error in positioner mode	Change of program-related data or running of programs is prohibited in the positioner mode.

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
406	Flash busy reset timeout	Error erasing/writing the flash ROM
407	Control constant table management information mismatch error	The management information regarding the control constant table is invalid. If this error occurs when the controller is started, the control constant table may need to be updated.
408	Control constant table ID error	The control constant table ID is invalid.
409	Encoder control constant error (power-source voltage control)	An encoder control constant relating to power-source voltage control is invalid. The encoder power-source voltage cannot be adjusted (the encoder power will be supplied without voltage adjustment).
40A	Encoder power-source voltage calculation error	The encoder power-source voltage cannot be adjusted (the encoder power will be supplied without voltage adjustment). Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
40B	Speed control parameter calculation error	Check driver parameter Nos. 38, 39, 40, 43, 44, 45, etc.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
605	Forced discharge error	Abnormal forced discharge. The drive-source cutoff relay may be abnormal. The power must be reconnected.
606	Regenerative discharge error	Abnormal regenerative discharge. The power must be reconnected.
607	Motor power-source voltage low error	Low voltage was detected in the motor power circuit.
608	Power-supply board FRCDCSTR-ON timeout error	Power-supply board FRCDCSTR-ON could not be confirmed within the specified time.
609	Power-supply board RBONSTR-ON timeout error	Power-supply board RBONSTR-ON could not be confirmed within the specified time.
60A	Power-supply board RBONSTR-OFF timeout error	Power-supply board RBONSTR-OFF could not be confirmed within the specified time.
60B	Power-supply board FRCDCSTR-OFF timeout error	Power-supply board FRCDCSTR-OFF could not be confirmed within the specified time.
60C	Power-system overheat error	An overheated power-supply board, regenerative resistor, etc., was detected. The power must be reconnected.
60D	Slave board CPU ready OFF error (other than power supply)	A ready status of the driver board, etc. (other than power-supply board) cannot be confirmed.
60E	Dynamic brake ON/OFF timeout error	Dynamic brake ON/OFF cannot be confirmed within the specified time.
613	Driver synchronous communication driver read error	A communication failure occurred between the driver board and FPGA (main).
614	Driver synchronous communication LRC error	A communication failure occurred between the driver board and FPGA (main).
615	Driver synchronous communication toggle error	A communication failure occurred between the driver board and FPGA (main).
623	Driver error detail code acquisition error	A driver error occurred, but an error detail code could not be acquired.
624	Undefined driver error	A driver error occurred.
625	Driver-side detection synchronous communication error	A communication failure occurred between the driver board and FPGA (main).
626	Driver IPM15V voltage low error	A low voltage was detected in the driver IPM15V circuit.
627	Driver current detection A/D offset over error	A driver current detection A/D offset error was detected.
628	Driver error	(Driver error for future expansion)
629	Driver error	(Driver error for future expansion)
62A	Driver error	(Driver error for future expansion)
62B	Driver error	(Driver error for future expansion)
62C	Driver error	(Driver error for future expansion)
62D	Driver error	(Driver error for future expansion)
62E	Driver error	(Driver error for future expansion)
62F	Driver error	(Driver error for future expansion)

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
630	Updating system code error (Application detection)	The updating system code is invalid.
631	Updating unit code error (Application detection)	The updating unit code is invalid.
632	Updating device number error (Application detection)	The updating device number is invalid.
633	Feedback pulse synchronization error (Detected in the speed loop)	Abnormal feedback pulse synchronization (detected in the speed loop).
634	Feedback pulse synchronization error (Detected in the position loop)	Abnormal feedback pulse synchronization (detected in the position loop).
635	Deadman/enable switch requiring reset recovery open	Reset the deadman/enable switch, and then reconnect the power.
636	Serial encoder command busy error	The system was busy when the serial encoder command was issued.
637	Serial encoder command timeout error	Completion of the serial encoder command cannot be confirmed after the specified time.
638	Speed control parameter setting command busy error	The system was busy when the speed control parameter setting command was issued.
639	Speed control parameter setting command timeout error	Completion of the speed control parameter setting command cannot be confirmed after the specified time.
63A	ABZ encoder logic error	An encoder phase-A/B electrical level pattern error was detected. The power must be reconnected.
63B	Encoder/motor control constant table flash ROM status error	Data is not written correctly to the flash ROM, or the data is of an old, incompatible version.
63C	Encoder/motor control constant table checksum error	The flash ROM data is corrupted.
63D	ABZ encoder specification error	An ABZ encoder cannot be installed for this axis. Check the “motor/encoder configuration information” in driver parameter No. 26 and encoder parameter No. 11.
63E	ABZ encoder magnetic-pole sensor signal logic error	Check if the encoder cable is connected.
63F	Encoder control constant error	The encoder control constant is invalid.
640	Motor control constant error	The motor control constant is invalid.
641	Encoder power-source voltage control parameter error	Check driver parameter Nos. 32, 33, etc.
642	Speed loop parameter error	Check driver parameter Nos. 43, 44, 45, etc.
643	Encoder resolution division error	Check “Axis-specific parameter No. 43: Encoder division ratio.”
644	Encoder/motor combination mismatch error (encoder resolution)	Check driver parameter No. 26, encoder parameter No. 11.
645	DAC transfer completion check timeout error when encoder power was supplied	A timeout occurred during DAC transfer when the encoder power was supplied.
646	Encoder EEPROM read busy error	The encoder is faulty or an encoder communication failure occurred.
647	Encoder EEPROM write address mismatch error	The encoder is faulty or an encoder communication failure occurred.
648	Encoder EEPROM read address mismatch error	The encoder is faulty or an encoder communication failure occurred.
649	Undefined serial encoder installation error	Installation of serial encoder is not defined. Check the “motor/encoder configuration information” in driver parameter No. 26 and encoder parameter No. 11.
64A	Undefined serial encoder command error	The serial encoder command is not defined.

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
64B	Serial encoder command packet error	The serial encoder command packet is invalid.
64C	1-revolution data reset error at servo ON (serial encoder command)	A 1-revolution data reset was commanded when the servo was ON. Turn OFF the servo.
64D	Encoder reset command timeout error (serial encoder command)	An encoder communication failure.
64E	ABS data query command timeout error (serial encoder command)	An encoder communication failure.
64F	Encoder error reset error at servo ON (serial encoder command)	Turn OFF the servo before resetting an encoder error.
650	Encoder receive timeout error (during initialization communication)	An encoder communication failure.
651	Speed control interruption control job error	The speed control interruption error job is invalid.
652	Serial encoder command control job error	The serial encoder command control job is invalid.
653	Encoder control job logic error	The encoder control job logic is invalid.
654		
655	Encoder receive timeout error at serial encoder command issuance	An encoder communication failure.
656	Torque limit logic error	The torque limit logic is invalid.
657	Torque limit parameter error	Check driver parameter Nos. 38, 39, 40, etc.
658	Movement error during ABZ encoder counter initialization	Axis movement was detected while initializing the ABZ encoder counter following power on. The power may have been turned on or a software reset executed while the actuator was moving due to external force such as reactive force of a self-supported cable or while the installation location was vibrating.
65A	Unsupported encoder ID error	The encoder is not supported. No encoder control constant record is available that corresponds to the encoder ID. Check the installed encoder.
65B	Unsupported encoder error (main information)	The encoder is not supported. No encoder control constant record is available that corresponds to the encoder ID, or the record is invalid. Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
65C	Unsupported motor error (main information)	The motor is not supported. No motor control constant record is available that corresponds to the motor ID, or the record is invalid. Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
65D	Unsupported motor error (driver information)	The motor is not supported. The motor ID bit number is outside the range of "maximum supported motor ID number" when the driver parameter, "Use motor control data in driver flash ROM" is specified. Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
65E	Current detection circuit type mismatch error	The motor control constant, "Current detection circuit specification" does not match the driver parameter, "Installation type word 1, current detection circuit type." Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
65F	Main/driver motor control data mismatch error	A motor control constant does not match the corresponding driver parameter (rated speed, maximum speed, rated current, maximum current number of pole pairs, linear motor lead, linear motor specification). Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
660	Maximum motor speed mismatch error	The axis-specific parameter, "Maximum motor speed" does not match the motor control constant, "Maximum speed." Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
661	Encoder/motor combination mismatch error (linear/rotary type)	The linear/rotary type does not match between the encoder and motor. Check the "motor/encoder configuration information" in driver parameter No. 26 and encoder parameter No. 11.
662	Mechanical angle 360-degree pulse count calculation error	The calculated pulse count based on 360 mechanical angle degrees is invalid. (The calculated value is "0," or in the case of a linear encoder, the calculated value has fraction.)
663	Software DB specification error	The value in the driver parameter, "Software DB specification" is invalid.
664	Current control band number specification error	The value in the driver parameter, "Current control band number" is invalid.
665	Driver/encoder communication line channel number specification error	All-axis parameter No. 101 or 102, "Driver/encoder communication line channel setting" is invalid (invalid value, duplicate specifications).
666	Driver initialization communication type specification error	All-axis parameter No. 103 or 104, "Driver initialization communication type setting" is invalid (invalid value, duplicate specifications, mismatch).
667	Invalid driver initialization communication line specification error at specification of valid axis	Initialization communication line channel number is not specified for a valid axis. Check all-axis parameter No. 1, "Valid axis pattern," Nos. 101 and 102, "Driver/encoder communication line channel setting" and Nos. 103 and 104, "Driver initialization communication type setting."
668	Driver target information initialization error	The initialization sequence of driver target information did not complete successfully. Check the installed driver board. Check all-axis parameter Nos. 101, 102, 103 and 104, or driver parameter No. 26, encoder parameter No. 11.
669	Encoder target information initialization error	The initialization sequence of encoder target information did not complete successfully. Check the installed encoder. Check all-axis parameter Nos. 101, 102, 103 and 104, or driver parameter No. 26, encoder parameter No. 11.
66A	Power-system target information initialization error	The initialization sequence of power-system target information did not complete successfully. Check the installed power-supply board. Check the power-supply board parameters.
66B	Slave communication error response error	An error response was received during slave communication.
66C	SCI LRC error (slave communication)	The message LRC of slave communication is invalid.
66D	Slave communication target ID error	The target ID of slave communication is invalid.
66E	Slave communication block number error	The block number of slave communication is invalid.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
66F	Target specification error due to no axis number	The specified target of slave communication (driver or encoder) is invalid (no axis number is assigned for the target ID, or an internal driver board axis is specified).
670	Target board type error	The target board type is invalid.
671	Encoder control data error	The encoder control data is invalid or cannot be acquired. Take the same actions specified for error Nos. 65A, 65B and 669.
672	Motor control data error	The motor control data is invalid or cannot be acquired. Take the same actions as specified for error Nos. 65C, 65D, 668 and 669.
680	Magnetic-pole detection parameter error	Invalid parameter used for magnetic-pole detection. Check driver parameter Nos. 49, 50, etc.
682	I/O function specification error	Wrong I/O function specification. Check I/O parameter Nos. 30 through 61 and 251 through 282.
683	Axis operation error in system semi-locked (encoder stopped) status	An attempt was made to operate an axis by turning on the servo, executing an absolute reset, etc., when the system was in semi-locked status (encoder was stopped).
690	Motor overcurrent error	Excessive current flew through the motor.
691	Driver error	(Driver error for future expansion)
692	Driver error	(Driver error for future expansion)
693	Driver error	(Driver error for future expansion)
694	Driver error	(Driver error for future expansion)
695	Driver error	(Driver error for future expansion)
696	Driver error	(Driver error for future expansion)
697	Driver error	(Driver error for future expansion)
698	Driver error	(Driver error for future expansion)
699	Driver error	(Driver error for future expansion)

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
801	SCIF overrun status (IAI protocol reception)	Communication failure. Check for noise, connected equipment and communication setting.
802	SCIF receive ER status (IAI protocol reception)	Communication failure. Check for noise, shorted/disconnected communication cable, connected equipment and communication setting. This error will also occur when establishing communication with the PC/TP wrongly connected to SIO-CH1 being opened to the user.
803	Receive timeout status (IAI protocol reception)	The transfer interval after the first received byte is too long. Possible causes include disconnected communication cable and error in the connected equipment.
804	SCIF overrun status (SEL reception)	Communication failure. Check for noise, connected equipment and communication setting.
805	SCIF receive ER status (SEL reception)	Communication failure. Check for noise, shorted/disconnected communication cable, connected equipment and communication setting.
806	SCIF receive ER status due to other factor (SEL reception)	Communication failure. Take the same action specified for error No. 804 or 805.
807	Drive-source cutoff relay ER status	The motor-drive power ON status remains ON even when the drive source is cut off. The drive-source cut-off relay contacts may have been melted.
808	Power OFF status during slave parameter write	The power was turned off while writing slave parameters. (This error can be detected only when a backup battery is used.)
809	Power OFF status during data write to flash ROM	The power was turned off while writing data to the flash ROM. (This error can be detected only when a backup battery is used.)
80F	Ethernet control status 1	Ethernet control information (for analysis)
810	Ethernet control status 2	Ethernet control information (for analysis)
811	Maintenance information 1	Maintenance information (for analysis)
812	Maintenance information 2	Maintenance information (for analysis)
813	Maintenance information 3	Maintenance information (for analysis)
814	Maintenance information 4	Maintenance information (for analysis)
815	Maintenance information 5	Maintenance information (for analysis)
820	DRV status 820 (TO_SELECTEDDATA)	(This is not an error, but maintenance information.)

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
900	Blank step shortage error	There are not enough blank steps to save step data. Provide enough blank steps needed to save step data.
901	Step number error	The step number is invalid.
902	Symbol-definition table number error	The symbol-definition table number is invalid.
903	Point number error	The point number is invalid.
904	Variable number error	The variable number is invalid.
905	Flag number error	The flag number is invalid.
906	I/O port/flag number error	The I/O port/flag number is invalid.
910	Command error (IAI protocol HT reception)	The command ID is not supported or invalid. (For future expansion)
911	Message conversion error (IAI protocol HT reception)	The transmitted message does not match the message format or contains invalid data. (For future expansion)
912	PC/TP servo-movement command acceptance-enable input OFF error	Any axis movement command issued to the axis specified in I/O parameter No. 78 from the PC/TP will not be accepted while the input port specified in I/O parameter No. 77 is OFF. (Important: The acceptance-enable input port will become invalid once the operation is started.)
913	Multiple-program simultaneous start inhibition error	Simultaneously starting of multiple programs is inhibited.
914	Absolute-data backup battery voltage error	Check the connection of the absolute-data backup battery and replace the battery if necessary, and also check the connection of the encoder cable, and then perform an absolute reset.
A01	System-memory backup battery voltage-low warning	The voltage of the system-memory backup battery is low. Replace the battery. (Above the minimum data-backup voltage)
A02	Abnormal system-memory backup battery voltage	The voltage of the system-memory backup battery is low. Replace the battery. (Below the minimum data-backup voltage)
A03	Absolute-data backup battery voltage-low warning (Driver analysis)	The voltage of the absolute-data backup battery is low. Check the battery connection or replace the battery.
A04	System mode error at core update	An update command was received when the system was not in the core update mode. Before updating the core, confirm that a chip resistance for setting core update mode is provided on the board. (For maintenance)
A05	Motorola S record format error	The update program file is invalid. Check the file.
A06	Motorola S checksum error	The update program file is invalid. Check the file.
A07	Motorola S load address error	The update program file is invalid. Check the file.
A08	Motorola S write address over error	The update program file is invalid. Check the file.
A09	Flash-ROM timing limit over error (Write)	Error writing the flash ROM
A0A	Flash-ROM timing limit over error (Erase)	Error erasing the flash ROM

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A0B	Flash-ROM verify error	Error erasing/writing the flash ROM
A0C	Flash-ROM ACK timeout	Error erasing/writing the flash ROM
A0D	Head sector number specification error	Error erasing the flash ROM
A0E	Sector count specification error	Error erasing the flash ROM
A0F	Write-destination offset address error (Odd-numbered address)	Error writing the flash ROM
A10	Write-source data buffer address error (Odd-numbered address)	Error writing the flash ROM
A11	Invalid core-code sector block ID error	The core program already written to the flash ROM is invalid.
A12	Core-code sector block ID erase count over	The number of times the flash ROM can be erased was exceeded.
A13	Flash-ROM write request error when erase is incomplete	When updating, a flash-ROM write command was received before a flash-ROM erase command. Check the update program file and perform update again.
A14	Busy-status reset timeout error at EEPROM write	A busy-status reset timeout occurred after executing EEPROM write.
A15	EEPROM write request error due to no-EEPROM in target	An EEPROM write request was received for a driver or other unit with CPU not equipped with EEPROM.
A16	EEPROM read request error due to no-EEPROM in target	An EEPROM read request was received for a driver or other unit with CPU not equipped with EEPROM.
A17	Message checksum error (IAI protocol reception)	The checksum in the received message is invalid.
A18	Message header error (IAI protocol reception)	The header in the received message is invalid. Invalid header position (message is 9 bytes or less) is suspected, among other reasons.
A19	Message station number error (IAI protocol reception)	The station number in the received message is invalid.
A1A	Message ID error (IAI protocol reception)	The ID in the received message is invalid.
A1C	Message conversion error	The transmitted message does not match the message format or contains invalid data. Check the transmitted message.
A1D	Start mode error	A start not permitted in the current mode (MANU/AUTO) was attempted.
A1E	Start condition non-satisfaction error	Start was attempted when the start condition was not satisfied, such as when an all-operation-cancellation factor (see the 7-segment display: Drive-source cutoff, mode switching, error, auto-start switch OFF edge, deadman switch, safety gate, emergency stop, etc.) was present or the flash ROM was being written.
A1F	Axis duplication error (SIO · PIO)	The applicable axis is currently in use.
A20	Servo-control-right acquisition error (SIO · PIO)	The servo control right is not available.
A21	Servo-control-right duplicate-acquisition error (SIO · PIO)	The servo control right has already been acquired.
A22	Servo-control-right non-acquisition error (SIO · PIO)	An attempt to retain the servo control right has failed.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A23	Absolute-data backup battery voltage-low warning (Main analysis)	The voltage of the absolute-data backup battery is low. Check the battery connection or replace the battery.
A25	Step count specification error	The specified number of steps is invalid.
A26	Program count specification error	The specified number of programs is invalid.
A27	Program non-registration error	The applicable program is not registered.
A28	Reorganization disable error during program run	A program-area reorganization operation was attempted while a program was running. End all active programs first.
A29	Active-program edit disable error	An edit operation was attempted to a program currently not running. End the applicable program first.
A2A	Program inactive error	The specified program is not running.
A2B	Program-run command refusal error in AUTO mode	Programs cannot be run from the TP/PC software connector in the AUTO mode.
A2C	Program number error	The program number is invalid.
A2D	Inactive program resumption error	A resumption request was received for a program currently not running.
A2E	Inactive program pause error	A pause request was received for a program currently not running.
A2F	Breakpoint error	The step number specified as a breakpoint is invalid.
A30	Breakpoint setting-count specification error	The number of breakpoints to be set exceeds the limit value.
A31	Parameter change value error	The value of parameter changed is invalid.
A32	Parameter type error	The parameter type is invalid.
A33	Parameter number error	The parameter number is invalid.
A34	Card-parameter buffer read error	Error reading the card-parameter buffer
A35	Card-parameter buffer write error	Error writing the card-parameter buffer
A36	Parameter change refusal error during operation	Parameters cannot be changed during operation (program is running, servo is in use, etc.).
A37	Card manufacturing/function information change refusal error	The card manufacturing/function information cannot be changed.
A38	Parameter change refusal error during servo ON	An attempt was made to change a parameter whose change is not permitted while the servo is ON.
A39	Non-acquired card parameter change error	An attempt was made to change a parameter for a card not recognized at reset.
A3A	Device number error	The device number is invalid.
A3C	Memory initialization type specification error	The specified memory initialization type is invalid.
A3D	Unit type error	The unit type is invalid.
A3E	SEL write data type specification error	The specified SEL write data type is invalid.
A3F	Flash-ROM write refusal error during program run	The flash ROM cannot be written while a program is running.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A40	Data change refusal error during flash ROM write	Data cannot be changed while the flash ROM is being written.
A41	Duplicate flash-ROM write commands refusal error	Another flash-ROM write command was received while the flash ROM was being written.
A42	Direct monitor prohibition error during flash ROM write	Direct monitor is prohibited while the flash ROM is being written.
A43	P0/P3-area direct monitor prohibition error	Direct monitor in the P0/P3 areas is prohibited.
A44	Point-data count specification error	The specified number of point data is invalid.
A45	Symbol-record count specification error	The specified number of symbol records is invalid.
A46	Variable-data count specification error	The specified number of variable data is invalid.
A48	Error-detail query type 1 error	Error-detail query type 1 is invalid.
A49	Error-detail query type 2 error	Error-detail query type 2 is invalid.
A4A	Monitoring data type error	The data type for monitoring data query is invalid.
A4B	Monitoring-record count specification error	The specified number of records for monitoring data query is invalid.
A4C	Monitoring-operation special command register busy error	The driver special command ACK generated a timeout during monitoring operation.
A4E	Parameter register busy error at issuance of slave command	The driver special command ACK generated a timeout at issuance of a slave command.
A4F	Software reset refusal error during operation	Software reset (SIO) is prohibited during operation (program is running, servo is in use, etc.).
A50	Drive-source recovery request refusal error	The drive-source cutoff factor (error, deadman switch, safety gate, emergency stop, etc.) has not been removed.
A51	Operation-pause reset request refusal error	The all-operation-pause factor (drive-source cutoff, operation-pause signal, deadman switch, safety gate, emergency stop, etc.) has not been removed.
A53	Refusal error due to servo ON	A processing not permitted during servo ON was attempted.
A54	Refusal error due to unsupported function	The function is not supported.
A55	Refusal error due to exclusive manufacturer function	A processing not opened to users other than the manufacturer was attempted.
A56	Refusal error due to invalid data	The data is invalid.
A57	Program start duplication error	An attempt was made to start a program currently running.
A58	BCD error warning	The BCD value being read may be invalid, or the value being written (variable 99) may be a negative value, among other reasons.
A59	IN/OUT command port flag error warning	The number of I/O ports (flags) may have exceeded 32, among other reasons. Check the I/O port (flag) specifications.
A5B	Character-string → value conversion error warning	The specified number of converting characters is invalid or characters that cannot be converted to value are included.
A5C	Copying-character count error warning with SCPY command	The specified number of copying characters is invalid.
A5D	SCIF open error in non-AUTO mode	The channel was opened in a non-AUTO mode. In the MANU mode, the PC/TP connection must be forcibly disconnected before opening the serial channel opened to the user. Exercise caution.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A5E	I/O-port/flag count specification error	The specified number of I/O ports/flags is invalid.
A5F	Fieldbus error (LERROR-ON)	A LERROR-ON was detected.
A60	Fieldbus error (LERROR-BLINK)	A LERROR-BLINK was detected.
A61	Fieldbus error (HERROR-ON)	A HERROR-ON was detected.
A62	Fieldbus error (HERROR-BLINK)	A HERROR-BLINK was detected.
A63	Fieldbus not ready	Fieldbus ready cannot be confirmed.
A69	Data change refusal error during operation	An attempt was made to change data whose change is prohibited during operation (program is running, servo is in use, etc.).
A6A	Software reset refusal error during write	Software reset is prohibited while data is being written to the flash ROM or slave parameters are being written.
A6B	Fieldbus error (FBRS link error)	A FBRS link error was detected.
A6C	PC/TP start command refusal error in AUTO mode	Starting from the PC software/TP connector is prohibited in the AUTO mode.
A6D	P0/P3/FROM-area direct write prohibition error	Direct write to the P0/P3/FROM areas is prohibited.
A6E	Refusal error during write	A processing not permitted while data is being written to the flash ROM or slave parameters are being written was attempted.
A6F	Driver monitor type mismatch error	The monitor type supported by the standard DIO board or based on the capacity of FROM on the main CPU board does not match the monitor type on the PC software side (selected on the monitor screen).

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
B00	SCHA setting error	The setting of SCHA command is invalid.
B01	TPCD setting error	The setting of TPCD command is invalid.
B02	SLEN setting error	The setting of SLEN command is invalid.
B03	Home-return method error	The setting of "Axis-specific parameter No. 10, Home-return method" is invalid. (Not incremental encoder AND current position 0 home is specified, etc.)
B04	1-shot-pulse output excessive simultaneous use error	The number of BTPN and BTPF timers operating in one program simultaneously exceeds the upper limit (16).
B05	Estimate-stroke over error at home return	The operation at home return exceeded the estimate stroke. The home sensor or creep sensor may be faulty, among other reasons.
B10	Phase-Z search timeout error	Phase Z cannot be detected. Check for operation restriction, wiring, encoder, motor, etc.
B11	Home-sensor pull-out timeout error	Pull-out from the home sensor cannot be confirmed. Check for operation restriction, wiring, motor, home sensor, etc.
B12	Storage variable number error for SEL command return code	The variable number specified for storing SEL command's return code is invalid.
B13	Backup SRAM data checksum error	The backup SRAM data has been destroyed. Check the battery.
B15	Input-port debug filter type error	The setting of input-port debug filter type is invalid.
B16	SEL operand specification error	The operand specification of SEL command is invalid.
B17	Parameter register busy error at issuance of slave command	The driver special command ACK generated a timeout at issuance of a slave command.
B18	Device number error	The device number is invalid.
B19	Unit type error	The unit type is invalid
B1A	Absolute reset specification error	The specification for absolute reset using an optional function, etc., is invalid. (Two or more axes are specified simultaneously, non-absolute-encoder axis is specified, etc.)
B1B	Ethernet non-closed socket open error	An attempt was made to open a socket without closing it first.
B1C	Ethernet in-use-by-other-task error	An attempt was made to open a channel already opened by other task.
B1D	Ethernet non-open error	An attempt was made to use a channel not opened by own task.
B1E	Ethernet multiple WRIT execution error	WRIT commands were executed simultaneously by multiple tasks for the same channel.
B1F	Ethernet job busy error	An attempt was made to start a new process when the Ethernet mailbox control job was busy.
B20	Ethernet non-initialization device use error	An attempt was made to use the Ethernet system when Ethernet device initialization was not yet complete. Check I/O parameter Nos. 123 to 159, 14, 15, etc., depending on the purpose of use.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
B21	Ethernet IP address error	An error will generate under the following conditions during normal use. When IP address (H) (first octet) through IP address (L) (fourth octet) are given as IP_H, IP_MH, IP_ML and IP_L, the error conditions are described as follows: IP_H ≤ 0 or IP_H = 127 or IP_H > 255 or IP_MH < 0 or IP_MH > 255 or IP_ML < 0 or IP_ML > 255 or IP_L ≤ 0 or IP_L ≥ 255 Check I/O parameter Nos. 132 to 135, 149 to 152, and 154 to 157, the IP address of connection destination specified by an IPCN command in an integer variable, or the like.
B22	Ethernet port number error	An error will generate if own port number < 1025, or own port number > 65535, or own port number duplication, or connection-destination port number for client ≤ 0, or connection-destination port number for client > 65535, or connection-destination port number for server < 0, or connection-destination port number for server > 65535 is satisfied. Check I/O parameter Nos. 144 to 148, 159, 153, and 158, the port number of connection destination specified by an IPCN command in an integer variable, or the like.
B86	SEL PTRQ command preprocessing error	The PTRQ command setting is abnormal. Check the setting for abnormality, such as deviation from the allowable range.
B92	Excessive arc interpolation radius error	The radius of arc interpolation is too large. Use a CIR/ARC command, etc.
C02	Executable program count over error	Execution requests were received for programs exceeding the number that can be executed simultaneously.
C03	Non-registered program specification error	The specified program is not registered.
C04	Program entry point non-detection error	A request was made to execute a program number for which no program steps are registered.
C05	Program first-step BGSR error	The program specified for execution starts with BGSR.
C06	Executable step non-detection error	The program specified for execution does not contain executable program steps.
C07	Subroutine non-definition error	The subroutine specified for call is not defined.
C08	Subroutine duplicate-definition error	The same subroutine number is defined at multiple locations.
C0A	Tag duplicate-definition error	The same tag number is defined at multiple locations.
C0B	Tag non-definition error	The tag specified as the jump destination of a GOTO statement is not defined.
C0C	DW/IF/IS/SL pair-end mismatch error	The branching command syntax is invalid. Correspondence with the last appearing branching command is invalid when EDIF, EDDO or EDSL is used. Check the correspondence between IF/IS command and EDIF, DO command and EDDO or SLCT command and EDSL.
C0D	DW/IF/IS/SL no pair-end error	EDIF, EDDO or EDSL is not found. Check the correspondence between IF/IS command and EDIF, DO command and EDDO or SLCT command and EDSL.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
C0E	BGSR no pair-end error	There is no EDSR for BGSR, or no BGSR for EDSR. Check the correspondence between BGSR and EDSR.
C0F	DO/IF/IS over-nesting error	The number of nests in a DO or IF/IS command exceeds the limit value. Check for excessive nesting or branching out of or into the syntax using a GOTO command.
C10	SLCT over-nesting error	The number of nests in a SLCT command exceeds the limit value. Check for excessive nesting or branching out of or into the syntax using a GOTO command.
C11	Subroutine over-nesting error	The number of nests in a subroutine exceeds the limit value. Check for excessive nesting or branching out of or into the syntax using a GOTO command.
C12	DO/IF/IS under-nesting error	The EDIF or EDDO position is invalid. Check the correspondence between IF/IS command and EDIF or DO command and EDDO, or branching out of or into the syntax using a GOTO command.
C13	SLCT under-nesting error	The EDSL position is invalid. Check the correspondence between SLCT and EDSR, or branching out of or into the syntax using a GOTO command.
C14	Subroutine under-nesting error	The EDSR position is invalid. Check the correspondence between BGSR and EDSR, or branching out of or into the syntax using a GOTO command.
C15	SLCT next-step command code error	The program step next to SLCT must be WHEQ, WHNE, WHGT, WHGE, WHLT, WHLE, WSEQ, WSNE, OTHE or EDSL.
C16	Create stack failed	Initialization of the input-condition-status storage stack has failed.
C17	Expansion-condition code error	Input program step error. The expansion condition code is invalid.
C18	Expansion-condition LD simultaneous processing over error	The number of LDs processed simultaneously exceeds the limit value.
C19	Expansion-condition LD shortage error 1	There is not enough LD when expansion condition A or O is used.
C1A	Expansion-condition LD shortage error 2	There is not enough LD when expansion condition AB or OB is used.
C1C	Unused-LD detection error	An attempt was made to execute a command based on multiple LD condition that has been saved, without using it in expansion condition AB or OB.
C1F	Input-condition CND shortage error	The necessary input condition is not found when an expansion condition is used.
C21	Input-condition use error with input-condition prohibited command	Input-condition prohibited commands prohibit the use of input conditions.
C22	Invalid command position error with input-condition prohibited command	A command for which input condition is prohibited cannot be included in an input condition nest.
C23	Invalid operand error	Program step error. The necessary operand data is invalid.
C24	Operand type error	Program step error. The operand data type is invalid.
C25	Actuator control declaration error	The setting of actuator control declaration command is invalid.
C26	Timer setting-range over error	The timer setting is invalid.
C27	Timeout setting-range over error during wait	The timeout setting is invalid.
C28	Tick count setting-range error	The Tick count setting is invalid.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
C29	DIV command divisor 0 error	“0” was specified as the divisor in the DIV command.
C2A	SQR command range error	The operand value in the SQR command is invalid. Input a value larger than “0” as data in a SQR command.
C2B	BCD display digit range error	The specified number of BCD display digits is invalid. Specify a value between 1 and 8.
C2C	Program number error	The program number is invalid.
C2D	Step number error	The step number is invalid.
C2E	Blank step shortage error	There are not enough blank steps to save step data. Provide enough blank steps needed to save step data.
C2F	Axis number error	The axis number is invalid.
C30	Axis pattern error	The axis pattern is invalid.
C32	Operating-axis addition error during command execution	An operating axis for point data was added during continuous point movement or push-motion movement calculation.
C33	Base axis number error	The base axis number is invalid.
C34	Zone number error	The zone number is invalid.
C35	Point number error	The point number is invalid.
C36	I/O port/flag number error	The I/O port/flag number is invalid.
C37	Flag number error	The flag number is invalid.
C38	Tag number error	The tag number is invalid.
C39	Subroutine number error	The subroutine number is invalid.
C3A	User-open communication channel number error	The channel number of the communication channel opened to the user is invalid.
C3B	Parameter number error	The parameter number is invalid.
C3C	Variable number error	The variable number is invalid.
C3D	String number error	The string number is invalid.
C3E	String-variable data count specification error	The specified number of string variables exceeds the area, etc.
C40	String-variable delimiter non-detection error	Delimiter cannot be detected in the string variable.
C41	String-variable copy size over error	The copy size of string variable is too large.
C42	Character count non-detection error during string processing	The character-string length is not defined in string processing. Execute a string processing command after defining the length with a SLEN command.
C43	Character-string length error during string processing	The character-string length used in string processing is invalid. Check the value of character-string length defined by a SLEN command.
C45	Symbol definition table number error	The symbol definition table number is invalid.
C46	Blank area shortage error with source-symbol storage table	There is not enough area to store the source symbols. Check the number of times source symbol can be used.

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
C47	Symbol search error	Definitions are not found for the symbols used in the program steps.
C48	SIO-message continuous conversion error	The transmitted SIO message does not match the message format or contains invalid data. Check the transmitted message.
C49	SEL-SIO in-use error	The SIO is being used by other interpreter task.
C4A	SCIF unopen error	Serial channel 1 opened to the user is not opened in the target task. Open the channel using an OPEN command first.
C4B	Delimiter non-definition error	An end character is not defined. Set an end character using a SCHA command first.
C4E	SIO1 invalid usage OPEN error	The usage of serial channel opened to the user does not match the parameter. Check "I/O parameter No. 90, Usage of SIO channel opened to user."
C4F	SEL program/source symbol checksum error	The flash ROM data has been destroyed.
C50	Symbol definition table checksum error	The flash ROM data has been destroyed.
C51	Point data checksum error	The flash ROM data has been destroyed.
C52	Backup SRAM data destruction error	The backup SRAM data has been destroyed. Check the battery.
C53	Invalid flash-ROM SEL global data/error list error	The SEL global data/error lists in the flash ROM are invalid.
C54	Flash-ROM SEL global data/error list duplication error	The SEL global data/error lists in the flash ROM are duplicated.
C55	Flash-ROM erase count over error for SEL global data/error lists	The number of time the flash ROM containing SEL global data/error lists can be erased was exceeded.
C56	Timing limit over error (Flash ROM erase)	Error erasing the flash ROM
C57	Flash-ROM verify error (Flash ROM erase)	Error erasing the flash ROM
C58	Flash-ROM ACK timeout error (Flash ROM erase)	Error erasing the flash ROM
C59	Head sector number specification error (Flash ROM erase)	Error erasing the flash ROM
C5A	Sector count specification error (Flash ROM erase)	Error erasing the flash ROM
C5B	Timing limit over error (Flash ROM write)	Error writing the flash ROM
C5C	Flash-ROM verify error (Flash ROM write)	Error writing the flash ROM
C5D	Flash-ROM ACK timeout error (Flash ROM write)	Error writing the flash ROM
C5E	Write-destination offset address error (Flash ROM write)	Error writing the flash ROM
C5F	Write-source data buffer address error (Flash ROM write)	Error writing the flash ROM
C60	No SEL global data/error list write area error	There is no area to write the erased SEL global data/error lists.
C61	SEL-data flash-ROM erase count over error	The number of times the flash ROM containing SEL data can be erased was exceeded.
C62	Operation command error at servo OFF	An attempt was made to execute an operation command when the servo was OFF.
C63	Servo operation condition error	The servo is not in an operation-enabled condition.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
C64	Invalid servo acceleration/deceleration error	The internal servo acceleration/deceleration is invalid.
C65	Servo ON/OFF logic error	The servo ON/OFF logic between the main and driver is invalid.
C66	Axis duplication error	An attempt was made to acquire the control right to an axis already in use.
C67	Servo-control-right acquisition error	There is no space in the servo user management area.
C68	Servo-control-right duplicate-acquisition error	The servo control right has already been acquired.
C69	Servo-control-right non-acquisition error	A user who doesn’t have the servo control right attempted to retain the control right.
C6A	Push-motion flag logic error	The internal logic for push-motion processing is invalid.
C6B	Deviation overflow error	The command cannot be followed. Check for operation restriction, wiring, encoder, motor, etc.
C6C	Movement error during absolute data acquisition	Axis movement was detected while acquiring absolute encoder data after the power was turned on. The power may have been turned or a software reset executed while the actuator was moving due to external force such as reactive force of a self-supported cable or while the installation location was vibrating. Or, a software reset may have been executed. Absolute coordinates cannot be confirmed in this condition.
C6D	Maximum installable axes over error	The specified number of axes exceeded the number of installable axes as a result of axis shift with a base command.
C6E	Servo-OFF axis use error	An attempt was made to use an axis whose servo is OFF.
C6F	Home-return incomplete error	Home return has not completed yet. This error may also occur if operation is performed immediately after changing an encoder parameter, performing an absolute encoder reset or resetting an encoder error, without first executing a software reset or reconnecting the power.
C70	Absolute coordinate non-confirmation error	Absolute coordinates have not been confirmed. The power must be reconnected. This error may also occur if operation is performed immediately after changing an encoder parameter, performing an absolute encoder reset or resetting an encoder error, without first executing a software reset or reconnecting the power.
C71	Synchro slave-axis command error	A command was issued to the synchro slave axis.
C72	Overrun error	The overrun sensor was actuated.
C73	Target-locus soft limit over error	The target position or movement locus exceeds a soft limit. * In the case of a SCARA specification, position data may not exist for the applicable axis.
C74	Actual-position soft limit over error	The actual position exceeds a soft limit by the “soft limit/actual position margin” or more.
C75	Motion-data-packet generation logic error	The motion-data-packet generation logic is invalid.
C76	Movement-point count over error	Too many packets are generated simultaneously.
C77	Handling-packet overflow error	The servo handling packets overflowed.
C78	Motion-data-packet overflow error	The servo motion data packets overflowed.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
C79	Pole sense operation error	Operation is disabled in the pole sense mode.
C7A	Servo unsupported function error	An attempt was made to use an unsupported function.
C7B	Odd-pulse slide error	Internal servo calculation error
C7C	Odd-pulse processing logic error	Internal servo calculation error
C7D	Packet pulse shortage error	Internal servo calculation error
C7E	Quadratic equation solution error	An error was detected while calculating a quadratic equation solution.
C7F	No valid specified axis error	No valid axes are specified.
C80	Servo-packet calculation logic error	Internal servo calculation error If the controller is of absolute encoder specification and the system has just been moved or “Error No. C74, Actual-position soft limit over error” has also generated, the controller may be experiencing a servo-packet calculation overflow caused by abnormal current position resulting from an unsuccessful absolute reset. Perform an absolute reset again by following the operation manual. (Simply selecting “Encoder error reset” on the absolute reset screen will not allow the controller to recognize the correct position. Always perform an absolute reset by strictly following the specified procedure.)
C81	Operation-amount logic during servo ON	Servo processing logic error
C82	Servo direct command type error	Servo processing logic error
C83	Servo calculation method type error	The servo calculation method type is invalid.
C84	In-use axis servo OFF error	The servo of an axis currently in use (being processed) was turned off.
C85	Non-installed driver error	Driver is not installed for the applicable axis.
C86	Driver servo ready OFF error	The ready signal for the driver of the applicable axis is OFF.
C87	SEL unsupported function error	An attempt was made to use a function not supported by SEL.
C88	Speed specification error	The specified speed is invalid.
C89	Acceleration/deceleration specification error	The specified acceleration/deceleration is invalid.
C8B	Circle/arc calculation logic error	The arc calculation logic is invalid.
C8D	Circle/arc calculation error	Position data that cannot be used in arc movement was specified. Check the position data.
C8E	Point deletion error during command execution	The final point data was deleted while continuous point movement was being calculated.
C8F	Axis operation type error	The axis operation type is invalid. Check “Axis-specific parameter No. 1, Axis operation type” and perform operation appropriate for the operation type specified.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
C90	Spline calculation logic error	The spline processing logic is invalid.
C91	Push-motion axis multiple specification error	Two or more push-motion axes were specified.
C92	Push-motion approach distance/speed specification error	The specified push-motion approach distance/speed is invalid.
C93	System output operation error	The user attempted a system output operation (through the port specified by I/O parameter for output function selection or the zone output port specified by axis-specific parameter).
C94	PIO program number error	The PIO-specified program number is invalid.
C95	AUTO program number error	The setting of “Other parameter No. 1, Auto-start program number” is invalid.
C96	Start error from operation-abort program	(This error should not occur now that the specification has been changed.)
C97	Program number error for I/O processing program at operation/program abort	The setting of “Other parameter No. 2, I/O processing program number at operation/program abort” is invalid.
C98	Program number error for I/O processing program at operation pause	The setting of “Other parameter No. 3, I/O processing program number at all operation pause” is invalid.
C99	Home sensor non-detection error	The home sensor cannot be detected. Check the wiring and sensor.
C9A	Creep sensor non-detection error	The creep sensor cannot be detected. Check the wiring and sensor.
C9B	Phase Z non-detection error	Phase Z cannot be detected. Check the wiring and encoder.
C9C	Defective phase-Z position error	The phase-Z position is defective. Normal wear and tear of the mechanical ends and home sensor may also be a reason. Readjustment is necessary.
C9D	Card parameter write error	Error writing card parameters
C9E	Servo calculation overflow error	Internal servo calculation error
CA1	Abnormal absolute-data backup battery voltage (Driver analysis)	Check the connection of the absolute-data backup battery/replace the battery and/or check the encoder cable connection, and then perform an absolute reset.
CA2	Abnormal absolute-data backup battery voltage (Main analysis)	Check the connection of the absolute-data backup battery/replace the battery and/or check the encoder cable connection, and then perform an absolute reset.
CA3	Slave setting data out-of-range error	The data set to the slave is outside the allowable range.
CA4	Slave error response	An error response was returned from the slave.
CA5	Stop deviation overflow error	Movement may have occurred during stopping due to external force or operation may have been restricted during deceleration. This error may also generate when jog operation is restricted (due to contact with an obstacle, contact with a mechanical end before home return, etc.) or when wiring error, faulty encoder or faulty motor is detected during deceleration.
CA6	Palletizing number error	The specified palletizing number is invalid.
CA7	Setting error of even-numbered row count for palletizing zigzag	The set even-numbered row count for palletizing zigzag is invalid.
CA8	Setting error of palletizing pitches	The set palletizing pitches are abnormal.
CA9	Setting error of placement points in palletizing-axis directions	The set X/Y-axis direction counts for palletizing are invalid.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
CAA	Palletizing PASE/PAPS non-declaration error	Neither PASE nor PAPS palletizing-setting command is set. Set either command.
CAB	Palletizing position number error	The specified palletizing position number is invalid.
CAC	Palletizing position number setting over	The specified palletizing position number exceeds the position number range calculated for the current palletizing setting.
CAD	Palletizing PX/PY/PZ-axis duplication error	Any two of the specified PX, PY and PZ-axes for palletizing are the same axis.
CAE	Insufficient valid axes for palletizing 3-point teaching data	There are not enough valid axes in the point data for palletizing 3-point teaching. Axes to comprise the palletizing PX/PY planes cannot be specified.
CAF	Excessive valid axes for palletizing 3-point teaching data	There are too many valid axes in the point data for palletizing 3-point teaching. Axes to comprise the palletizing PX/PY planes cannot be specified.
CB0	Mismatched valid axes for palletizing 3-point teaching data	The valid axis pattern in the point data for palletizing 3-point teaching does not match.
CB1	Offset setting error at palletizing 3-point teaching	Zigzag offset (not zero) cannot be set in palletizing 3-point teaching, if the reference point is the same as the end point of the PX-axis.
CB2	BGPA/EDPA pair-end mismatch error	The BGPA/EDPA syntax is invalid. EDPA was declared before BGPA, or another BGPA was declared after BGPA without first declaring EDPA.
CB4	Arch-motion Z-axis non-declaration error	Z-axis has not been declared by PCHZ or ACHZ.
CB5	BGPA non-declaration error during palletizing setting	Palletizing setting cannot be performed without first declaring BGPA. Declare BGPA.
CB6	Palletizing point error	The palletizing points are invalid (non-Z-axis components for arch-motion movement are absent, etc.).
CB7	Arch-trigger non-declaration error	Declare arch triggers using PTRG or ATRG.
CB8	No 3-point teaching setting error at palletizing angle acquisition	The palletizing angle cannot be acquired until setting by palletizing 3-point teaching is complete.
CB9	PX/PY-axis indeterminable error at palletizing angle acquisition	Angle cannot be calculated because there are too many valid axes in the 3-point teaching data and thus PX/PY-axes cannot be specified.
CBA	Reference-axis/PY/PY-axis mismatch error at palletizing angle acquisition	Angle cannot be calculated because the reference axis for angle calculation is neither of the axes comprising the PX/PY-axes as set by 3-point teaching.
CBB	Reference-point/PX-axis end-point duplication error at palletizing angle acquisition	Angle cannot be calculated because the reference point of 3-point teaching is the same as the PX-axis end-point data other than the PZ-axis component and thus arc tangent cannot be calculated.
CBC	Palletizing motion calculation error	Trapezoid control calculation error for palletizing motion
CBD	MOD command divisor 0 error	“0” was specified as the divisor in the MOD command.
CBE	Target-locus boundary over error	The target position or movement locus exceeded the positioning boundary in the infinite-stroke mode.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
CBF	Positioning distance overflow error	The positioning distance is too large. If the controller is of absolute encoder specification and the system has just been moved or “Error No. C74, Actual-position soft limit over error” has also generated, the controller may be experiencing a servo-packet calculation overflow caused by abnormal current position resulting from an unsuccessful absolute reset. Perform an absolute reset again by following the operation manual. (Simply selecting “Encoder error reset” on the absolute reset screen will not allow the controller to recognize the correct position. Always perform an absolute reset by strictly following the specified procedure.)
CC0	Axis mode error	The axis mode is invalid.
CC1	Speed change condition error	An attempt was made to change the speed of an axis whose speed cannot be changed (axis operating in S-motion, etc.).
CC2	Driver parameter list number error	The driver parameter list number is invalid.
CC3	Angle error	The angle is invalid.
CC4	SEL data error	The SEL data is invalid.
CC5	Positioning boundary pull-out error	An attempt was made to execute a command not permitted outside the positioning boundary.
CC6	Driver error primary detection	A driver error was found by primary detection.
CC7	Palletizing movement PZ-axis pattern non-detection error	PZ-axis component is not found in the axis pattern during palletizing movement.
CC8	Arch top Z-axis pattern non-detection error	Z-axis component relating to the highest point of arch motion is not found in the axis pattern during arch motion operation.
CC9	Arch trigger Z-axis pattern non-detection error	Z-axis component relating to arch motion is not found in the axis pattern of the arch-trigger declaration point data.
CCA	Arch top/end-point reversing error	The coordinates of highest point and end point are reversed during arch motion operation.
CCB	Arch start-point/trigger reversing error	The coordinates of start point and start-point arch trigger are reversed during arch motion operation.
CCC	Arch end-point/trigger reversing error	The coordinates of end point and end-point arch trigger are reversed during arch motion operation.
CCD	Drive-source cutoff axis use error	An attempt was made to use an axis whose drive source is cut off.
CCE	Error axis use error	An attempt was made to use an axis currently generating an error.
CCF	Palletizing reference-point/valid-axis mismatch error	The PX/PY(/PZ)-axes set by PASE/PCHZ are not valid in the axis pattern of the reference-point data set by PAST.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
D01	Encoder EEPROM-write timeout error	The encoder is faulty or failure occurred in the encoder communication.
D02	Encoder EEPROM-read timeout error	The encoder is faulty or failure occurred in the encoder communication.
D03	Encoder count error	Faulty encoder or defective encoder assembly condition is suspected.
D04	Encoder one-revolution reset error	The encoder is faulty or has turned.
D05	Encoder-EEPROM write acceptance error	The encoder is faulty or failure occurred in the encoder communication.
D06	Encoder received-data error	The encoder is faulty or failure occurred in the encoder communication.
D07	Driver logic error	The driver CPU board is in a condition where it cannot operate normally.
D08	Encoder CRC error	The encoder is faulty or failure occurred in the encoder communication.
D09	Driver overspeed error	The motor speed exceeded the upper limit.
D0A	Driver overload error	The power input to the motor exceeded the upper limit.
D0B	Driver EEPROM data error	Failure during write or EEPROM failure
D0C	Encoder EEPROM data error	Failure during write or EEPROM failure
D0E	Axis sensor error	An error occurred in the axis sensor.
D0F	Power stage temperature error	The power stage board exceeded the upper temperature limit.
D10	IPM error	A failure occurred in the motor drive circuit.
D11	Driver abnormal interruption error	The driver CPU board is in a condition where it cannot operate normally.
D12	Encoder disconnection error	The encoder cable is disconnected. The power must be reconnected.
D13	FPGA watchdog timer error	Failure in the interface with the main CPU
D14	Current loop underrun error	Failure in the interface with the main CPU
D15	Driver-CPU down status error	An error occurred in the driver CPU board.
D17	Main-CPU alarm status error	Failure in the interface with the main CPU
D18	Speed loop underrun error	Failure in the interface with the main CPU
D19	Encoder receive timeout error	The encoder is faulty or failure occurred in the encoder communication.
D1A	Driver command error	An error occurred in the CPU bus command.
D1B	Serial bus receive error	Failure in the interface with the main CPU
D1C	Encoder overspeed error	The motor speed exceeded the upper limit.
D1D	Encoder full-absolute status error	The motor speed exceeded the upper limit.
D1E	Encoder counter overflow error	The encoder rotation counter exceeded the upper limit.
D1F	Encoder rotation error	Faulty encoder or defective encoder assembly condition is suspected.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
D20	Driver error	(Refer to error No. CA1.)
D22	Encoder rotation reset error	The encoder is faulty or has turned.
D23	Encoder alarm reset error	Faulty encoder
D24	Encoder ID error	The encoder is faulty or failure occurred in the encoder communication.
D25	Encoder configuration mismatch error	The encoder configuration information is outside the function information range.
D26	Motor configuration mismatch error	The motor configuration information is outside the function information range.
D50	Fieldbus error (FBMIRQ timeout)	A FBMIRQ timeout was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D51	Fieldbus error (FBMIRQ reset)	A FBMIRQ reset error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D52	Fieldbus error (FBMBSY)	A FBMBSY was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D53	Fieldbus error (BSYERR)	A BSYERR was detected. The power must be reconnected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D54	Window lock error (LERR)	A LERR was detected. The power must be reconnected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D55	Fieldbus error (Min busy)	A Min busy error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D56	Fieldbus error (MinACK timeout)	A Min ACK timeout was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D57	Fieldbus error (MoutSTB timeout)	A Mout STB timeout was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.

(In the panel window, the three digits after "E" indicate an error number.)

Error No.	Error name	Description, action, etc.
D58	Fieldbus error (INIT timeout)	An INIT timeout was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D59	Fieldbus error (DPRAM write/read)	A DPRAM write/read error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D5A	Fieldbus error (TOGGLE timeout)	A TOGGLE timeout was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D5B	Fieldbus error (Access-privilege retry over)	An access-privilege retry over error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D5C	Fieldbus error (Access-privilege open error)	An access-privilege open error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D5D	Fieldbus error (FBRS link error)	A FBRS link error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D5E	Fieldbus error (Mailbox response)	A mailbox response error was detected. Check the status of the monitor LED on the front face of the board by referring to the operation manual for the field network board.
D67	Motor/encoder configuration information mismatch error	The "motor/encoder configuration information" (motor identification number and encoder identification number) in driver parameter No. 26 does not match the "motor/ encoder configuration information" (motor identification number and encoder identification number) in encoder parameter No. 11. Check the parameter values, encoder cable connection, etc.
D68	No remote-mode control support board error	Hardware supporting remote-mode control is not installed, although remote-mode control (AUTO/MANU) is specified in I/O parameter No. 79.
D69	External terminal block overcurrent or power-supply error	Overcurrent or power-supply error in the external terminal block
D70	Option use permission error	Check if any option whose use is not permitted is specified in the system program.
D6A	Hardware unsupported function error	An attempt was made to use a function not supported by the hardware.
D6B	Overrun error	The overrun sensor was actuated.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
D6C	Actual-position soft limit over error	The actual position exceeded a soft limit by the “soft limit/actual position margin” or more.
D6D	Logic error	A logic error occurred.
D6F	Optional password error	The optional function the controller is attempting to use requires an optional password. Check other parameter Nos. 30 through 32, etc., in accordance with the applicable function.
E01	DMA address error	DMA transfer error
E02	SCIF send-buffer overflow error	The SCIF send buffer overflowed.
E03	SCI send-buffer overflow error	The SCI send buffer overflowed.
E04	SCIF receive-buffer overflow error	The SCIF receive buffer overflowed. Excessive data was received from outside.
E05	SCI receive-buffer overflow error	The SCI receive buffer overflowed. Excessive data was received from the slave.
E06	Receive timeout error (Slave communication)	Response from the slave cannot be recognized.
E07	SCI overrun error (Slave communication)	Communication failure. Check for noise, circuit failure and slave card.
E08	SCI framing error (Slave communication)	Communication failure. Check for noise, shorting, circuit failure and slave card.
E09	SCI parity error (Slave communication)	Communication failure. Check for noise, shorting, circuit failure and slave card.
E0A	SCI CRC error (Slave communication)	The CRC in the message is invalid.
E10	SCIF communication mode error	The communication mode is invalid.
E11	SCI communication mode error	The communication mode is invalid.
E14	SCI receive-data-register full wait timeout error	Communication failure. Check for noise, shorting, circuit failure and slave card.
E15	SCI overrun error	Communication failure. Check for noise, shorting, circuit failure and slave card.
E16	Program end confirmation timeout error	The program cannot be ended.
E17	I/O-processing-program start logic error	The I/O-processing-program start logic is invalid.
E18	Task ID error	The task ID is invalid.
E19	WAIT factor error	The WAIT factor is invalid.
E1A	WAIT logic error	The WAIT logic is invalid.
E1B	Point-data valid address error	Point-data valid address is not set.
E1C	Source data error	The source data is invalid.
E1D	Unaffected output number error	The unaffected output number is invalid. A value other than an output port number (“0” is acceptable) may be input in I/O parameter Nos. 70 to 73.
E1E	Zone parameter error	A value other than an output port/global flag number (“0” is acceptable) or duplicate numbers may be input in axis-specific parameter Nos. 88, 91, 94 and 97, or the output number specified as system output in the I/O parameter for output function selection may be duplicated, among other reasons.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
E1F	I/O assignment parameter error	A value other than an I/O port number (“-1” is acceptable) or other than an I/O head port number + [multiple of 8] may be input in I/O parameter Nos. 2 to 9, or a value other than a [multiple of 8] may be input in I/O parameter Nos. 14 to 17.
E20	I/O assignment duplication error	I/O assignments are duplicated. Check I/O parameter Nos. 2 to 9 and 14 to 17 and the I/O slot card type (number of I/Os), etc.
E21	I/O assignment count over error	The I/O assignments exceed the specified range. Check I/O parameter Nos. 2 to 9 and 14 to 17 and the I/O slot card type (number of I/Os).
E22	Header error (Slave communication)	The header in the message received from the slave card is invalid.
E23	Card ID error (Slave communication)	The card ID in the message received from the slave card is invalid.
E24	Response type error (Slave communication)	The response type in the message received from the slave card is invalid.
E25	Command type error (Slave communication)	The command type of the transmitting command is invalid.
E26	Target type error	The target type is invalid.
E27	No target error	Target (driver card, I/O card, encoder or other slave card) is not installed.
E29	EEPROM error (EWEN/EWDS not permitted)	EEPROM access error (when writing)
E2A	Read compare mismatch error during EEPROM write	EEPROM access error (when writing)
E2B	Abnormal response error when sending EEPROM information acquisition command	An abnormal response was received when a slave-EEPROM information acquisition command was sent.
E2C	Maximum receive size over error when sending EEPROM information acquisition command	The maximum receive size exceeds the limit value when a slave-EEPROM information acquisition command is sent.
E2D	Receive-data checksum error when sending EEPROM information acquisition command	The checksum of receive data is invalid when a slave-EEPROM information acquisition command is sent.
E33	Slave response logic error	The slave response logic is invalid.
E34	Slave block number out of range	The slave block number is out of range.
E37	Slave data setting prohibited	Setting of slave data is prohibited.
E38	Faulty slave EEPROM	The slave EEPROM is faulty.
E39	No encoder EEPROM error	The encoder is not equipped with EEPROM.
E3A	Absolute encoder error	Absolute encoder is specified illegally.
E3C	Undefined slave-command error code detected	An undefined slave-command error code was detected.
E3D	SEL program/point/parameter flash ROM status error	Data is not written to the flash ROM correctly or written in an old, incompatible application version.
E3E	Parameter checksum error	The flash ROM data has been destroyed.
E3F	Gain parameter error	The setting of “Axis-specific parameter No. 60, Position gain,” etc., is invalid.
E40	Rotational-movement axis parameter error	Check axis-specific parameter Nos. 67, 66, 38, 37, 1, etc.
E41	Servo-motion data packet shortage error	There are not enough servo-motion data packets.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
E42	Servo job error	The servo job is invalid.
E45	Servo undefined command detection error	An undefined command was detected during servo processing.
E46	Maximum receive size over error at absolute-data acquisition	The receive size is too large when acquiring absolute data.
E47	No normal response error at absolute-data acquisition	Normal response is not received when acquiring absolute data.
E49	Encoder rotation error	An encoder rotation error was detected.
E4A	Encoder rotation counter overflow error	An encoder rotation counter overflow error was detected.
E4B	Encoder count error	An encoder count error was detected.
E4C	Encoder overspeed error	An encoder overspeed error was detected.
E4D	Driver phase-Z detection logic error	A phase-Z detection completion status was notified from the driver in a mode other than the phase-Z detection operation mode.
E4E	Phase-Z count parameter error	Check axis-specific parameter Nos. 23, 38, 37, etc.
E4F	Synchro parameter error	Check axis-specific parameter Nos. 65, 39, all-axis parameter No. 1, etc.
E50	Driver special command ACK-timeout error	ACK cannot be detected for the driver special command.
E51	Drive unit error (DRVESR)	Error notification from the driver
E52	Encoder error (DRVESR)	Error notification from the driver
E53	Driver CPU error (DRVESR)	Error notification from the driver
E54	Servo control error (DRVESR)	Error notification from the driver
E55	Command error (DRVESR)	Error notification from the driver
E56	Motor temperature error (DRVESR)	Error notification from the driver
E58	Servo ON/OFF timeout error	Servo ON/OFF cannot be confirmed.
E59	Brake ON/OFF timeout error	Brake ON/OFF cannot be confirmed.
E5A	Pole sense non-detection error	Motor magnetic pole cannot be detected.
E5B	Detection OFF error upon pole sense completion	The motor-magnetic-pole detection status bit (Psenex) is turned OFF after completion of pole sense.
E5C	Hold-at-stop servo job error	The servo job is invalid.
E5D	Servo packet error	The servo packets are invalid.
E5E	Servo-control-right management array number error	The servo-control-right management array number is invalid.
E5F	Length conversion parameter error	Check axis-specific parameter Nos. 47, 50, 51, 42, 1, etc.
E60	Slave maximum receive size over error	The slave receive size is too large.
E61	Slave no normal response reception error	Normal response cannot be received from the slave.
E62	Sending-slave CPU type error	The CPU type of the sending slave is invalid.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
E63	Message-buffer information type error	The message-buffer information type is invalid.
E64	Abnormal standby power detection error	Abnormal standby power was detected.
E65	Regenerative resistance temperature error	A regenerative resistance temperature error was detected.
E66	AC-power overvoltage error	An AC-power overvoltage error was detected.
E67	Motor-power overvoltage error	A motor-power overvoltage error was detected.
E68	Emergency-stop status requiring reset recovery (not error)	Reset the emergency stop and then reconnect the power.
E69	Abnormal 24-V I/O power source	The 24-V I/O power source is abnormal.
E6A	Safety-gate open status requiring reset recovery (not error)	Close the safety gate and then reconnect the power.
E6B	Shutdown factor indeterminable error	Shutdown factor cannot be determined.
E6C	DO output current error	The DO output current is abnormal.
E6D	Drive-source cutoff relay error	The drive-source cutoff relay may have been melted.
E71	Encoder configuration information outside supported function information range	An encoder whose configuration information is outside the range supported by the driver unit is installed.
E72	Motor configuration information outside supported function information range	A motor whose configuration information is outside the range supported by the driver unit is installed.
E73	Encoder resolution mismatch error	The encoder resolution in the system’s axis-specific parameter and that of the installed encoder do not match.
E74	Encoder division ratio mismatch error	The encoder division ratio in the system’s axis-specific parameter and that of the installed encoder do not match.
E75	Encoder linear/rotary type mismatch error	The encoder linear/rotary type in the system’s axis-specific parameter and that of the installed encoder do not match.
E76	Encoder ABS/INC type mismatch error	The encoder ABS/INC type in the system’s axis-specific parameter and that of the installed encoder do not match.
E77	Magnetic-pole sensor installation specification mismatch error	The magnetic-sensor installation specification in the system’s axis-specific parameter and that of the installed encoder do not match.
E78	Brake installation specification mismatch error	The brake installation specification in the system’s axis-specific parameter and that of the installed encoder do not match.
E79	Abnormal response error when sending EEPROM-data setting slave command	An abnormal response was received when an EEPROM-data setting slave command was sent.
E7A	Maximum receive size over error when sending EEPROM-data setting slave command	The receive size exceeded the limit value when an EEPROM-data setting slave command was sent.
E7B	Motor-drive power ON timeout error	Abnormal current flow from the motor-drive power source
E7C	Register read/write test error	Error reading/writing the register
E7D	Linear-movement axis parameter error	Check axis-specific parameter Nos. 38, 68, 1, etc.
E7E	Parameter error	The parameter is invalid.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
E7F	Stroke parameter error	Check axis-specific parameter Nos. 7, 8, 1, etc.
E80	Unsupported card error	An unsupported card is installed in an I/O slot.
E81	Priority auto-assignment card non-detection error	Priority auto-assignment card cannot be detected.
E82	Card mismatch error	The combination or positioning of I/O slot cards has a problem.
E83	I/O slot card error	The I/O slot card is invalid.
E84	Resolution parameter error	Check axis-specific parameter Nos. 47, 50, 51, 44, 42, 43, 1, 37, etc.
E85	Driver ready OFF factor indeterminable error	Driver ready OFF factor cannot be determined.
E86	Fieldbus error (FBVCCER)	A fieldbus error (FBVCCER) was detected.
E87	Fieldbus error (FBPOWER)	A fieldbus error (FBPOWER) was detected.
E88	Power error (Other)	A power error (Other) was detected. This error also generates when the power OFF → ON interval is short. After the power has been turned off, be sure to wait for at least 5 seconds before turning it back on. Abnormal regenerative resistance temperature is also suspected.
E89	SCIF open error in non-AUTO mode (Servo in use)	In a mode other than AUTO, opening of the serial 1 channel (also used by the PC software/TP port) from a SEL program is prohibited while the servo is in use (to ensure safety).
E8A	SEL program flash-ROM status error	Data is not written to the flash ROM correctly or written in an old, incompatible application version.
E8B	Symbol definition table flash-ROM status error	Data is not written to the flash ROM correctly or written in an old, incompatible application version.
E8C	Point data flash-ROM status error	Data is not written to the flash ROM correctly or written in an old, incompatible application version.
E8D	Parameter flash-ROM status error	Data is not written to the flash ROM correctly or written in an old, incompatible application version.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
FF0 ~ F00	Shutdown error (hi_sysdwn () definition)	A shutdown error (hi_sysdwn () definition) was detected.
F03 ~ F58	Shutdown error (OS call error)	A shutdown error (OS call error) was detected.
F60	System-down level error-call procedure error	A system-down level error-call procedure error was detected.
F61	Interpreter-task end task ID error	An interpreter-task end task ID error was detected.
F62	Abnormal standby power detection error	Abnormal standby power was detected.
F63	Regenerative resistance temperature error	A regenerative resistance temperature error was detected.
F64	AC-power overvoltage error	An AC-power overvoltage error was detected.
F65	Motor-power overvoltage error	A motor-power overvoltage error was detected.
F66	Servo control underrun error	A servo control underrun error was detected.
F67	FROM-write bus width error	A write operation other than 32-bit long word access was detected while writing the flash ROM.
F68	FROM write protect error	Write operation to a write-protected flash ROM area (FRMWE bit in DEVCTR = 1) was detected.
F69	Boot watchdog error	A FPGA boot watchdog was detected. The core program may not be running properly.
F6A ~ FA0	Undefined exception/interruption error	An undefined exception/interruption occurred.
FB0	TMU0 interruption error	A TMU0 interruption error was detected.
FB1	Application code SDRAM copy error (Checksum)	The sum of 4 bytes does not match between the corresponding sections after FROM → SDRAM program copy.
FB2	Installed flash ROM type mismatch (Application)	The flash ROM type anticipated in the software does not match the flash ROM type actually installed. Check the combination of software and hardware.
FB8	Undefined NMI error	An undefined NMI interruption occurred.

◎ Error List (MAIN core) (In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A70	SCIF overrun error	Communication error. Check for noise, connected equipment and communication setting. (When updating the application, connect to a PC and use IAI’s update tool.)
A71	SCIF framing error	Communication error. Check for noise, shorted/disconnected communication cable, connected equipment and communication setting. (When updating the application, connect to a PC and use IAI’s update tool.)
A72	SCIF parity error	Communication error. Check for noise, shorted/disconnected communication cable, connected equipment and communication setting. (When updating the application, connect to a PC and use IAI’s update tool.)
A73	IAI protocol header error	Communication protocol error. Check for noise and connected equipment. (When updating the application, connect to a PC and use IAI’s update tool.)
A74	IAI protocol terminal ID error	Communication protocol error. Check for noise and connected equipment. (When updating the application, connect to a PC and use IAI’s update tool.)
A75	IAI protocol command ID error	Communication protocol error. Check for noise and connected equipment. (When updating the application, connect to a PC and use IAI’s update tool.)
A76	IAI protocol checksum error	Communication protocol error. Check for noise and connected equipment. (When updating the application, connect to a PC and use IAI’s update tool.)
A77	Motorola S record type error	The update program file is invalid. Check the file.
A78	Motorola S checksum error	The update program file is invalid. Check the file.
A79	Motorola S load address error	The update program file is invalid. Check the file.
A7A	Motorola S write address over error	The update program file is invalid. Check the file.
A7B	Flash timing limit over error (Write)	Error writing the flash ROM (When updating)
A7C	Flash timing limit over error (Erase)	Error erasing the flash ROM (When updating)
A7D	Flash verify error	Error erasing/writing the flash ROM (When updating)
A7E	Flash ACK timeout	Error erasing/writing the flash ROM (When updating)
A7F	Head sector number specification error	Error erasing the flash ROM (When updating)
A80	Sector count specification error	Error erasing the flash ROM (When updating)
A81	Write-destination offset address error (Odd-numbered address)	The address written during flash ROM write (when updating) is invalid. Check the update program file.
A82	Write-source data buffer address error (Odd-numbered address)	Error writing the flash ROM (When updating)
A83	Invalid code sector block ID error	The flash ROM is new, or the program currently written to the flash ROM is invalid because the last update was aborted. The ROM can be updated without problem.
A84	Code sector block ID erase count over	The number of times the flash ROM was erased exceeded the allowable count.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
A85	FROM write request error before erase is complete	When updating, a flash-ROM write command was received before a flash-ROM erase command. Confirm that the update program file is valid and then perform update again.
A86	Absolute-encoder backup battery voltage-low warning (Driver detection)	The voltage of the absolute-data backup battery is low. Check the battery connection or replace the battery.
A87	Motorola S-byte count error (Core detection)	The update program file is invalid. Check the file.
A88	Message conversion error (Core detection)	The received message does not conform to the message format or contains invalid data. Check the message sent from the host communication device.
A89	Updating target non-specification error (Core detection)	During update, an update command was received before the updating target was specified properly. Check if an appropriate updating PC tool is used and the target specification and other settings in the updating PC tool are correct.
A8A	Updating system code error (Core detection)	The system code in the message received with the updating target specification command does not match the controller system. Check the target specification and other settings in the updating PC tool.
A8B	Updating unit code error (Core detection)	The unit code in the message received with the updating target specification command does not match any updatable unit in the controller. Check the target specification and other settings in the updating PC tool.
A8C	Updating device number error (Core detection)	The specified device number in the message received with the updating target specification command is not appropriate. Check the target specification and other settings in the updating PC tool.
A8D	Flash busy reset timeout (Core detection)	Error erasing/writing the flash ROM
A8E	Unit type error (Core detection)	The unit type specified in the message received with the command is invalid or not supported.
CD0	Drive error (Driver detection)	Error notification from the driver
CD1	Encoder error (Driver detection)	Error notification from the driver
CD2	Driver CPU error (Driver detection)	Error notification from the driver
CD3	Servo control error (Driver detection)	Error notification from the driver
CD4	Command error (Driver detection)	Error notification from the driver
CD5	Motor temperature error (Driver detection)	Error notification from the driver

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
E90	Core code flash-ROM status error	The core program is invalid. Contact the manufacturer.
E91	Application code flash-ROM status error	The application program is invalid. Contact the manufacturer.
E92	Core code sum error	The core program is invalid. Contact the manufacturer.
E93	Application code sum error	The application program is invalid. Contact the manufacturer.
E94	Timing limit over error (Flash erase)	Error erasing the flash ROM
E95	Flash verify error (Flash erase)	Error erasing the flash ROM
E96	Flash ACK timeout (Flash erase)	Error erasing the flash ROM
E97	Head sector number specification error (Flash erase)	Error erasing the flash ROM
E98	Sector count specification error (Flash erase)	Error erasing the flash ROM
E99	Timing limit over error (Flash write)	Error writing the flash ROM
E9A	Flash verify error (Flash write)	Error writing the flash ROM
E9B	Flash ACK timeout (Flash write)	Error writing the flash ROM
E9C	Write-destination offset address error (Flash write)	Error writing the flash ROM
E9D	Write-source data buffer address error (Flash write)	Error writing the flash ROM
E9E	Watchdog reset occurrence error	A WDT (watchdog timer) was manually reset (error detection).
E9F	Exception occurrence error while BL = 1 (NMI)	An exception occurred while the block bit in the CPU status register was “1.” (NMI)
EA0	Exception occurrence error while BL = 1 (Other than NMI)	An exception occurred while the block bit in the CPU status register was “1.” (Other than NMI)
EA1	Bit exception reset due to command/data TLB duplication	This reset occurs when there are multiple TLB entries corresponding to the virtual address.
EA2	Undefined exception/interruption error	An undefined exception/interruption occurred.
EA3	AC-power cutoff detection error	An AC-power cutoff was detected.
EA4	Abnormal standby power detection error	Abnormal standby power was detected.
EA5	Regenerative resistance temperature error	A regenerative resistance temperature error was detected.
EA6	AC-power overvoltage error	An AC-power overvoltage error was detected.
EA7	Motor-power overvoltage error	A motor-power overvoltage error was detected.
EA8	FROM-write bus width error	A write operation other than 32-bit long word access was detected while writing the flash ROM.
EA9	FROM write protect error	Write operation to a write-protected flash ROM area (FRMWE bit in DEVCTR = 1) was detected.
EAA	SDRAM write/read test error	The SDRAM is faulty. Contact the manufacturer.
EAB	Application-update SCIF send-queue overflow error	An overflow occurred in the send queue.

(In the panel window, the three digits after “E” indicate an error number.)

Error No.	Error name	Description, action, etc.
EAC	Servo control underrun error	A servo control underrun error was detected.
EAD	Boot error	A FPGA boot watchdog was detected. The core program may not be running properly.
EAE	Application-update SCIF receive-queue overflow error	Excessive data is received from outside. (Confirm that a PC and IAI’s update tool are used to update the application.)
EAF	Installed flash ROM type mismatch (Core)	The flash ROM type anticipated in the software does not match the flash ROM type actually installed. Check the combination of software and hardware.
EB0	Undefined NMI error (Core)	An undefined NMI interruption occurred.
EB1	FPGA read/write test error (Core)	A read/write error of the FPGA.
EB2	Flash busy reset timeout (Core detection)	Flash ROM malfunction. The busy status of the flash ROM is not reset.

◎ Troubleshooting of ASEL Controller

After the optional panel unit was connected, the panel window began displaying an error number every time an error generates.
When the power is turned on, normally “rdy” or “Ardy” will be displayed. “P01” or other code will be displayed while a program is running.
When an error generates, the panel window will show “EA1D” or other code starting with “E.” (Some errors do not begin with “E.”)

Status	Panel window display
After turning on the power	rdy, Ardy
Program is running	P01, P64, etc.
Error has generated	EA1D, ED03, etc.

* Among the alphabets, B and D are shown in lower case.

Depending on the error number, it may be possible to reset the error after removing the cause of the error, or the power must be reconnected to reset the error.
Also, some error numbers are output to the LED display in the panel window, while others are not.
For details, see “◎ Error Level Control.”

Troubleshooting (Causes and Countermeasures for Key Errors)

Error No.	Error name	Cause	Countermeasure
dCF	DC power cutoff	Momentary power failure has occurred or the voltage has dropped.	Check the power-source voltage. (24-VDC specification)
ErG	Emergency stop (This is not an error.)	Emergency-stop signal is input.	Emergency-stop signal is input in the following condition: 1. The emergency-stop button on the teaching pendant is pressed. 2. The applicable input terminal in the system connector is turned ON. 3. The port switch on the front panel is set to the manual side. (The teaching-pendant/PC-software connector is not connected.) 4. The actuator is of sensor specification and the slider is stopped on either end of the slider.
enb	Safety gate open	The safety gate is open.	Check the system connector wiring.
C9C	Defective phase-Z position error	The phase-Z position is defective or the reversing amount at home return is small.	Check to see if foreign object has entered the actuator. Check to see if the mounting bolts are contacting the slider. * Change axis-specific parameter No. 22 to “100.”
914 CA2	Abnormal absolute-data backup battery voltage	The PG cable was disconnected from the controller. Absolute reset has not been executed after the initial setup. The voltage of the absolute-data backup battery has dropped.	Connect the PG cable to the controller and execute an absolute reset. Replace the absolute-data backup battery and execute an absolute reset.
CA5	Stop deviation overflow error	Operation is mechanically disabled. If there is no problem in the mechanical function, the power stage board is faulty.	Check to see if the actuator mounting bolts are contacting inside the axes, or if the slider attachment is contacting any surrounding mechanical parts. Replace the board.
C6b	Deviation overflow error	Operation is mechanically disabled.	Check to see if the actuator mounting bolts are contacting inside the axes, or if the slider attachment is contacting any surrounding mechanical parts.

Error No.	Error name	Cause	Countermeasure
d03	Faulty encoder or attachment of dust	The encoder is faulty or dust is attached.	Remove the motor cover and apply cleaning air spray for OA equipment, etc., over the cord wheel. If the problem persists, replace/readjust the encoder.
d06	Encoder received-data error	The encoder cable is disconnected.	Replace the encoder cable.
690	Motor overcurrent error	The motor coil is damaged.	Measure the inter-phase resistances among U, V and W. If the measured resistances are not the same, burn damage is suspected. Replace the motor. If the measured resistances are roughly the same, there is no burn damage.
		If the motor coil is not damaged, the driver's CPU board (the board to which the motor drive cable is connected) is faulty.	Replace the board.
d19	Encoder receive timeout error	The encoder cable is disconnected.	Replace the encoder cable.
d18	Speed loop underrun error	The driver CPU board was damaged due to noise in the encoder cable.	Replace the board and implement noise control measures.
807	Shutdown relay ER status	The transistor on the power-supply board (to which the power cable is connected) is damaged.	Replace the board.

Trouble Report Sheet

Trouble Report Sheet				Date:	
Company name		Department		Reported by	
TEL	(Ext)	FAX			
IAI agent		Purchase date			
Serial number		Manufacture date			

[1] Number of axes ☐ axis(es)

Type

[2] Type of problem

1. Disabled operation 2. Position deviation 3. Runaway machine
4. Error Error code = ☐☐☐
5. Other ( )

[3] Problem frequency and condition

Frequency =

Condition ____

____

____

____

____

[4] When did the problem occur?

1. Right after the system was set up
2. After operating for a while (Operating hours: ________ year(s) and ________ month(s))

[5] Operating direction

1. Horizontal 2. Horizontal + Vertical

[6] Load condition

1. Load transfer 2. Push-motion operation 3. Load: Approx. ________ kg
4. Speed: Approx. _________ mm/sec

[7] Special specification (option, etc.)

Change History

Revision Date	Description of Revision
	First edition
September 2007	Second edition
August 2008	Third edition
June 2010	Fourth edition ▪ Added "Please Read Before Use" on the first page after the cover. ▪ Deleted "Precautions on Safety" before the table of contents and added "Safety Guide" on the first page after the table of contents. ▪ Inserted the attached "Introductory Text on Installation Environment" as the preceding sentence of 2, "Installation Environment" on page 22. ▪ Added "Change History" on the last page. ▪ Updated the back cover to the latest one (address change of the head office and sales offices, 24-hour customer service Eight, etc.)
October 2010	Fifth edition ▪ Warning notes for “Position during servo-on” are added below the servo-on descriptions in pages 58, 161, 162, 301, 303, 305, 311, 315, 322, 325, 330, 335, 343, 354 and 367.
April 2011	Sixth edition ▪ Swapped over the page for CE Marking
February 2013	Seventh edition ▪ “Home return is desired in vertical installation” added to 2. Utilization Examples of Axis-specific Parameters in Appendix Parameter Utilization ▪ Note added in Remarks of Driver Parameter No. 39 Push torque limit at home return ▪ Contents revised in descriptions in I/O Parameters No. 120 to No. 129
July 2013	Eighth edition ▪ Connector of the teaching port changed
March 2015	Edition 8C ▪ Notes related to cancel signal deleted as there is no cancel signal in teaching mode on page 329. ▪ Cancel input deleted as there is no cancel signal in teaching modeon page 336.

Manual No.: ME0165-8C (March 2015)

IAI
Quality and Innovation

IAI Corporation

Head Office: 577-1 Obane Shimizu-KU Shizuoka City Shizuoka 424-0103, Japan
TEL +81-54-364-5105 FAX +81-54-364-2589
website: www.iai-robot.co.jp/

Technical Support available in USA, Europe and China

IAI America, Inc.

Head Office: 2690 W. 237th Street, Torrance, CA 90505
TEL (310) 891-6015 FAX (310) 891-0815
Chicago Office: 110 East State Parkway, Schaumburg, IL 60173
TEL (847) 908-1400 FAX (847) 908-1399
Atlanta Office: 1220 Kennestone Circle, Suite 108, Marietta, GA 30066
TEL (678) 354-9470 FAX (678) 354-9471
website: www.intelligentactuator.com

IAI Industrieroboter GmbH

Ober der Röth 4, D-65824 Schwalbach am Taunus, Germany
TEL 06196-88950 FAX 06196-889524

IAI (Shanghai) Co., Ltd.

SHANGHAI JIAHUA BUSINESS CENTER A8-303, 808, Hongqiao Rd. Shanghai 200030, China
TEL 021-6448-4753 FAX 021-6448-3992
website: www.iai-robot.com

IAI Robot (Thailand) Co., Ltd.

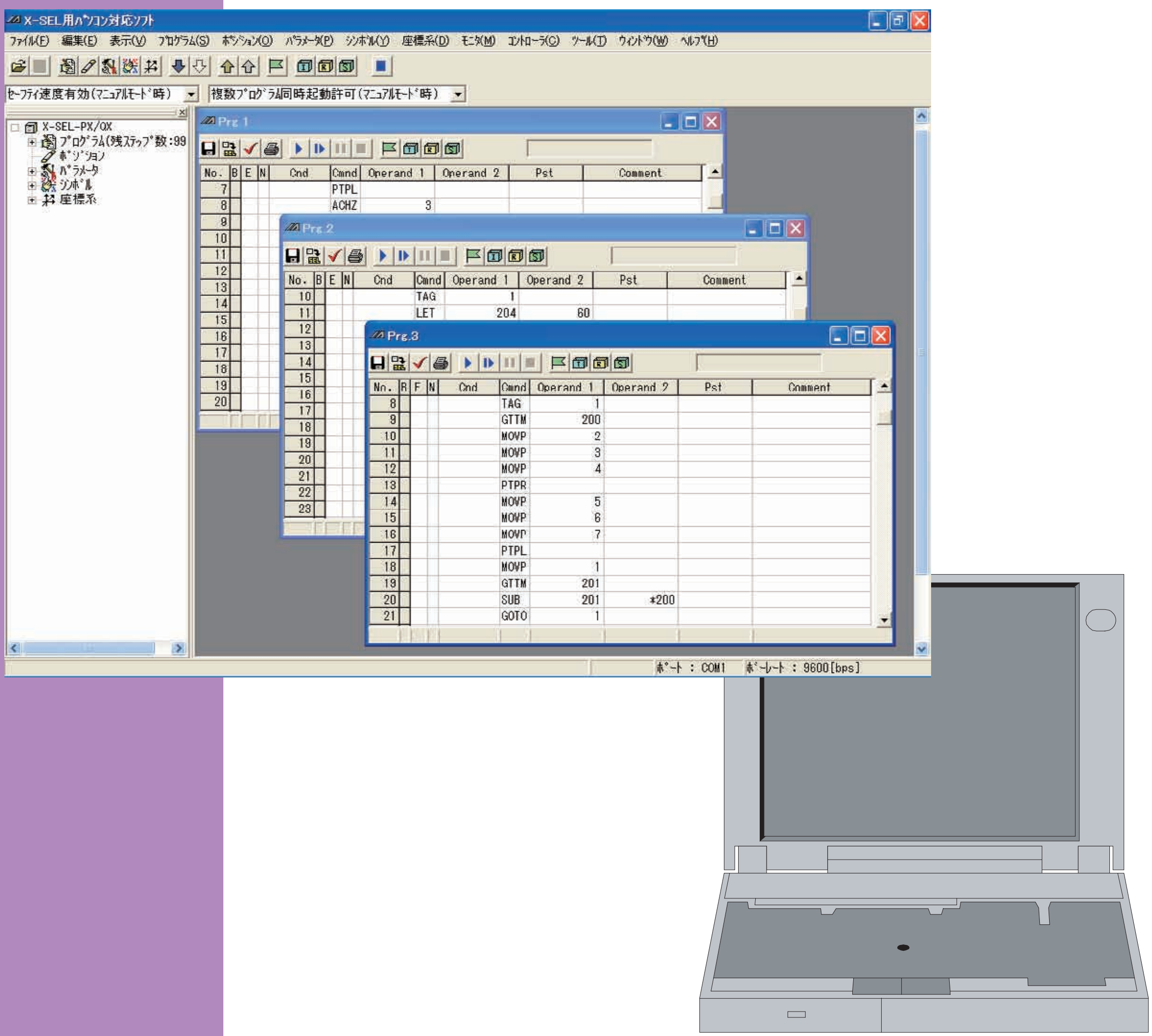
825 PhairojKijja Tower 12th Floor, Bangna-Trad RD., Bangna, Bangna, Bangkok 10260, Thailand
TEL +66-2-361-4458 FAX +66-2-361-4456

15.03.000

INTELLIGENT ACTUATOR

SEL Language Programming Manual

Eighth Edition



IAI America, Inc.

INTELLIGENT ACTUATOR

Please Read Before Use

Thank you for purchasing our product.

This Instruction Manual describes all necessary information items to operate this product safely such as the operation procedure, structure and maintenance procedure.

Before the operation, read this manual carefully and fully understand it to operate this product safely. The enclosed CD/DVD in this product package includes the Instruction Manual for this product.
For the operation of this product, print out the necessary sections in the Instruction Manual or display them using the personal computer.

After reading through this manual, keep this Instruction Manual at hand so that the operator of this product can read it whenever necessary.

[Important]

- This Instruction Manual is original.
- The product cannot be operated in any way unless expressly specified in this Instruction Manual. IAI shall assume no responsibility for the outcome of any operation not specified herein.
- Information contained in this Instruction Manual is subject to change without notice for the purpose of product improvement.
- If you have any question or comment regarding the content of this manual, please contact the IAI sales office near you.
- Using or copying all or part of this Instruction Manual without permission is prohibited.
- The company names, names of products and trademarks of each company shown in the sentences are registered trademarks.

Contents

Table of Contents of Commands in Alphabetical Order

Some commands cannot be used depending on the actuator. For details, refer to individual commands.

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
CIR	Move along circle	○	○	○	○	○	○	○	○	○	○	383
CIR2	Move along circle 2	○	○	○	○	○	○	○	○	○	○	366
CIRS	Move three-dimensionally along circle	○	○	○	○	○	○	○	○	○	○	370
CLOS	Close channel	○	○	○	○	○	○	○	○	○	○	410
CLR	Clear variable	○	○	○	○	○	○	○	○	○	○	233
COS	Cosine	○	○	○	○	○	○	○	○	○	○	240
CP☐☐	Compare number of variable comparisons based on free comparison	○	○	○	○	○	○	○	○	○	○	247
D												
DCL	Set deceleration	○	○	○	○	○	○	○	○	○	○	296
DCLS	Set deceleration ratio for PTP operation			○	○		○			○ (PCX/PGX only)		297
DEG	Set division angle	○	○	○	○	○	○	○	○	○	○	303
DFIF	Define coordinates of simple interference check zone			○	○		○			○ (PCX/PGX only)		336
DFTL	Define tool coordinate system			○	○		○			○ (PCX/PGX only)		320
DFWK	Define load coordinate system			○	○		○			○ (PCX/PGX only)		325
DIS	Set spline division distance	○	○	○	○	○	○	○	○	○	○	312
DIV	Divide	○	○	○	○	○	○	○	○	○	○	237
DW☐☐	Loop [EQ, NE, GT, GE, LT, LE]	○	○	○	○	○	○	○	○	○	○	392
E												
ECMD1	Get motor current value		○		○	○	○	○	○			461
ECMD2	Get home sensor status		○			○	○					462
ECMD3	Get overrun sensor status		○			○	○					463
ECMD4	Get creep sensor status		○			○	○					464
ECMD5	Get axis operation status		○			○	○	○	○	○		465
ECMD6	Current position acquirement on each axis system				○					○ (PCX/PGX only)		466
ECMD20	Get parameter		○			○	○	○	○	○		467
ECMD250	Set torque limit/detection time for torque limit over error		○		○	○	○					469
EDDO	Declare end of DO	○	○	○	○	○	○	○	○	○	○	395
EDIF	Declare end	○	○	○	○	○	○	○	○	○	○	391
EDPA	Declare end of palletizing setting	○	○	○	○	○	○	○	○	○	○	434

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
EDSL	Declare end of SLCT	○	○	○	○	○	○	○	○	○	○	400
EDSR	End subroutine	○	○	○	○	○	○	○	○	○	○	268
ELSE	Declare execution destination when IF command condition is not satisfied	○	○	○	○	○	○	○	○	○	○	390
EOR	Logical exclusive OR	○	○	○	○	○	○	○	○	○	○	246
EXIT	End program	○	○	○	○	○	○	○	○	○	○	269
EXPG	Start program	○	○	○	○	○	○	○	○	○	○	270
EXSR	Execute subroutine	○	○	○	○	○	○	○	○	○	○	266
F												
FMIO	Set IN (B)/OUT (B) command format	○	○	○	○	○	○	○	○	○	○	259
G												
GACC	Get acceleration data	○	○	○	○	○	○	○	○	○	○	289
GARM	Get current arm system			○	○		○			○ (PCX/PGX only)		404
GDCL	Get deceleration data	○	○	○	○	○	○	○	○	○	○	290
GOTO	Jump	○	○	○	○	○	○	○	○	○	○	264
GRP	Set group axes	○	○	○	○	○	○	○	○	○	○	305
GTIF	Get definition coordinates of simple interference check zone			○	○		○			○ (PCX/PGX only)		340
GTTL	Get tool coordinate system definition data			○	○		○			○ (PCX/PGX only)		323
GTAM	Acquirement of target arm system data						○			○ (PCX/PGX only)		287
GTTM	Get time	○	○	○	○	○	○	○	○	○	○	250
GTVD	Image capture command		○			○				○ (PC/PG only)		547
GTWK	Get load coordinate system definition data			○	○		○			○ (PCX/PGX only)		328
GVEL	Get speed data	○	○	○	○	○	○	○	○	○	○	288
H												
HOLD	Declare port to pause	○	○	○	○	○	○	○	○	○	○	306
HOME	Return to home	○	○		○	○	○	○	○	○	○	347
I												
IF□□	Compare [EQ, NE, GT, GE, LT, LE]	○	○	○	○	○	○	○	○	○	○	388
INB	Input BCD (8 digits max.)	○	○	○	○	○	○	○	○	○	○	256
IN	Input binary (32 bits max.)	○	○	○	○	○	○	○	○	○	○	255

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
IS□□	Compare strings	○	○	○	○	○	○	○	○	○	○	389
ITER	Repeat DO	○	○	○	○	○	○	○	○	○	○	394
J												
J□W□	Jog [FN, FF, BN, BF]	○	○		○	○	○	○	○	○	○	359
L												
LEAV	Pull out from DO	○	○	○	○	○	○	○	○	○	○	393
LEFT	Change current arm system to left arm			○	○		○			○ (PCX/PGX only)		331
LET	Assign	○	○	○	○	○	○	○	○	○	○	231
M												
MOD	Calculate remainder	○	○	○	○	○	○	○	○	○	○	238
MOVD	Move by direct value specification							○	○	○ (PCX/PGX only)		356
MOVL	Move to specified position via interpolation	○	○	○	○	○	○	○	○	○	○	350
MOVP	Move to specified position	○	○	○	○	○	○	○	○	○	○	348
MULT	Multiply	○	○	○	○	○	○	○	○	○	○	236
MVDI	Move incrementally by direct value specification							○	○	○ (PC/PG only)		357
MVLI	Move to relative position via interpolation	○	○	○	○	○	○	○	○	○	○	354
MVPI	Move to relative position	○	○	○	○	○	○	○	○	○	○	352
N												
NBND	Set close distance	○										345
NTCH	Anti-Vibration Control Parameter Set Select		○			○	○					549
O												
OFAZ	Set arch-motion Z-axis offset	○	○	○	○	○	○	○	○	○	○	432
OFPZ	Set palletizing Z-axis offset	○	○	○	○	○	○			○	○	447
OFST	Set offset	○	○	○	○	○	○	○	○	○	○	302
OPEN	Open channel	○	○	○	○	○	○	○	○	○	○	409
OR	Logical OR	○	○	○	○	○	○	○	○	○	○	245
OTHE	Declare branching destination when condition is not satisfied	○	○	○	○	○	○	○	○	○	○	399
OTPS	Output current position								○			263
OUT	Output binary (32 bits max.)	○	○	○	○	○	○	○	○	○	○	257
OUTB	Output BCD (8 digits max.)	○	○	○	○	○	○	○	○	○	○	258

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
OUTR	Output relay for ladder	○	○	○	○	○	○	○	○	○	○	152
OVRD	Set speed coefficient	○	○	○	○	○	○	○	○	○	○	293
P												
PACC	Assign position acceleration	○	○	○	○	○	○	○	○	○	○	283
PACH	Palletizing point arch motion	○	○	○	○	○	○			○	○	456
PAPG	Get palletizing calculation data	○	○	○	○	○	○	○	○	○	○	453
PAPI	Set palletizing counts	○	○	○	○	○	○	○	○	○	○	435
PAPN	Set palletizing pattern	○	○	○	○	○	○	○	○	○	○	436
PAPR	Set PUSH command distance, speed	○	○	○	○	○	○	○	○	○	○	314
PAPS	Set palletizing points for 3-point or 4-point teaching	○	○	○	○	○	○	○	○	○	○	440
PAPT	Set palletizing pitches	○	○	○	○	○	○	○	○	○	○	438
PARG	Get palletizing angle	○	○	○	○	○	○	○	○	○	○	452
PASE	Set palletizing axes	○	○	○	○	○	○	○	○	○	○	437
PAST	Set palletizing reference point	○	○	○	○	○	○	○	○	○	○	439
PATH	Move along path	○	○	○	○	○	○	○	○	○	○	358
PAXS	Read axis pattern	○	○	○	○	○	○	○	○	○	○	285
PBND	Set positioning band	○	○	○	○	○	○	○	○	○	○	380
PCHZ	Set palletizing Z-axis	○	○	○	○	○	○			○	○	444
PCLR	Clear position data	○	○	○	○	○	○	○	○	○	○	276
PCPY	Copy position data	○	○	○	○	○	○	○	○	○	○	277
PDCL	Assign position deceleration	○	○	○	○	○	○	○	○	○	○	284
PDEC	Decrement palletizing position number by 1	○	○	○	○	○	○	○	○	○	○	450
PEND	Wait for end of operation of axis using current program	○										387
PEXT	Set palletizing composition	○	○	○	○	○	○			○	○	446
PGET	Assign position to variable 199	○	○	○	○	○	○	○	○	○	○	274
PGST	Get program status	○	○	○	○	○	○	○	○	○	○	402
PINC	Increment palletizing position number by 1	○	○	○	○	○	○	○	○	○	○	449
PMVL	Move to palletizing points via interpolation	○	○		○	○	○	○	○	○ (PC/PG only)	○	455
PMVP	Move to palletizing points via PTP	○	○	○	○	○	○	○	○	○	○	454
POTP	Set PATH output type	○	○	○	○	○	○	○	○	○	○	313
PPUT	Assign value of variable 199	○	○	○	○	○	○	○	○	○	○	275

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
PRDQ	Read current axis position (1 axis direct)	○	○	○	○	○	○	○	○	○	○	279
PRED	Read current axis position	○	○	○	○	○	○	○	○	○	○	278
PSET	Set palletizing position number directly	○	○	○	○	○	○	○	○	○	○	451
PSIZ	Confirm position size	○	○	○	○	○	○	○	○	○	○	286
PSLI	Set zigzag	○	○	○	○	○	○	○	○	○	○	443
PSPL	Move along spline	○	○	○	○	○	○	○	○	○	○	362
PTAM	Substitution of target arm system data						○			○ (PCX/PGX only)		280
PTNG	Get palletizing position number	○	○	○	○	○	○	○	○	○	○	448
PTPD	Specify current arm as PTP target arm system			○	○		○			○ (PCX/PGX only)		334
PTPE	Specify current arm as PTP target arm system			○	○		○			○ (PCX/PGX only)		335
PTPL	Specify left arm as PTP target arm system			○	○		○			○ (PCX/PGX only)		333
PTPR	Specify right arm as PTP target arm system			○	○		○			○ (PCX/PGX only)		332
PTRG	Set palletizing arch triggers	○	○	○	○	○	○			○	○	445
PTRQ	Change push torque limit parameter		○	○	○	○	○	○	○	○	○	365
PTST	Confirm position data	○	○	○	○	○	○	○	○	○	○	281
PUSH	Move by push motion	○	○	○	○	○	○	○	○	○	○	363
PVEL	Assign position speed	○	○	○	○	○	○	○	○	○	○	282
Q												
QRTN	Set quick return mode	○	○			○		○	○	○ (PC/PG only)	○	315
R												
RAXS	Set RC axis pattern		○		○	○	○					486
RCST	Get RC axis status		○		○	○	○					496
READ	Read from channel	○	○	○	○	○	○	○	○	○	○	411
RGAD	Assign RC axis position acceleration/ deceleration to variable 199		○		○	○	○					483
RGIP	Assign RC axis position positioning width to variable 199		○		○	○	○					484
RGTQ	Assign RC axis position current-limiting value to variable 199		○		○	○	○					485
RGVL	Assign RC axis position speed to variable 199		○		○	○	○					482

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
RHOM	Return RC axis to its home		○		○	○	○					489
RIGH	Change right arm of current arm system			○	○		○			○ (PCX/PGX only)		330
RMDI	Incremental move by RC axis direct specification		○		○	○	○					493
RMPI	Incremental move by RC axis position specification		○		○	○	○					491
RMVD	Move by RC axis direct specification		○		○	○	○					492
RMVP	Move by RC axis position specification		○		○	○	○					491
RPAD	Assign variable 199 to RC axis position acceleration/ deceleration		○		○	○	○					479
RPCP	Copy RC axis position data		○		○	○	○					475
RPCR	Clear RC axis position data		○		○	○	○					474
RPGT	Assign RC axis position to variable 199		○		○	○	○					472
RPIP	Assign variable 199 to RC axis position positioning band		○		○	○	○					480
RPPT	Assign variable 199 to RC axis position		○		○	○	○					473
RPRD	Read current RC axis position		○		○	○	○					476
RPRQ	Read current RC axis position (1 axis, direct)		○		○	○	○					477
RPTQ	Assign variable 199 to RC axis position current-limiting value		○		○	○	○					481
RPUS	Move by RC axis push-motion operation		○		○	○	○					494
RPVL	Assign variable 199 to RC axis position speed		○		○	○	○					478
RSOF	Turn RC axis servo OFF		○		○	○	○					488
RSON	Turn RC axis servo ON		○		○	○	○					487
RSPG	Resume program	○	○	○	○	○	○					273
RSTP	Decelerate RC axis to stop		○		○	○	○					495
S												
SCHA	Set end character	○	○	○	○	○	○	○	○	○	○	417
SCMP	Compare character strings	○	○	○	○	○	○	○	○	○	○	419
SCPY	Copy character string	○	○	○	○	○	○	○	○	○	○	418
SCRV	Set sigmoid motion ratio	○	○	○	○	○	○	○	○	○	○	298
SEIF	Specify error type for simple contact check area			○	○		○			○ (PCX/PGX only)		339
SGET	Get character	○	○	○	○	○	○	○	○	○	○	420
SIN	Sine	○	○	○	○	○	○	○	○	○	○	239

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
SLCT	Declare start of multi-branching	○	○	○	○	○	○	○	○	○	○	396
SLEN	Set length	○	○	○	○	○	○	○	○	○	○	426
SLTL	Select tool coordinate system			○	○		○			○ (PCX/PGX only)		322
SLVS	Declare use of Vision System		○			○				○ (PC/PG only)		545
SLWK	Select load coordinate system			○	○		○			○ (PCX/PGX only)		327
SOIF	Specify output for simple interference check zone			○	○		○			○ (PCX/PGX only)		338
SPUT	Set character	○	○	○	○	○	○	○	○	○	○	421
SQR	Root	○	○	○	○	○	○	○	○	○	○	243
SSPG	Pause program	○	○	○	○	○	○	○	○	○	○	272
STOP	Decelerate and stop axis	○	○	○	○	○	○	○	○	○	○	361
STR	Convert character string; decimal	○	○	○	○	○	○	○	○	○	○	422
STRH	Convert character string; hexadecimal	○	○	○	○	○	○	○	○	○	○	423
SUB	Subtract	○	○	○	○	○	○	○	○	○	○	235
SV□□	Servo [ON, OF]	○	○	○	○	○	○	○	○	○	○	346
SYST	Get system status	○	○	○	○	○	○	○	○	○	○	403
T												
TAG	Jump destination	○	○	○	○	○	○	○	○	○	○	265
TAN	Tangent	○	○	○	○	○	○	○	○	○	○	241
TIMC	Cancel waiting	○	○	○	○	○	○	○	○	○	○	249
TIMR	Timer relay for ladder	○	○	○	○	○	○	○	○	○	○	152
TIMW	Wait	○	○	○	○	○	○	○	○	○	○	248
TMLI	Move relatively between positions on tool coordinate system via interpolation			○	○		○			○ (PCX/PGX only)		382
TMPI	Move relatively between positions on tool coordinate system			○	○		○			○ (PCX/PGX only)		381
TMRD	Set read timeout value	○		○							○ (TT only)	413
TMRW	Set read timeout value		○		○	○	○	○	○	○	○ (TTA only)	415
TPCD	Specify processing to be performed when input condition is not specified	○	○	○	○	○	○	○	○	○	○	459
TRAC	Image capturing and tracking command		○		○	○	○					542
TRAN	Copy	○	○	○	○	○	○	○	○	○	○	232

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
TSLP	Task sleep	○	○	○	○	○	○	○	○	○	○	460
TRMD	Declare use of Conveyor Tracking		○		○	○	○					541
V												
VAL	Convert character string data; decimal	○	○	○	○	○	○	○	○	○	○	424
VALH	Convert character string data; hexadecimal	○	○	○	○	○	○	○	○	○	○	425
VEL	Set speed	○	○	○	○	○	○	○	○	○	○	291
VELS	Set speed ratio for PTP operation			○	○		○			○ (PCX/PGX only)		292
VLMX	Specify VLMX speed	○	○		○	○	○	○	○	○	○	311
W												
WGHT	Set tip load mass/inertial moment				○		○			○ (PCX/PGX only)		341
WGT2	Tip load condition setting 2						○					343
WH□□	Branch value [EQ, NE, GT, GE, LT, LE]	○	○	○	○	○	○	○	○	○	○	397
WRIT	Output to channel	○	○	○	○	○	○	○	○	○	○	416
WS□□	Branch character string [EQ, NE]	○	○	○	○	○	○	○	○	○	○	398
WT□□	Wait for I/O, flag [ON, OF]	○	○	○	○	○	○	○	○	○	○	254
WZFA	Wait for zone OFF, with AND	○	○		○	○	○	○	○	○	○	407
WZFO	Wait for zone OFF, with OR	○	○		○	○	○	○	○	○	○	408
WZNA	Wait for zone ON, with AND	○	○		○	○	○	○	○	○	○	405
WZNO	Wait for zone ON, with OR	○	○		○	○	○	○	○	○	○	406
X												
XAST	Acquire extension motion control board axis status		○			○						540
XAXS	Set extension motion control board axis patterns		○			○						514
XCAS	Start synchronizing extension motion control board axis electronic cam (indicating main axis)		○			○						529
XCRP	Clear input counter record for extension motion control board axis		○			○						498
XCTM	Move extension motion control board axis individual electronic cam (indicating time)		○			○						534
XGAC	Read extension motion control board axis acceleration data		○			○						511
XGDC	Read extension motion control board axis deceleration data		○			○						512

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
XGIP	Read extension motion control board axis poaitioning width data		○			○						513
XGTP	Acquire current record of extension motion control board axis input counter		○			○						499
XGVL	Read extension motion control board axis speed data		○			○						510
XHOM	Return extension motion control board axis to home position		○			○						517
XJ☐☐	Perform extension motion control board axis jog operation		○			○						524
XMDI	Move extension motion control board axis to directly indicated relative position		○			○						523
XMLI	Move extension motion control board axis for position relative interpolation		○			○						521
XMPI	Perform extension motion control board axis position relative movement		○			○						519
XMVD	Move extension motion control board axis to directly indicated absolute position		○			○						522
XMVL	Move extension motion control board axis for position indicated interpolation		○			○						520
XMVP	Move extension motion control board axis to indicated position		○			○						518
XPAC	Write extension motion control board axis acceleration data		○			○						507
XPCP	Copy extension motion control board axis position data		○			○						503
XPCR	Erase extension motion control board axis position data		○			○						502
XPDC	Write extension motion control board axis deceleration data		○			○						508
XPED	Waiting for extension motion control board axis to finish positioning operation of axis used by self-program		○			○						525
XPGT	Read extension motion control board axis position data		○			○						500
XPIP	Write extension motion control board axis positioning complete width data		○			○						509
XPPT	Write extension motion control board axis position data		○			○						501

Command	Function	XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
XPRD	Read extension motion control board axis current command position		○			○						504
XPRQ	Read extension motion control board axis current command position (single-axis direct)		○			○						505
XPVL	Write extension motion control board axis speed data		○			○						506
XSFS	Start synchronizing of extension motion control board axis electronic shaft		○			○						536
XSOF	Extension motion control board axis servo OFF		○			○						516
XSON	Extension motion control board axis servo ON		○			○						515
XSTP	Cancel operation of extension motion control board axis		○			○						526
XSYE	Synchronizing of extension motion control board axis completed		○			○						538
XWIP	Waiting for extension motion control board axis positioning complete signal to be turned ON		○			○						527

Table of Contents of Commands by Function

Some commands cannot be used depending on the actuator. For details, refer to individual commands.

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Variable assignment	Clear variable	CLR	○	○	○	○	○	○	○	○	○	○	233
	Assign	LET	○	○	○	○	○	○	○	○	○	○	231
	Copy	TRAN	○	○	○	○	○	○	○	○	○	○	232
Arithmetic operation	Add	ADD	○	○	○	○	○	○	○	○	○	○	234
	Divide	DIV	○	○	○	○	○	○	○	○	○	○	237
	Calculate remainder	MOD	○	○	○	○	○	○	○	○	○	○	238
	Multiply	MULT	○	○	○	○	○	○	○	○	○	○	236
	Subtract	SUB	○	○	○	○	○	○	○	○	○	○	235
Function operation	Inverse tangent	ATN	○	○	○	○	○	○	○	○	○	○	242
	Cosine	COS	○	○	○	○	○	○	○	○	○	○	240
	Sine	SIN	○	○	○	○	○	○	○	○	○	○	239
	Root	SQR	○	○	○	○	○	○	○	○	○	○	243
	Tangent	TAN	○	○	○	○	○	○	○	○	○	○	241
Logical operation	Logical AND	AND	○	○	○	○	○	○	○	○	○	○	244
	Logical exclusive OR	EOR	○	○	○	○	○	○	○	○	○	○	246
	Logical OR	OR	○	○	○	○	○	○	○	○	○	○	245
Comparison	Compare	CP□□	○	○	○	○	○	○	○	○	○	○	247
Timer	Get time	GTTM	○	○	○	○	○	○	○	○	○	○	250
	Cancel waiting	TIMC	○	○	○	○	○	○	○	○	○	○	249
	Wait	TIMW	○	○	○	○	○	○	○	○	○	○	248
I/O, flag operation	Output OFF pulse	BTPF	○	○	○	○	○	○	○	○	○	○	253
	Output ON pulse	BTPN	○	○	○	○	○	○	○	○	○	○	252
	Output, flag [ON, OF, NT]	BT□□	○	○	○	○	○	○	○	○	○	○	251
	Set IN (B)/OUT (B) command format	FMIO	○	○	○	○	○	○	○	○	○	○	259
	Input binary (32 bits max.)	IN	○	○	○	○	○	○	○	○	○	○	255
	Input BCD (8 digits max.)	INB	○	○	○	○	○	○	○	○	○	○	256
	Output current position	OTPS								○			263
	Output binary (32 bits max.)	OUT	○	○	○	○	○	○	○	○	○	○	257
	Output BCD (8 digits max.)	OUTB	○	○	○	○	○	○	○	○	○	○	258
	Wait for I/O, flag [ON, OF]	WT□□	○	○	○	○	○	○	○	○	○	○	254
Program control	Start subroutine	BGSR	○	○	○	○	○	○	○	○	○	○	267
	End subroutine	EDSR	○	○	○	○	○	○	○	○	○	○	268
	Execute subroutine	EXSR	○	○	○	○	○	○	○	○	○	○	266
	Jump	GOTO	○	○	○	○	○	○	○	○	○	○	264
	Declare jump destination	TAG	○	○	○	○	○	○	○	○	○	○	265

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Task management	Stop other program	ABPG	○	○	○	○	○	○	○	○	○	○	271
	End program	EXIT	○	○	○	○	○	○	○	○	○	○	269
	Start program	EXPG	○	○	○	○	○	○	○	○	○	○	270
	Resume program	RSPG	○	○	○	○	○	○	○	○	○	○	273
	Pause program	SSPG	○	○	○	○	○	○	○	○	○	○	272
Position operation	Get acceleration data	GACC	○	○	○	○	○	○	○	○	○	○	289
	Get deceleration data	GDCL	○	○	○	○	○	○	○	○	○	○	290
	Acquirement of target arm system data	GTAM						○			○ (PCX/PGX only)		287
	Get speed data	GVEL	○	○	○	○	○	○	○	○	○	○	288
	Assign position acceleration	PACC	○	○	○	○	○	○	○	○	○	○	283
	Read axis pattern	PAXS	○	○	○	○	○	○	○	○	○	○	285
	Clear position data	PCLR	○	○	○	○	○	○	○	○	○	○	276
	Copy position data	PCPY	○	○	○	○	○	○	○	○	○	○	277
	Assign position deceleration	PDCL	○	○	○	○	○	○	○	○	○	○	284
	Assign position to variable 199	PGET	○	○	○	○	○	○	○	○	○	○	274
	Assign value of variable 199	PPUT	○	○	○	○	○	○	○	○	○	○	275
	Read current axis position (1 axis direct)	PRDQ	○	○	○	○	○	○	○	○	○	○	279
	Read current axis position	PRED	○	○	○	○	○	○	○	○	○	○	278
	Confirm position size	PSIZ	○	○	○	○	○	○	○	○	○	○	286
	Substitution of target arm system data	PTAM						○			○ (PCX/PGX only)		280
	Confirm position data	PTST	○	○	○	○	○	○	○	○	○	○	281
	Assign position speed	PVEL	○	○	○	○	○	○	○	○	○	○	282
Actuator control declaration	Set acceleration	ACC	○	○	○	○	○	○	○	○	○	○	294
	Set acceleration ratio for PTP operation	ACCS			○	○		○			○ (PCX/PGX only)		295
	Indicate ACMX acceleration	ACMX		○			○	○		○			308
	Set reference axis	BASE	○	○		○	○	○	○	○	○	○	304
	Declare port to abort	CANC	○	○	○	○	○	○	○	○	○	○	307
	Set deceleration	DCL	○	○	○	○	○	○	○	○	○	○	296
	Set deceleration ratio for PTP operation	DCLS			○	○		○			○ (PCX/PGX only)		297
	Set division angle	DEG	○	○	○	○	○	○	○	○	○	○	303
	Define coordinate for simple contact check area	DFIF			○	○		○			○ (PCX/PGX only)		336
	Define tool coordinate system	DFTL			○	○		○			○ (PCX/PGX only)		320
	Define load coordinate system	DFWK			○	○		○			○ (PCX/PGX only)		325

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Actuator control declaration	Set spline division distance	DIS	○	○	○	○	○	○	○	○	○	○	312
	Set group axes	GRP	○	○	○	○	○	○	○	○	○	○	305
	Get defined coordinate for simple contact check area	GTIF			○	○		○			○ (PCX/PGX only)		340
	Get defined data of tool coordinate system	GTTL			○	○		○			○ (PCX/PGX only)		323
	Get defined data of load coordinate system	GTWK			○	○		○			○ (PCX/PGX only)		328
	Declare port to pause	HOLD	○	○	○	○	○	○	○	○	○	○	306
	Change left arm of current arm system	LEFT			○	○		○			○ (PCX/PGX only)		331
	Set close distance	NBND	○										345
	Set offset	OFST	○	○	○	○	○	○	○	○	○	○	302
	Set speed coefficient	OVRD	○	○	○	○	○	○	○	○	○	○	293
	Set PUSH command distance, speed	PAPR	○	○	○	○	○	○	○	○	○	○	314
	Set output type	POTP	○	○	○	○	○	○	○	○	○	○	313
	Specify current arm of PTP target arm system	PTPD			○	○		○			○ (PCX/PGX only)		334
	Specify current arm of PTP target arm system	PTPE			○	○		○			○ (PCX/PGX only)		335
	Specify left arm of PTP target arm system	PTPL			○	○		○			○ (PCX/PGX only)		333
	Specify right arm of PTP target arm system	PTPR			○	○		○			○ (PCX/PGX only)		332
	Set quick return mode	QRTN	○	○			○		○	○	○ (PC/PG only)	○	315
	Change right arm of current arm system	RIGH			○	○		○			○ (PCX/PGX only)		330
	Set sigmoid motion ratio	SCRV	○	○	○	○	○	○	○	○	○	○	298
	Specify error type for simple contact check area	SEIF			○	○		○			○ (PCX/PGX only)		339
	Select tool coordinate system	SLTL			○	○		○			○ (PCX/PGX only)		322
	Select load coordinate system	SLWK			○	○		○			○ (PCX/PGX only)		327
	Specify output for simple interference check zone	SOIF			○	○		○			○ (PCX/PGX only)		338
	Set speed	VEL	○	○	○	○	○	○	○	○	○	○	291
	Set speed ratio for PTP operation	VELS			○	○		○			○ (PCX/PGX only)		292
	Specify VLMX speed	VLMX	○	○		○	○	○	○	○	○	○	311

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Actuator control command	Set tip load mass/inertial moment	WGHT				○		○			○ (PCX/PGX only)		341
	Tip load condition setting 2	WGT2						○					343
	Move along arc (ARC2 is recommended)	ARC	○	○	○	○	○	○	○	○	○	○	385
	Move along arc 2 (arc interpolation)	ARC2	○	○	○	○	○	○	○	○	○	○	368
	Move along arc via specification of center position and center angle	ARCC	○	○	○	○	○	○	○	○	○	○	378
	Move along arc via specification of end position and center angle	ARCD	○	○	○	○	○	○	○	○	○	○	376
	Move three-dimensionally along arc	ARCS	○	○	○	○	○	○				○	372
	Change speed	CHVL	○	○		○	○	○	○	○	○	○	374
	Move along circle (CIR2 is recommended)	CIR	○	○	○	○	○	○	○	○	○	○	383
	Move along circle 2 (arc interpolation)	CIR2	○	○	○	○	○	○	○	○	○	○	366
	Move three-dimensionally along circle	CIRS	○	○	○	○	○	○	○	○	○	○	370
	Return to home	HOME	○	○		○	○	○	○	○	○	○	347
	Jog [FN, FF, BN, BF]	J☐W☐	○	○		○	○	○	○	○	○	○	359
	Move by direct value specification	MOVD							○	○	○ (PC/PG only)		356
	Move to specified position via interpolation	MOVL	○	○	○	○	○	○	○	○	○	○	350
	Move to specified position	MOVP	○	○	○	○	○	○	○	○	○	○	348
	Move incrementally by direct value specification	MVDI							○	○	○ (PC/PG only)		357
	Move to relative position via interpolation	MVLI	○	○	○	○	○	○	○	○	○	○	354
	Move to relative position	MVPI	○	○	○	○	○	○	○	○	○	○	352
	Move along path	PATH	○	○	○	○	○	○	○	○	○	○	358
	Set positioning width	PBND	○	○	○	○	○	○	○	○	○	○	380
	Wait for end of operation of axis using current program	PEND	○										387
	Move along spline	PSPL	○	○	○	○	○	○	○	○	○	○	362
	Change push torque limit parameter	PTRQ		○	○	○	○	○	○	○	○	○	365
	Move by push motion	PUSH	○	○	○	○	○	○	○	○	○	○	363
	Decelerate and stop axis	STOP	○	○	○	○	○	○	○	○	○	○	361
	Servo [ON, OF]	SV☐☐	○	○	○	○	○	○	○	○	○	○	346

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Actuator control comman	Move relatively between positions on tool coordinate system via interpolation	TMLI			○	○		○			○ (PC/PG only)		382
	Move relatively between positions on tool coordinate system	TMPI			○	○		○			○ (PC/PG only)		381
Structural IF	Declare end of IF	EDIF	○	○	○	○	○	○	○	○	○	○	391
	Declare execution destination when IF command condition is not satisfied	ELSE	○	○	○	○	○	○	○	○	○	○	390
	Compare [EQ, NE, GT, GE, LT, LE]	IF☐☐	○	○	○	○	○	○	○	○	○	○	388
	Compare strings	IS☐☐	○	○	○	○	○	○	○	○	○	○	389
Structural DO	Loop [EQ, NE, GT, GE, LT, LE]	DW☐☐	○	○	○	○	○	○	○	○	○	○	392
	Declare end of DO	EDDO	○	○	○	○	○	○	○	○	○	○	395
	Repeat DO	ITER	○	○	○	○	○	○	○	○	○	○	394
	Pull out from DO	LEAV	○	○	○	○	○	○	○	○	○	○	393
Multi-branching	Declare end	EDSL	○	○	○	○	○	○	○	○	○	○	400
	Declare branching destination when condition is not satisfied	OTHE	○	○	○	○	○	○	○	○	○	○	399
	Declare start of multi-branching	SLCT	○	○	○	○	○	○	○	○	○	○	396
	Branch value [EQ, NE, GT, GE, LT, LE]	WH☐☐	○	○	○	○	○	○	○	○	○	○	397
	Branch character string [EQ, NE]	WS☐☐	○	○	○	○	○	○	○	○	○	○	398
System information acquisition	Get axis status	AXST	○	○	○	○	○	○	○	○	○	○	401
	Get current arm system	GARM			○	○		○			○ (PCX/PGX only)		404
	Get program status	PGST	○	○	○	○	○	○	○	○	○	○	402
	Get system status	SYST	○	○	○	○	○	○	○	○	○	○	403
Zone	Wait for zone OFF, with AND	WZFA	○	○		○	○	○	○	○	○	○	407
	Wait for zone OFF, with OR	WZFO	○	○		○	○	○	○	○	○	○	408
	Wait for zone ON, with AND	WZNA	○	○		○	○	○	○	○	○	○	405
	Wait for zone ON, with OR	WZNO	○	○		○	○	○	○	○	○	○	406
Communication	Close channel	CLOS	○	○	○	○	○	○	○	○	○	○	410
	Open channel	OPEN	○	○	○	○	○	○	○	○	○	○	409
	Read from channel	READ	○	○	○	○	○	○	○	○	○	○	411
	Set end character	SCHA	○	○	○	○	○	○	○	○	○	○	417
	Set read timeout value	TMRD	○		○							○ (TT only)	413
	Set timeout value	TMRW		○		○	○	○	○	○	○	○ (TTA only)	415
	Output to channel	WRIT	○	○	○	○	○	○	○	○	○	○	416

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
String operation	Compare character strings	SCMP	○	○	○	○	○	○	○	○	○	○	419
	Copy character string	SCPY	○	○	○	○	○	○	○	○	○	○	418
	Get character	SGET	○	○	○	○	○	○	○	○	○	○	420
	Set length	SLEN	○	○	○	○	○	○	○	○	○	○	426
	Set character	SPUT	○	○	○	○	○	○	○	○	○	○	421
	Convert character string; decimal	STR	○	○	○	○	○	○	○	○	○	○	422
	Convert character string; hexadecimal	STRH	○	○	○	○	○	○	○	○	○	○	423
	Convert character string data; decimal	VAL	○	○	○	○	○	○	○	○	○	○	424
	Convert character string data; hexadecimal	VALH	○	○	○	○	○	○	○	○	○	○	425
Arch motion	Set palletizing Z-axis offset	ACHZ	○	○	○	○	○	○	○	○	○	○	429
	Set arch motion composition	AEXT	○	○	○	○	○	○			○	○	431
	Arch motion	ARCH	○	○	○	○	○	○	○	○	○	○	427
	Set arch triggers	ATRG	○	○	○	○	○	○	○	○	○	○	430
	Set arch motion Z-axis offset	OFAZ	○	○	○	○	○	○	○	○	○	○	432
Palletizing definition	Declare start of palletizing setting	BGPA	○	○	○	○	○	○	○	○	○	○	433
	Declare end of palletizing setting	EDPA	○	○	○	○	○	○	○	○	○	○	434
	Set palletizing Z-axis offset	OFPZ	○	○	○	○	○	○			○	○	447
	Set palletizing counts	PAPI	○	○	○	○	○	○	○	○	○	○	435
	Set palletizing pattern	PAPN	○	○	○	○	○	○	○	○	○	○	436
Palletizing definition	Set palletizing points for 3-point or 4-point teaching	PAPS	○	○	○	○	○	○	○	○	○	○	440
	Set palletizing pitches	PAPT	○	○	○	○	○	○	○	○	○	○	438
	Set palletizing axes	PASE	○	○	○	○	○	○	○	○	○	○	437
	Set palletizing reference point	PAST	○	○	○	○	○	○	○	○	○	○	439
	Set palletizing Z-axis	PCHZ	○	○	○	○	○	○			○	○	444
	Set palletizing composition	PEXT	○	○	○	○	○	○			○	○	446
	Set zigzag	PSLI	○	○	○	○	○	○	○	○	○	○	443
	Set palletizing arch triggers	PTRG	○	○	○	○	○	○			○	○	445
Palletizing operation	Get palletizing calculation data	PAPG	○	○	○	○	○	○	○	○	○	○	453
	Get palletizing angle	PARG	○	○	○	○	○	○	○	○	○	○	452
	Decrement palletizing position number by 1	PDEC	○	○	○	○	○	○	○	○	○	○	450
	Increment palletizing position number by 1	PINC	○	○	○	○	○	○	○	○	○	○	449
	Set palletizing position number directly	PSET	○	○	○	○	○	○	○	○	○	○	451
	Get palletizing position number	PTNG	○	○	○	○	○	○	○	○	○	○	448

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Palletizing movement	Palletizing-point arch motion	PACH	○	○	○	○	○	○			○	○	456
	Move to palletizing points via interpolation	PMVL	○	○		○	○	○	○	○	○ (PC/PG only)	○	455
	Move to palletizing points via PTP	PMVP	○	○	○	○	○	○	○	○	○	○	454
Building of pseudo-ladder task	Change task level	CHPR	○	○	○	○	○	○	○	○	○	○	458
	Output relay for ladder	OUTR	○	○	○	○	○	○	○	○	○	○	152
	Timer relay for ladder	TIMR	○	○	○	○	○	○	○	○	○	○	152
	Specify processing to be performed when input condition is not specified	TPCD	○	○	○	○	○	○	○	○	○	○	459
	Task sleep	TSLP	○	○	○	○	○	○	○	○	○	○	460
Extended commands	Get motor current value	ECMD1		○		○	○	○	○	○			461
	Get home sensor status	ECMD2		○			○	○					462
	Get overrun sensor status	ECMD3		○			○	○					463
	Get creep sensor status	ECMD4		○			○	○					464
	Get axis operation status	ECMD5		○			○	○	○	○	○		465
	Current position acquirement on each axis system	ECMD6				○					○ (PCX/PGX only)		466
	Get parameter	ECMD 20		○			○	○	○	○	○		467
	Set torque limit/detection time for torque limit over error	ECMD 250		○		○	○	○					469
Vision System I/F Related	Declare use of Vision System	SLVS		○			○				○ (PC/PG only)		545
	Image Caputure command	GTVD		○			○				○ (PC/PG only)		547
Conveyor Tracking Related	Declare use of Conveyor Tracking	TRMD		○		○	○	○					541
	Image capturing and tracking command	TRAC		○			○	○					542
Anti-Vibration Control Related	Anti-Vibration Control Parameter Set Select	NTCH		○			○	○					549

RC Gateway Function Commands (Controllers with Gateway Function Only)

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
RC axis position operation	Assign RC axis position to variable 199	RPGT		○		○	○	○					472
	Assign variable 199 to RC axis position	RPPT		○		○	○	○					473
	Clear RC axis position data	RPCR		○		○	○	○					474
	Copy RC axis position data	RPCP		○		○	○	○					475
	Read current RC axis position	RPRD		○		○	○	○					476
	Read current RC axis position (1 axis, direct)	RPRQ		○		○	○	○					477
	Assign variable 199 to RC axis position speed	RPVL		○		○	○	○					478
	Assign variable 199 to RC axis position acceleration/ deceleration	RPAD		○		○	○	○					479
	Assign variable 199 to RC axis position positioning width	RPIP		○		○	○	○					480
	Assign variable 199 to RC axis position current-limiting value	RPTQ		○		○	○	○					481
	Assign RC axis position speed to variable 199	RGVL		○		○	○	○					482
	Assign RC axis position acceleration/ deceleration to variable 199	RGAD		○		○	○	○					483
	Assign RC axis position positioning width to variable 199	RGIP		○		○	○	○					484
	Assign RC axis position current-limiting value to variable 199	RGTQ		○		○	○	○					485

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
RC actuator control command	Set RC axis pattern	RAXS		○		○	○	○					486
	Turn RC axis servo ON	RSON		○		○	○	○					487
	Turn RC axis servo OFF	RSOF		○		○	○	○					488
	Return RC axis to its home	RHOM		○		○	○	○					489
	Move by RC axis position specification	RMVP		○		○	○	○					490
	Incremental move by RC axis position specification	RMPI		○		○	○	○					491
	Move by RC axis direct specification	RMVD		○		○	○	○					492
	Incremental move by RC axis direct specification	RMDI		○		○	○	○					493
	Move by RC axis push-motion operation	RPUS		○		○	○	○					494
	Decelerate RC axis to stop	RSTP		○		○	○	○					495
RC axis information acquisition	Get RC axis status	RCST		○		○	○	○					496

Electronic Cam Control System Related Commands

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Extension motion control board input operations	Clear input counter record for extension motion control board	XCRP		○			○						498
	Acquire current record of extension motion control board input counter	XGTP		○			○						499
Extension motion control board axis position operations	Read extension motion control board axis position data	XPGT		○			○						500
	Write extension motion control board axis position data	XPPT		○			○						501
	Erase extension motion control board axis position data	XPCR		○			○						502
	Copy extension motion control board axis position data	XPCP		○			○						503
	Read extension motion control board axis current command position	XPRD		○			○						504
	Read extension motion control board axis current command position (single-axis direct)	XPRQ		○			○						505
	Write extension motion control board axis speed data	XPVL		○			○						506
	Write extension motion control board axis acceleration data	XPAC		○			○						507
	Write extension motion control board axis deceleration data	XPDC		○			○						508
	Write extension motion control board axis positioning complete width data	XPIP		○			○						509
	Read extension motion control board axis speed data	XGVL		○			○						510
	Read extension motion control board axis acceleration data	XGAC		○			○						511
	Read extension motion control board axis deceleration data	XGDC		○			○						512
	Read extension motion control board axis positioning width data	XGIP		○			○						513

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Extension motion control board axis actuator control declarations	Set extension motion control board axis patterns	XAXS		○			○						514
Extension motion control board axis actuator control commands	Extension motion control board axis servo ON	XSON		○			○						515
	Extension motion control board axis servo OFF	XSOF		○			○						516
	Return extension motion control board axis to home position	XHOM		○			○						517
	Move extension motion control board axis to indicated position	XMVP		○			○						518
	Perform extension motion control board axis position relative movement	XMPI		○			○						519
	Move extension motion control board axis for position indicated interpolation	XMVL		○			○						520
	Move extension motion control board axis for position relative interpolation	XMLI		○			○						521
	Move extension motion control board axis to directly indicated absolute position	XMVD		○			○						522
	Move extension motion control board axis to directly indicated relative position	XMDI		○			○						523
	Perform extension motion control board axis jog operation	XJ□□		○			○						524
	Waiting for extension motion control board axis to finish positioning operation of axis used by self-program	XPED		○			○						525
	Cancel operation of extension motion control board axis	XSTP		○			○						526
	Waiting for extension motion control board axis positioning complete signal to be turned ON	XWIP		○			○						527
	Start synchronizing extension motion control board axis electronic cam (indicating main axis)	XCAS		○			○						529

Category	Function	Command	XSEL -J/K/ KE/KT/KET	XSEL -P/Q/ PCT/QCT	XSEL -JX/KX/ KETX	XSEL -PX/QX	XSEL -R/S	XSEL -RX/SX/ RXD/SXD	ASEL PSEL	SSEL	MSEL	TT/TTA	Page
Extension motion control board axis actuator control commands	Move extension motion control board axis individual electronic cam (indicating time)	XCTM		○			○						534
	Start synchronizing of extension motion control board axis electronic shaft	XSFS		○			○						536
	Cancel operation of extension motion control board axis	XSYE		○			○						538
Extension motion control board axis status acquirement	Acquire extension motion control board axis status (0 to 15 axis)	XAST		○			○						540

Safety Guide

"Safety Guide" has been written to use the machine safely and so prevent personal injury or property damage beforehand. Make sure to read it before the operation of this product.

Safety Precautions for Our Products

The common safety precautions for the use of any of our robots in each operation.

No.	Operation Description	Description
1	Model Selection	● This product has not been planned and designed for the application where high level of safety is required, so the guarantee of the protection of human life is impossible. Accordingly, do not use it in any of the following applications. 1) Medical equipment used to maintain, control or otherwise affect human life or physical health. 2) Mechanisms and machinery designed for the purpose of moving or transporting people (For vehicle, railway facility or air navigation facility) 3) Important safety parts of machinery (Safety device, etc.) ● Do not use the product outside the specifications. Failure to do so may considerably shorten the life of the product. ● Do not use it in any of the following environments. 1) Location where there is any inflammable gas, inflammable object or explosive 2) Place with potential exposure to radiation 3) Location with the ambient temperature or relative humidity exceeding the specification range 4) Location where radiant heat is added from direct sunlight or other large heat source 5) Location where condensation occurs due to abrupt temperature changes 6) Location where there is any corrosive gas (sulfuric acid or hydrochloric acid) 7) Location exposed to significant amount of dust, salt or iron powder 8) Location subject to direct vibration or impact ● For an actuator used in vertical orientation, select a model which is equipped with a brake. If selecting a model with no brake, the moving part may drop when the power is turned OFF and may cause an accident such as an injury or damage on the work piece.

No.	Operation Description	Description
2	Transportation	● When carrying a heavy object, do the work with two or more persons or utilize equipment such as crane. ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● When in transportation, consider well about the positions to hold, weight and weight balance and pay special attention to the carried object so it would not get hit or dropped. ● Transport it using an appropriate transportation measure. The actuators available for transportation with a crane have eyebolts attached or there are tapped holes to attach bolts. Follow the instructions in the instruction manual for each model. ● Do not step or sit on the package. ● Do not put any heavy thing that can deform the package, on it. ● When using a crane capable of 1t or more of weight, have an operator who has qualifications for crane operation and sling work. ● When using a crane or equivalent equipments, make sure not to hang a load that weighs more than the equipment's capability limit. ● Use a hook that is suitable for the load. Consider the safety factor of the hook in such factors as shear strength. ● Do not get on the load that is hung on a crane. ● Do not leave a load hung up with a crane. ● Do not stand under the load that is hung up with a crane.
3	Storage and Preservation	● The storage and preservation environment conforms to the installation environment. However, especially give consideration to the prevention of condensation. ● Store the products with a consideration not to fall them over or drop due to an act of God such as earthquake.
4	Installation and Start	(1) Installation of Robot Main Body and Controller, etc. ● Make sure to securely hold and fix the product (including the work part). A fall, drop or abnormal motion of the product may cause a damage or injury. Also, be equipped for a fall-over or drop due to an act of God such as earthquake. ● Do not get on or put anything on the product. Failure to do so may cause an accidental fall, injury or damage to the product due to a drop of anything, malfunction of the product, performance degradation, or shortening of its life. ● When using the product in any of the places specified below, provide a sufficient shield. 1) Location where electric noise is generated 2) Location where high electrical or magnetic field is present 3) Location with the mains or power lines passing nearby 4) Location where the product may come in contact with water, oil or chemical droplets

No.	Operation Description	Description
4	Installation and Start	(2) Cable Wiring ● Use our company's genuine cables for connecting between the actuator and controller, and for the teaching tool. ● Do not scratch on the cable. Do not bend it forcibly. Do not pull it. Do not coil it around. Do not insert it. Do not put any heavy thing on it. Failure to do so may cause a fire, electric shock or malfunction due to leakage or continuity error. ● Perform the wiring for the product, after turning OFF the power to the unit, so that there is no wiring error. ● When the direct current power (+24V) is connected, take the great care of the directions of positive and negative poles. If the connection direction is not correct, it might cause a fire, product breakdown or malfunction. ● Connect the cable connector securely so that there is no disconnection or looseness. Failure to do so may cause a fire, electric shock or malfunction of the product. ● Never cut and/or reconnect the cables supplied with the product for the purpose of extending or shortening the cable length. Failure to do so may cause the product to malfunction or cause fire.
		(3) Grounding ● The grounding operation should be performed to prevent an electric shock or electrostatic charge, enhance the noise-resistance ability and control the unnecessary electromagnetic radiation. ● For the ground terminal on the AC power cable of the controller and the grounding plate in the control panel, make sure to use a twisted pair cable with wire thickness $0.5mm^2$ (AWG20 or equivalent) or more for grounding work. For security grounding, it is necessary to select an appropriate wire thickness suitable for the load. Perform wiring that satisfies the specifications (electrical equipment technical standards). ● Perform Class D Grounding (former Class 3 Grounding with ground resistance 100Ω or below).

No.	Operation Description	Description
4	Installation and Start	(4) Safety Measures ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● When the product is under operation or in the ready mode, take the safety measures (such as the installation of safety and protection fence) so that nobody can enter the area within the robot's movable range. When the robot under operation is touched, it may result in death or serious injury. ● Make sure to install the emergency stop circuit so that the unit can be stopped immediately in an emergency during the unit operation. ● Take the safety measure not to start up the unit only with the power turning ON. Failure to do so may start up the machine suddenly and cause an injury or damage to the product. ● Take the safety measure not to start up the machine only with the emergency stop cancellation or recovery after the power failure. Failure to do so may result in an electric shock or injury due to unexpected power input. ● When the installation or adjustment operation is to be performed, give clear warnings such as "Under Operation; Do not turn ON the power!" etc. Sudden power input may cause an electric shock or injury. ● Take the measure so that the work part is not dropped in power failure or emergency stop. ● Wear protection gloves, goggle or safety shoes, as necessary, to secure safety. ● Do not insert a finger or object in the openings in the product. Failure to do so may cause an injury, electric shock, damage to the product or fire. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity.
5	Teaching	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● Perform the teaching operation from outside the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the "Stipulations for the Operation" and make sure that all the workers acknowledge and understand them well. ● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency. ● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly. ● Place a sign "Under Operation" at the position easy to see. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity. * Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated.

No.	Operation Description	Description
6	Trial Operation	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● After the teaching or programming operation, perform the check operation one step by one step and then shift to the automatic operation. ● When the check operation is to be performed inside the safety protection fence, perform the check operation using the previously specified work procedure like the teaching operation. ● Make sure to perform the programmed operation check at the safety speed. Failure to do so may result in an accident due to unexpected motion caused by a program error, etc. ● Do not touch the terminal block or any of the various setting switches in the power ON mode. Failure to do so may result in an electric shock or malfunction.
7	Automatic Operation	● Check before starting the automatic operation or rebooting after operation stop that there is nobody in the safety protection fence. ● Before starting automatic operation, make sure that all peripheral equipment is in an automatic-operation-ready state and there is no alarm indication. ● Make sure to operate automatic operation start from outside of the safety protection fence. ● In the case that there is any abnormal heating, smoke, offensive smell, or abnormal noise in the product, immediately stop the machine and turn OFF the power switch. Failure to do so may result in a fire or damage to the product. ● When a power failure occurs, turn OFF the power switch. Failure to do so may cause an injury or damage to the product, due to a sudden motion of the product in the recovery operation from the power failure.

No.	Operation Description	Description
8	Maintenance and Inspection	● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers. ● Perform the work out of the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the “Stipulations for the Operation” and make sure that all the workers acknowledge and understand them well. ● When the work is to be performed inside the safety protection fence, basically turn OFF the power switch. ● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency. ● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly. ● Place a sign “Under Operation” at the position easy to see. ● For the grease for the guide or ball screw, use appropriate grease according to the Instruction Manual for each model. ● Do not perform the dielectric strength test. Failure to do so may result in a damage to the product. ● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity. ● The slider or rod may get misaligned OFF the stop position if the servo is turned OFF. Be careful not to get injured or damaged due to an unnecessary operation. ● Pay attention not to lose the cover or untightened screws, and make sure to put the product back to the original condition after maintenance and inspection works. Use in incomplete condition may cause damage to the product or an injury. * Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated.
9	Modification and Dismantle	● Do not modify, disassemble, assemble or use of maintenance parts not specified based at your own discretion.
10	Disposal	● When the product becomes no longer usable or necessary, dispose of it properly as an industrial waste. ● When removing the actuator for disposal, pay attention to drop of components when detaching screws. ● Do not put the product in a fire when disposing of it. The product may burst or generate toxic gases.
11	Other	● Do not come close to the product or the harnesses if you are a person who requires a support of medical devices such as a pacemaker. Doing so may affect the performance of your medical device. ● See Overseas Specifications Compliance Manual to check whether complies if necessary. ● For the handling of actuators and controllers, follow the dedicated instruction manual of each unit to ensure the safety.

Alert Indication

The safety precautions are divided into "Danger", "Warning", "Caution" and "Notice" according to the warning level, as follows, and described in the Instruction Manual for each model.

Level	Degree of Danger and Damage	Symbol
Danger	This indicates an imminently hazardous situation which, if the product is not handled correctly, will result in death or serious injury.	⚠ Danger
Warning	This indicates a potentially hazardous situation which, if the product is not handled correctly, could result in death or serious injury.	⚠ Warning
Caution	This indicates a potentially hazardous situation which, if the product is not handled correctly, may result in minor injury or property damage.	⚠ Caution
Notice	This indicates lower possibility for the injury, but should be kept to use this product properly.	! Notice

1. Preparation in Advance

SEL language is the simplest type of language in many existing robot languages.
Even though SEL language is an interpreter program, it enables to perform high level controls in simple expression ways.
In this manual, describes how to use SEL language, explanations of command language, examples of how to create programs for each actuator, etc.

In this section, explains what are needed to be prepared beforehand to start programming, or the things that you need to know for programming.

1.1 Related Manuals

Please make sure to refer also to the instruction manuals for the controller and accessories that you intend to use. Listed below are the related instruction manuals.

No.	Name	Manual No.
1	XSEL-J/K Controller Instruction Manual	ME0116
2	XSEL-KT Controller Instruction Manual	ME0134
3	XSEL-JX/KX Controller Instruction Manual	ME0119
4	XSEL-P/Q/PCT/QCT Controller Instruction Manual	ME0148
5	XSEL-PX/QX Controller Instruction Manual	ME0152
6	PSEL Controller Instruction Manual	ME0172
7	ASEL Controller Instruction Manual	ME0165
8	SSEL Controller Instruction Manual	ME0154
9	TT Controller	ME0149
10	PC Software IA-101-X-MW/IA-101-X-USBMW	ME0154
11	Teaching Pendant SEL-T/TD/TG	ME0183
12	Teaching Pendant IA-T-X/XD	ME0160
13	DeviceNet Instruction Manual	ME0124
14	CC-Link Instruction Manual	ME0123
15	PROFIBUS Instruction Manual	ME0153
16	XSEL Ethernet Instruction Manual	ME0140
17	XSEL Controller RC Gateway Function Instruction Manual	ME0188
18	XSEL-P/Q/PCT/QCT Controller Electronic Cam Function Instruction Manual	ME0246
19	OMRON Vision Sensor Tracking Instruction Manual	ME0237
20	Keyence Vision Sensor Tracking Instruction Manual	ME0238
21	Cognex Vision Sensor Tracking Instruction Manual	ME0239
22	XSEL-P/Q/PCT/QCT Controller Vision System I/F Function Instruction Manual	ME0264
23	XSEL-R/S/RX/SX/RXD/SXD Controller Instruction Manual	ME0308
24	Tabletop Robot TTA Instruction Manual	ME0320
25	MSEL Instruction Manual	ME0336

1.2 Programming Tool

To create a program with SEL language, it is necessary to prepare a dedicated teaching pendant or PC software provided by IAI.
Please confirm in the table below that the controller you intend to use complies with the programming tool that you have.

No.	Item	Controller Model / Model Code of Programming Tool	XSEL-J/K/KE/KT/KET	XSEL-P/PCT/R	XSEL-Q/QCT/S	XSEL-JX/KX/KETX	XSEL-PX/RX/RXD	XSEL-QX/SX/SXD	ASEL	PSEL	SSEL	TT/TTA	MSEL
1	PC software (with RS232C cable + emergency stop box)	IA-101-X-MW	○	○	○	○	○	○				○	
2	PC software (with USB conversion adapter + RS232C cable + emergency stop box)	IA-101-X-USBMW	○	○	○	○	○	○				○	
3	PC software (with RS232C cable + emergency stop box + connector conversion cable)	IA-101-X-MW-J							○	○	○		○
4	PC software (with USB cable + dummy plug)	IA-101-X-USB							○	○	○		○
5	PC software (with safety category 4 cable + emergency stop box)	IA-101-XA-MW			○			○					
6	Teaching pendant	SEL-T	○ (J is excluded)	○	○	○	○	○	○	○	○	○	
7	Teaching pendant (with deadman switch)	SEL-TD	○ (J is excluded)	○	○	○	○	○	○	○	○	○	
8	Teaching pendant (with TP adapter for Safety Categories)	SEL-TG	○	○	○	○	○	○	○	○	○	○	
9	Teaching pendant	IA-T-X	○	○ (Q is excluded)	○	○	○	○ (QX is excluded)	○	○	○	○	
10	Teaching pendant (with deadman switch)	IA-T-XD	○	○ (Q is excluded)	○	○	○	○ (QX is excluded)	○	○	○	○	
11	Touch panel teaching	TB-01	○	○	○	○	○	○	○	○	○	○	○
12	Touch panel teaching (with deadman switch)	TB-01D	○	○	○	○	○	○	○	○	○	○	○

○: Applicable, Blank: Not applicable

1.3 PC Operational Environment

If you use the PC software, make sure your PC meets the following specifications before installing the software. [Refer to PC Software Instruction Manual for how to install it.]
Also, confirm in the next section that it is applicable for the controller that you intend to use.

1) Operating System (OS)
   Windows 2000 SP4 or later, Windows XP SP2 or later, Windows Vista, Windows 7
2) Main Memory
   It should possess memory capacity necessary to operate Windows®.
3) Display Monitor
   XGA or more.
4) Hard Disk
   Hard disk with free space of 20MB or more
   (This software is to be used with being installed in the hard disk.)
5) Serial Port
   There should be 1 unit of RS232C port that is capable for the communication speed setting of 9600bps or more.
   (Note) This is for the case the model code of PC software is IA-101-*-MW.
6) USB Port
   There should be 1 unit of USB port with its version 1.1 or more.
   (Note) This is for the case the model code of PC software is IA-101-*-USBMW.
7) Keyboard
   It should comply with the PC main unit. (PC/AT compatible keyboard)
8) Pointing Device
   It should be operated in Windows® OS.
9) Drive Device
   The PC should possess a CD-ROM drive device that complies with the PC or a compatible drive device that can read CD-ROM.

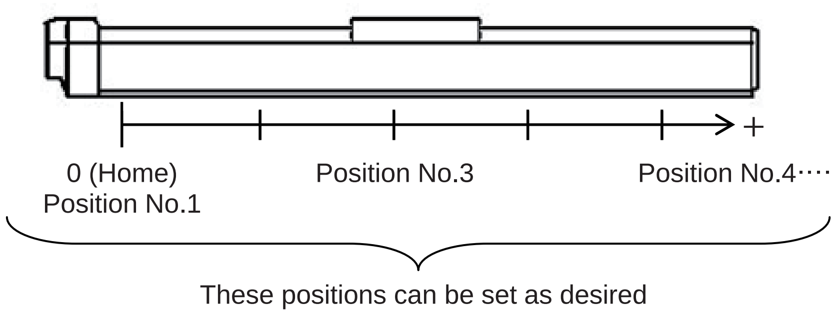
* SEL language is available on the following controllers.
   1) XSEL (all types)
   2) ASEL
   3) PSEL
   4) SSEL
   5) TT/TTA
   6) MSEL

1.4 Axes on Each Actuator and Precautions

In this section, explains the construction of axis number of each actuator on the 3-dimensional coordinate system, X, Y and Z.

1.4.1 Single-Direction Axis

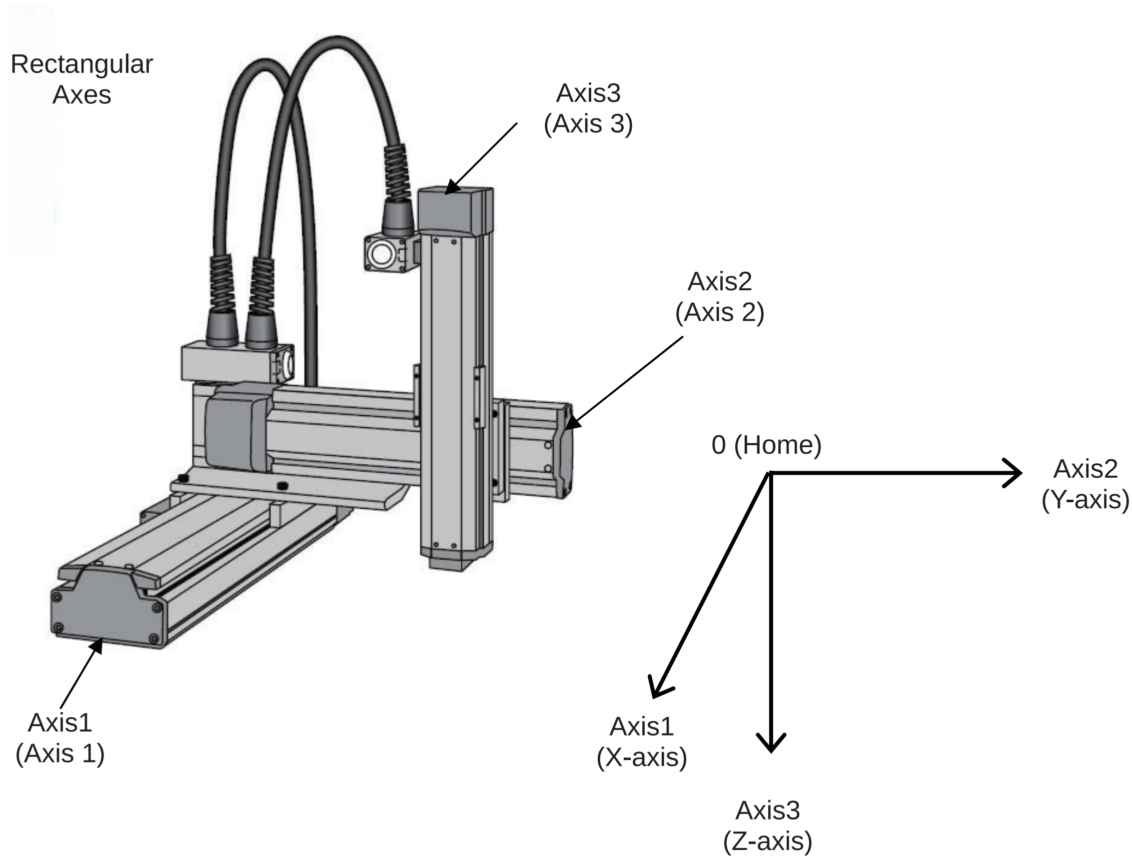
The coordinate value from the home corresponds to 0mm in position data.
Positions from the home represent position data.
The direction is reversed if the actuator is of reversed-home specification.



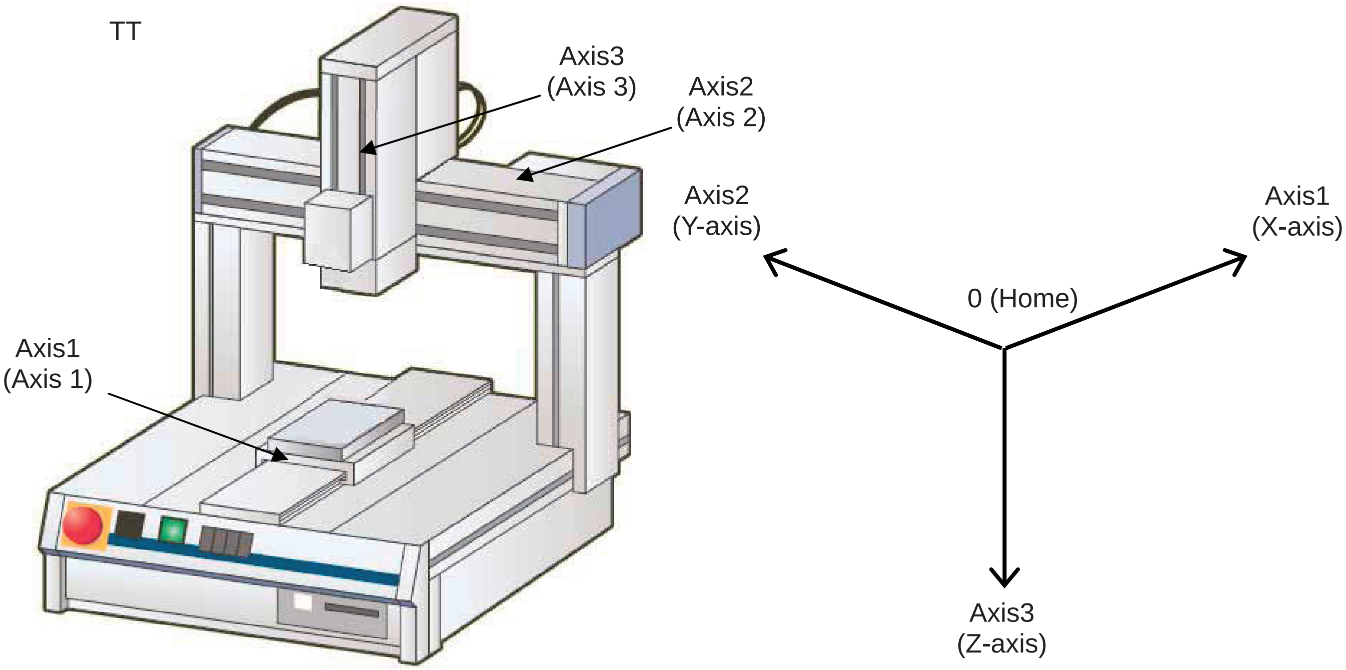
Axis Number of the position table will be that of the connector that is connected physically to the actuator motor and encoder cable. [Refer to the instruction manual of the controller for the details.]

1.4.2 Rectangular Axes, TT*

The coordinate value from the home of each axis corresponds to 0mm in position data.
With each axis, positions from the home represent position data.

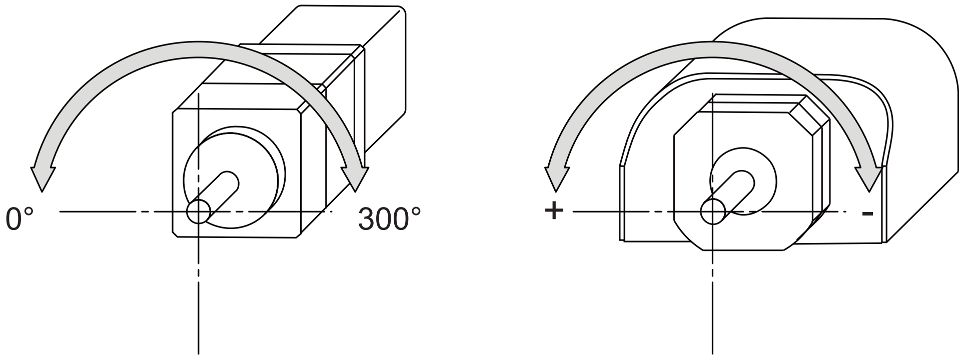


Axis Number is that the actuator motor and encoder cable is connected physically.
The coordinate system shown in the figure below is when each actuator motor and encoder cable is connected to the Axis Number in bracket.
Refer to the instruction manual of each controller for the details.



1.4.3 Rotational Axis

The coordinate value from the home corresponds to 0° in position data.
Rotational angles from the home represent position data.

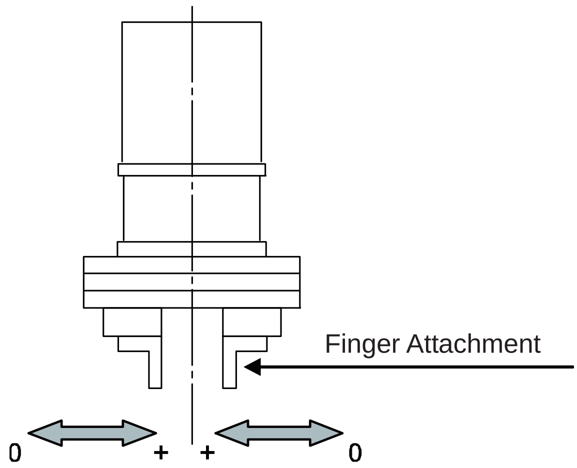


To the rotation axes, there is the finite stroke type that operates within the established angular range and the infinite stroke type that can rotate for a number of times in the indicated direction.
Refer to the instruction manual of each actuator for the details.

1.4.4 Gripper

The coordinate value (opening side) from the home corresponds to 0mm in position data.
1/2 stroke[*1] from the home represent position data.

*1 Stroke : Distance between both grippers



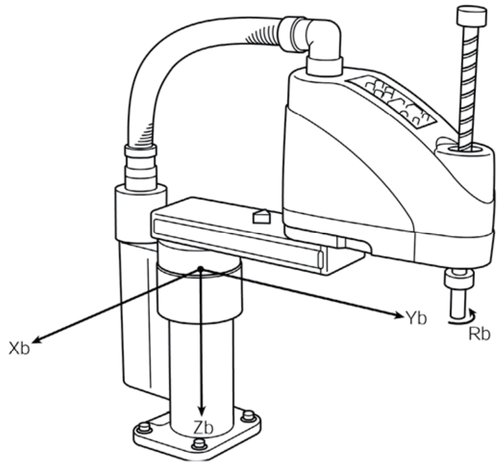
1.4.5 SCARA Robot

[1] Coordinates

A horizontal articulated (SCARA) robot has three coordinate systems including the base coordinate system, work coordinate system and tool coordinate system.

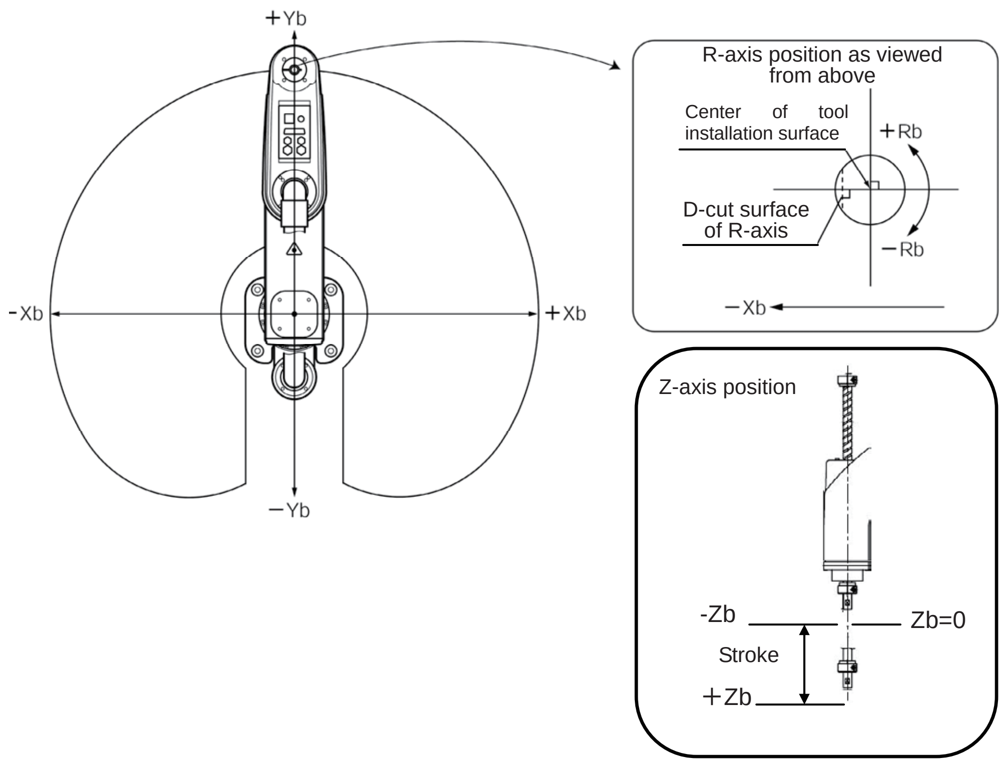
Coordinates
- Base coordinate system
- Work coordinate system
- Tool coordinate system

(1) Base coordinate system (= Work Coordinate System No. 0)
This is a combination of three-dimensional rectangular coordinates and rotational axis coordinates defined in the robot prior to shipment.
Work Coordinate System No. 0 (= 0 work coordinate system offsets) = Base coordinate system.



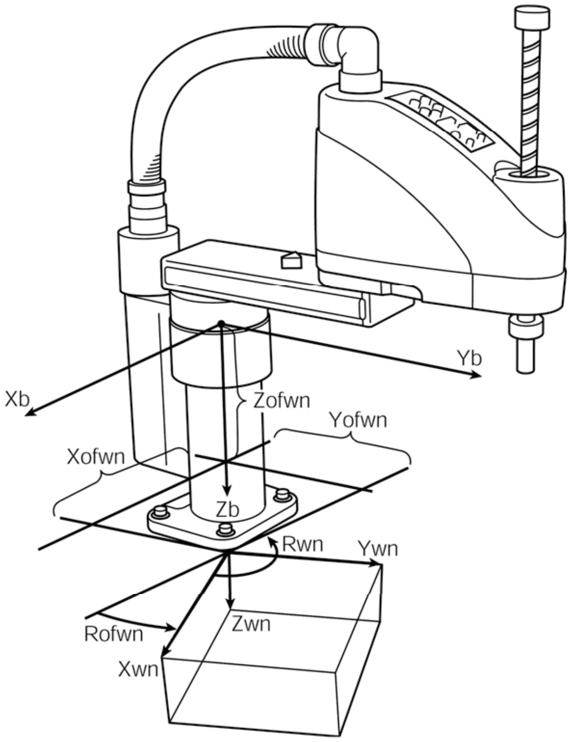
* There is no rotary axis on Three-Axis Type SCARA ROBOT (IXP-3N****). (Xb, Yb and Zb) are available to indicate as the target position.

The XY-axis home is the center of the base (center of rotation of arm 1).
The Z-axis home is the top edge of the effective Z-axis stroke.
The R-axis home is where the D-cut surface faces the -Xb direction.
The X-axis, Y-axis, Z-axis and R-axis on the base coordinate system are indicated as Xb, Yb, Zb and Rb, respectively.



(2) Work coordinate system

This is a combination of 32 types of three-dimensional rectangular coordinates and rotational axis coordinates defined by the offset of each axis relative to the base coordinate system. Note that Work Coordinate System No. 0 is reserved as the base coordinate (= 0 work coordinate system offset) by the system.



Xofwn: X work coordinate offset
Yofwn: Y work coordinate offset
Zofwn: Z work coordinate offset
Rofwn: R work coordinate offset

Xwn: Work coordinate system, X-axis
Ywn: Work coordinate system, Y-axis
Zwn: Work coordinate system, Z-axis
Rwn: Work coordinate system, R-axis

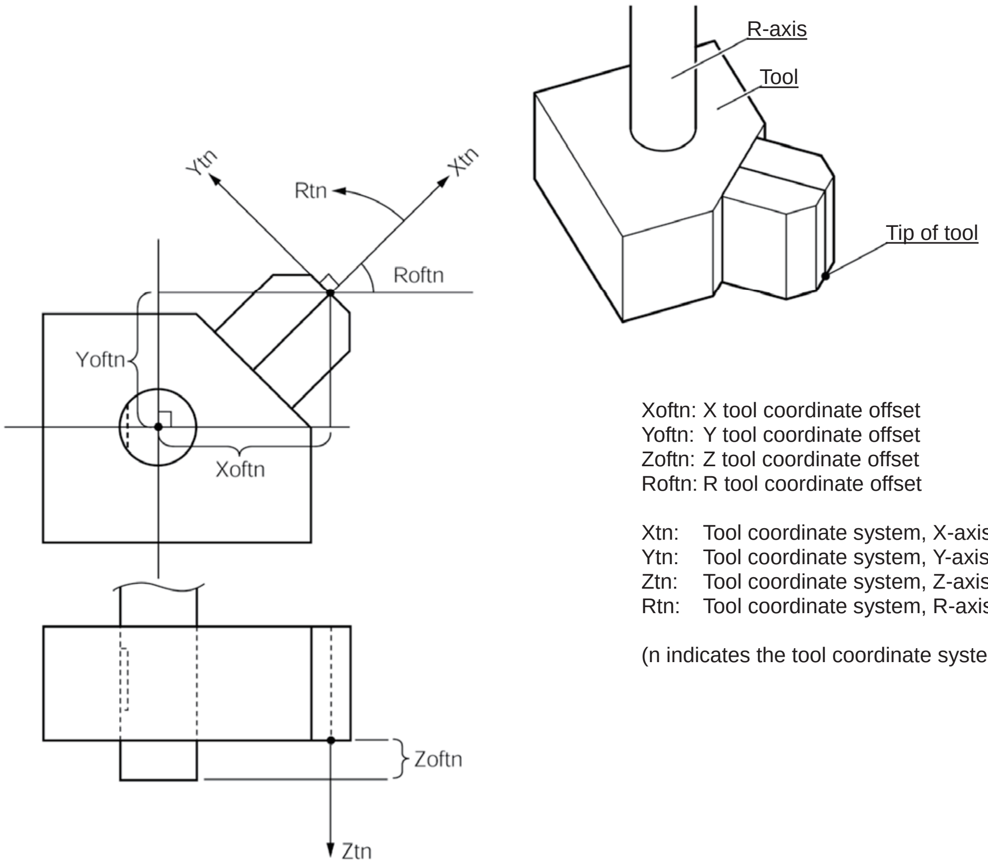
(n indicates the work coordinate system number.)

* R work coordinate offset is valid also on Three-Axis Type SCARA ROBOT (IXP-3N****).

(3) Tool coordinate system

This is a combination of 128 types of three-dimensional rectangular coordinates and rotational axis coordinates defined by the dimension (offset) of the tool (hand, etc.) installed on the tool installation surface. Note that Tool Coordinate System No. 0 is reserved as one with 0 tool coordinate system offset by the system.

When a defined tool coordinate system number is selected, the robot uses the tip of the tool, not the center of the tool installation surface, as the point to reach by positioning.

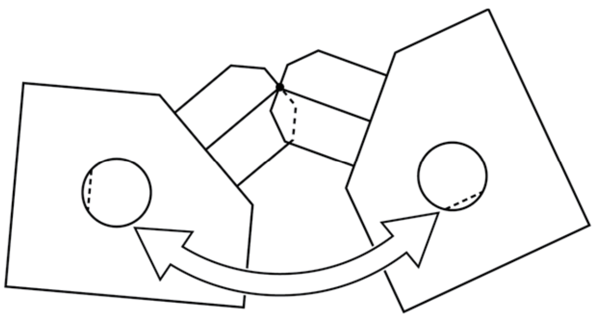


Xoftn: X tool coordinate offset
Yoftn: Y tool coordinate offset
Zoftn: Z tool coordinate offset
Roftn: R tool coordinate offset

Xtn: Tool coordinate system, X-axis
Ytn: Tool coordinate system, Y-axis
Ztn: Tool coordinate system, Z-axis
Rtn: Tool coordinate system, R-axis

(n indicates the tool coordinate system number.)

When a defined tool coordinate system is elected and the R-axis is jogged, the axis operates as shown below.



<u>The orientation of the tool coordinate system axis is always the same as the orientation of the base coordinate system axis</u> on Three-Axis Type SCARA ROBOT (IXP-3N****). As there is no R axis, the control of the orientation (posture) of the tool cannot be conducted. Therefore, there may be a case the tool end may not be the reachable point at positioning. Also, the setting of the tool coordinate system offset on the R axis will be ignored.

[2] CP Operation and PTP Operation

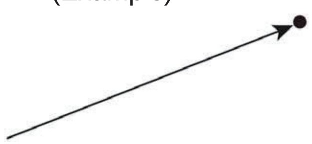
How CP operation and PTP operation differ as they pertain to SCARA robots is explained.

(1) CP operation

1) Path

The axes move to the target position while interpolating with one another. The path of the tip of movement can be specified by a command (linear, circle, arc, path movement, etc.).

(Example)



MOVL 1

Move from the current position to position No. 1 along a straight line.

The arm system does not change during CP operation.

CP operation commands: MOVL MVLI TMLI PATH PSPL PUSH CIR2 ARC2 ARCD ARCC CIRS ARCS CIR ARC

For details on these commands, refer to “Explanation of Each Command”

2) Setting of speed and acceleration/deceleration for CP operation

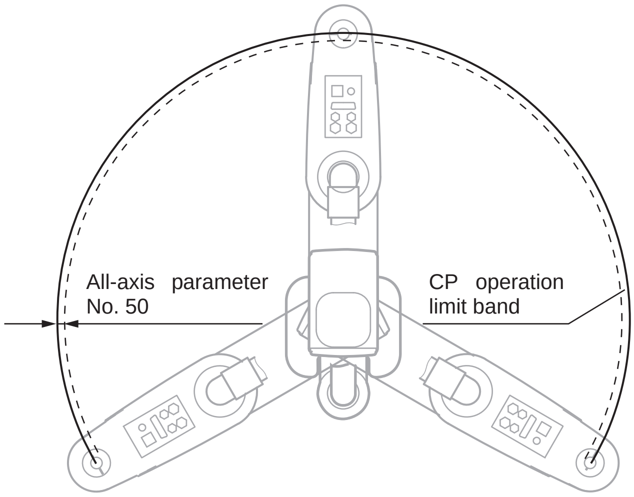
In CP operation, the speed and acceleration/deceleration are set beforehand in the program using a control declaration command.
Speed setting command “VEL” unit [mm/sec]
Acceleration setting command “ACC” unit [G]
Deceleration setting command “DCL” unit [G]

(Example)

ACC	0.5	Set the acceleration for CP operation to 0.5G.
DCL	0.5	Set the deceleration for CP operation to 0.5G.
VEL	500	Set the speed for CP operation to 500mm/sec.
MOVL	2	Move to position No. 2 along a straight line.

The speed and acceleration/deceleration for CP operation can also be set in the VEL, ACC and DCL fields of the position data table.
To set these items as part of position data, do so for each position number. If the VEL, ACC and DCL fields of the position data table contain settings for a given position number, they are given priority over the “VEL”, “ACC” and “DCL” commands in the program when moving to the applicable position number.

3) Notes on CP operation
The singular point is where both arms 1 and 2 extend straight.
If the actuator moves near the singular point via CP operation, poor path precision, vibration (abnormal noise) or error may occur. The following errors may generate: "D09: Driver overspeed error", "B91: Main overspeed error", "C64: Abnormal servo acceleration/deceleration error", "B74: CP operation limit band entry error", "CB6: Deviation overflow error", etc.
These errors may be avoided by lowering the speed and/or acceleration/deceleration.



With the area inside for the amount of either All Axes Parameter No. 50 or No. 227 from the singularity defined as the CP operation limit band, the CP operation is limited within this area. (In the figure on the left, the area between the solid line and dotted line is the CP operation limit band.)

The controller generates an error upon detecting an entry of the target path or actual movement path into the CP operation limit band as a result of path calculation. If the target movement path enters the CP operation band as a result of path calculation, "B7C: Target path in CP operation limit band error (PTP/jotting of axis permitted)" occurs.

If the actual movement path enters the CP operation limit band, on the other hand, "B74: CP operation limit band entry error (PTP/jogging of axis permitted)" or "C74: Actual position soft limit over error" occurs.
The width of the CP operation limit band (distance between the solid line and dotted line) varies depending on the arm length of the robot.
(If the arm length is 500 or 600, this width becomes approx. 0.5mm (All Axes Parameter No. 50 or No. 227, "Width of CP operation limit band around point directly above Arm 1/2").)

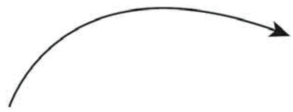
Avoid writing programs that cause the actuator to pass the CP operation limit band via CP operation.
The actuator cannot pull out from the CP operation limit band by means of CP operation. In this case, move the actuator in PTP operation. Exercise caution in situations where the condition of each arm is not recognized, such as when the program is started, etc.
With CP operation, conduct test operation at low speed at first and confirm absence of problems, and then gradually raise the speed to an appropriate level.

(2) PTP operation

1) Movement path

Each axis moves to the target position at the specified speed. The path of the tip of movement cannot be specified by a command.

(Example)



Position No. 1

MOVP 1

Move from the current position to position No. 1 via PTP operation.

The arm system may change during movement depending on the operation area and arm system control command.

PTP operation commands: MOVP MVPI TMPI PACH PMVP ARCH

For details on these commands, refer to “Explanation of Each Command”

2) Setting of speed and acceleration/deceleration for PTP operation

In PTP operation, the speed and acceleration/deceleration are set beforehand in the program using a control declaration command.

Speed setting command “VELS”, unit [% (percentage relative to the maximum PTP speed (SCARA axis) set in axis-parameter No. 28)]

Acceleration setting command “ACCS”, unit [% (percentage relative to the maximum PTP acceleration for SCARA axis set in axis-parameter No. 134)]

Deceleration setting command “DCLS”, unit [% (percentage relative to the maximum PTP deceleration for SCARA axis set in axis-parameter No. 135)]

(Example)

ACCS	50	Set the acceleration for PTP operation to 50% of the maximum PTP acceleration.
DCLS	50	Set the deceleration for PTP operation to 50% of the maximum PTP deceleration.
VELS	50	Set the speed for PTP operation to 50% of the maximum PTP speed.
MOVP	2	Move to position 2 via PTP operation.

3) Note on PTP operation

The arm system may change during an operation depending on the operation area, arm system control command or position data arm system.

Refer to “[4] Arm System”.

[3] Settings of Each Axis

(1) Base coordinate system

1) Positioning on Base Coordinate System

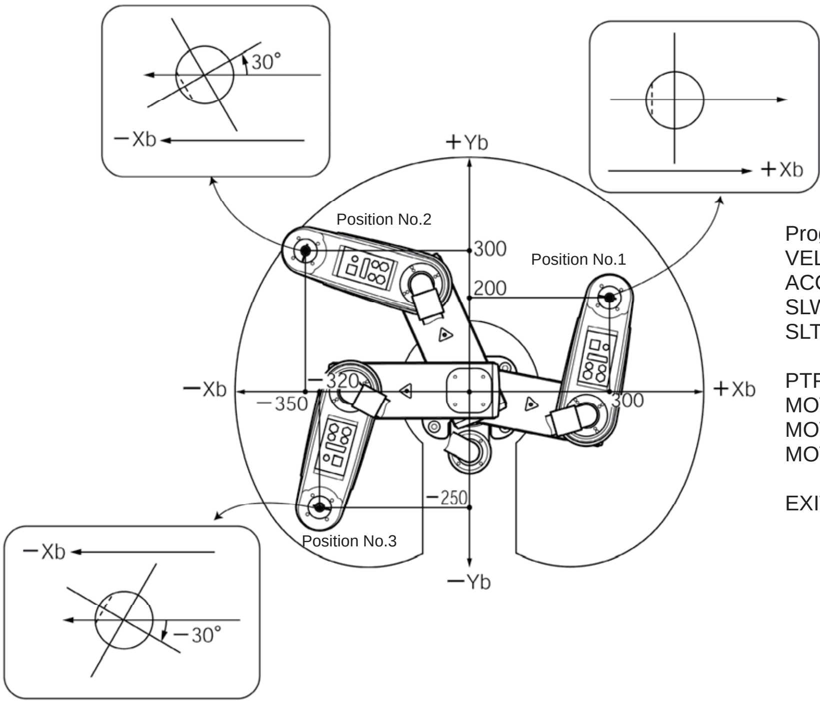
To select a work coordinate system number in the SEL program, use a SLWK command.
The work coordinate system selection number that has been set will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected. (For XSEL-RX/SX/RXD/SXD, battery is not necessary.)
The figure below shows a part of the position edit screen in the PC software for horizontal articulated robot.
In this example, the following teaching settings are assumed:
Position data for Position No. 1: X = 300, Y = 200, Z = 0, R = 0
Position data for Position No. 2: X = -350, Y = 300, Z = 50, R = 30
Position data for Position No. 3: X = -320, Y = -250, Z = 100, R = -30

The selected work coordinate system number is displayed.
Work Coordinate System No. 0 = Base coordinate system

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
1( )	300.000	200.000	0.000	0.000			
2( )	-350.000	300.000	50.000	30.000			
3( )	-320.000	-250.000	100.000	-30.000			
4( )							
5( )							

The selected tool coordinate system number is displayed. In the case of
Tool Coordinate System No. 0, the center of the tool installation surface is positioned.

2) Position to the position data shown above via PTP operation.



Program example

VELS	50	
ACCS	50	
SLWK	0	Select Work Coordinate System No. 0
SLTL	0	Select Tool Coordinate System No. 0
PTPR		Specify right arm of PTP target arm system
MOVP	1	
MOVP	2	
MOVP	3	
EXIT		

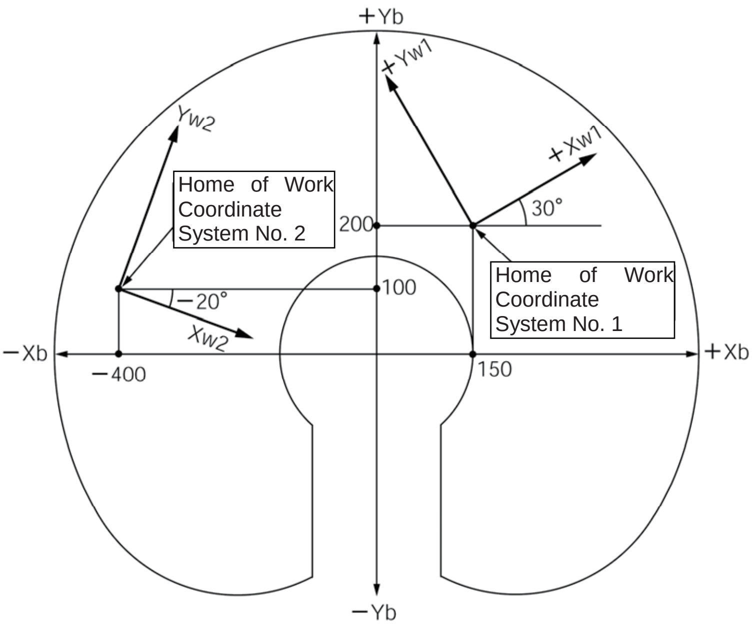
(2) Work coordinate system

1) Setting of Work Coordinate System

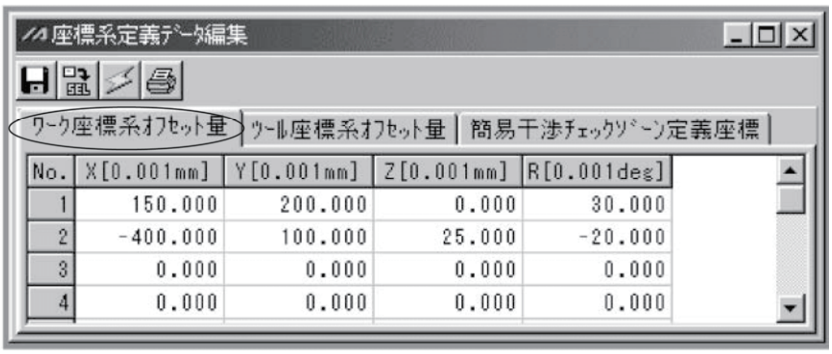
Set the offset relative to the base coordinate system.

(Example) Setting example of work coordinate system

Define Work Coordinate System No. 1 and No. 2 as shown below.



For the offset of Work Coordinate System No. 1, set Xofw1 = 150, Yofw1 = 200, Zofw1 = 0 and Rofw1 = 30.
For the offset of Work Coordinate System No. 2, set Xofw2 = -400, Yofw2 = 100, Zofw2 = 25 and Rofw2 = -20.
Shown below is the edit screen for work coordinate system definition data in the PC software for horizontal articulated robot, where Work Coordinate System No. 1 and No. 2 have been set.



No.	X[0.001mm]	Y[0.001mm]	Z[0.001mm]	R[0.001deg]
1	150.000	200.000	0.000	30.000
2	-400.000	100.000	25.000	-20.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000

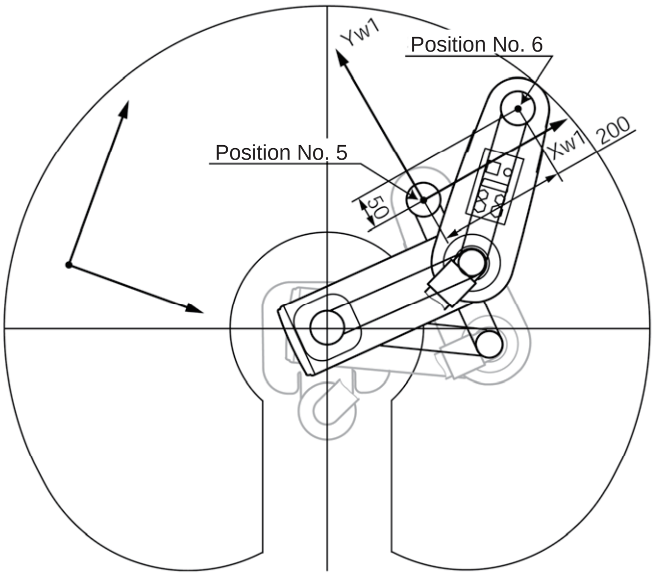
* To set a work coordinate system offset in the SEL program, use a DFWK command.

2) Positioning on Work Coordinate System
Perform positioning after selecting the work coordinate system you want to use.
To select a work coordinate system number in the SEL program, use a SLWK command.
The work coordinate system selection number that has been selected will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected.

(Example 1) Position to Position No. 5 and No. 6 via PTP operation on Work Coordinate System No. 1.

現在腕系 右腕系 変更
ｼﾞｮｸﾞ移動座標系 XY(ﾜｰｸ)座標系
ﾜｰｸ座標系選択No.(0=ﾍﾞｰｽ座標系) 1 変更
ﾂｰﾙ座標系選択No.(0=ﾂｰﾙｵﾌｾｯﾄ無し) 0 変更

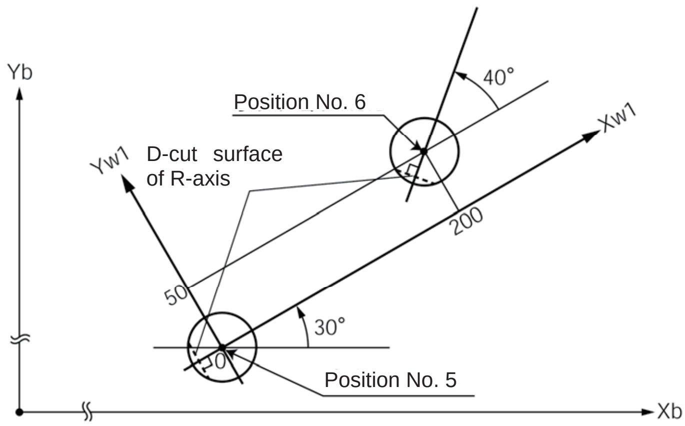
No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
4( )							
5( )	0.000	0.000	0.000	0.000			
6( )	200.000	50.000	20.000	40.000			
7( )							



Program example
:
:
:

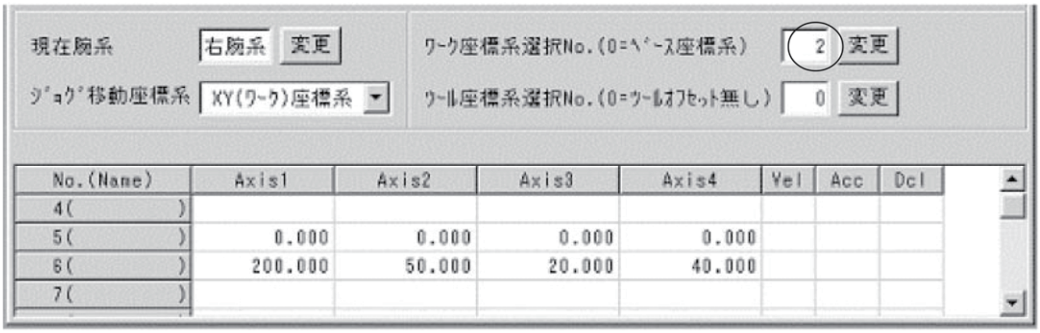
SLWK	1	Select Work Coordinate System No. 1
SLTL	0	Select Tool Coordinate System No. 0
PTPR		Specify right arm of PTP target arm system
MOVP	5	Move to Position No. 5.
MOVP	6	Move to Position No. 6.

:
:
:

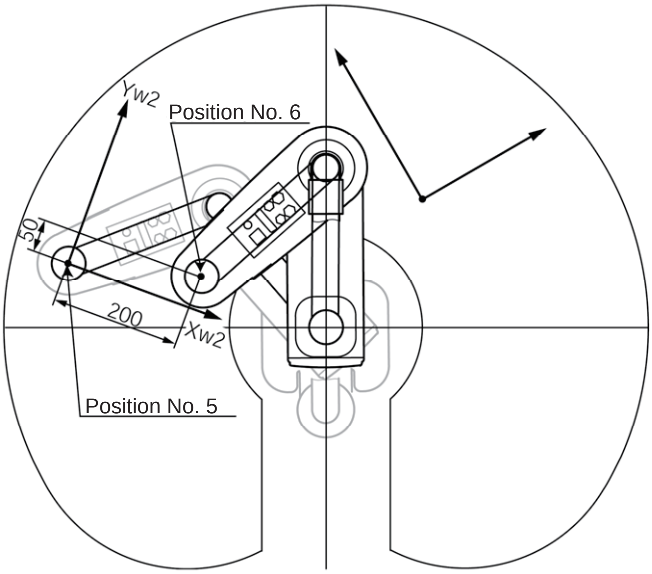


The R-axis position is shown to the left (viewed from above).
The Z-axis position is as follows:
Position No. 5 Zb = 0
Position No. 6 Zb = 20

(Example 2) Position to Position No. 5 and No. 6 via PTP operation on Work Coordinate System No. 2.

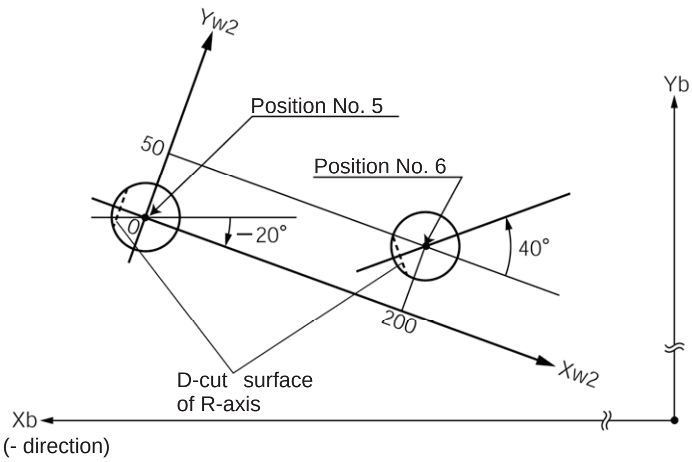


No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
4( )							
5( )	0.000	0.000	0.000	0.000			
6( )	200.000	50.000	20.000	40.000			
7( )							



Program example

```
     :
     :
SLWK    2    Select Work Coordinate System No. 2
SLTL    0    Select Tool Coordinate System No. 0
PTPR         Specify right arm of PTP target arm system
MOVP    5    Move to Position No. 5.
MOVP    6    Move to Position No. 6.
     :
     :
```



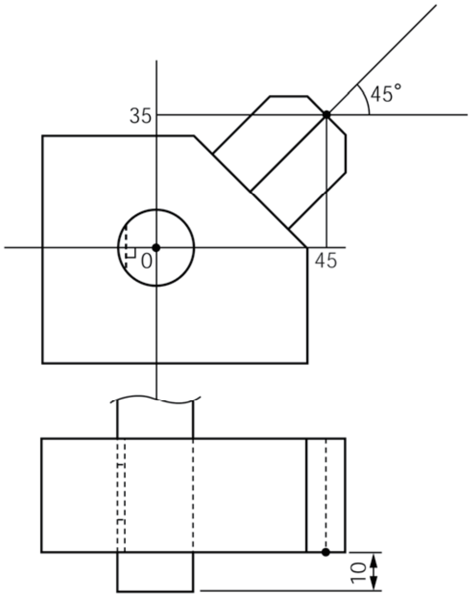
The R-axis position is shown to the left (viewed from above).
The Z-axis position is as follows:
Position No. 5 Zb = 25
Position No. 6 Zb = 45

(3) Tool coordinate system

1) Setting of Tool Coordinate System

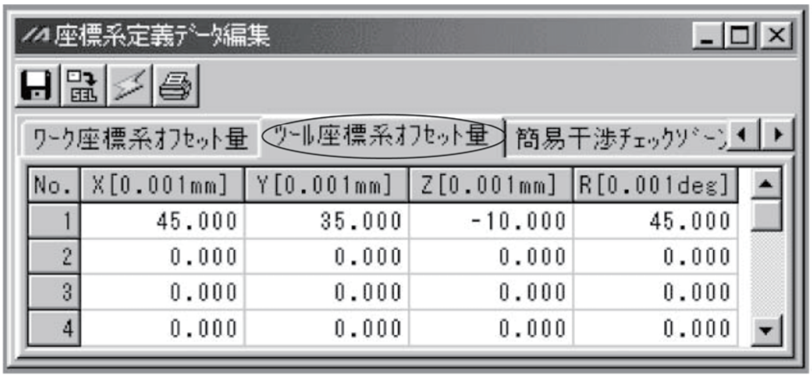
Set the offset from the center of the tool installation surface to tip of the tool.

(Example) Setting example of tool coordinate system
Define Tool Coordinate System No. 1 as shown below.



Offsets under Tool Coordinate System No. 1: Xoft1 = 45, Yoft1 = 35, Zoft1 = -10, Roft1 = 45

Shown below is the edit screen for tool coordinate system definition data in the PC software for horizontal articulated robot, where Tool Coordinate System No. 1 has been set.



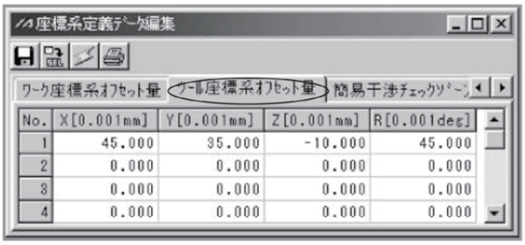
No.	X[0.001mm]	Y[0.001mm]	Z[0.001mm]	R[0.001deg]
1	45.000	35.000	-10.000	45.000
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000

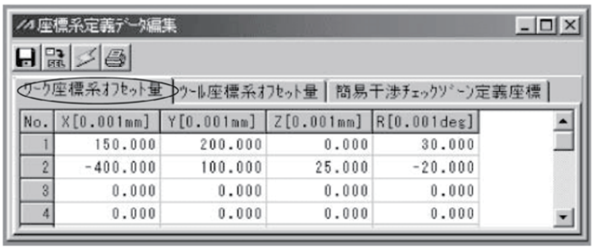
* To set a tool coordinate system offset in the SEL program, use a DFTL command.

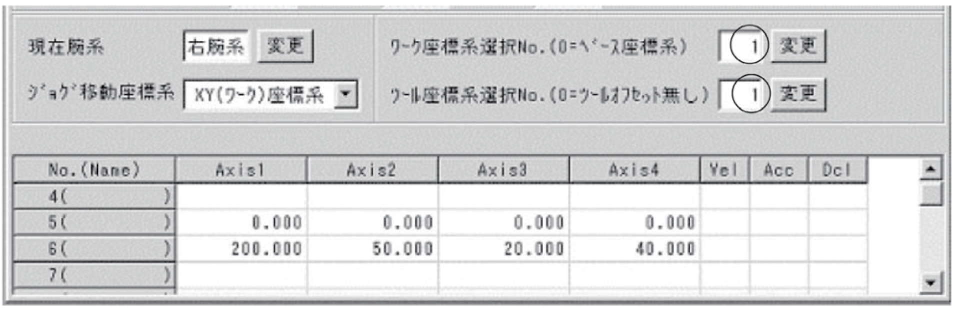
2) Positioning Using Tool Coordinate System Offset
Perform positioning after selecting the tool coordinate system you want to use.
To use a tool coordinate system number in the SEL program, use a SLTL command.
The tool coordinate system selection number that has been selected will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected. (For XSEL-RX/SX/RXD/SXD, battery is not necessary.)

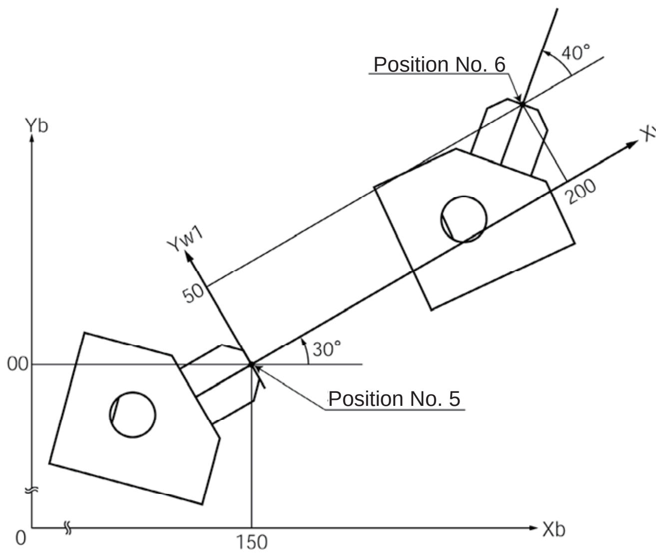
> The orientation of the tool coordinate system axis is always the same as the orientation of the base coordinate system axis on Three-Axis Type SCARA ROBOT (IXP-3N****). As there is no R axis, the control of the orientation (posture) of the tool cannot be conducted. Therefore, there may be a case the tool end may not be the reachable point at positioning. Also, the setting of the tool coordinate system offset on the R axis will be ignored.

(Example 1) Position the tip of the tool on Tool Coordinate System No. 1 to Position No. 5 and No. 6 on Work Coordinate System No. 1 via PTP operation.









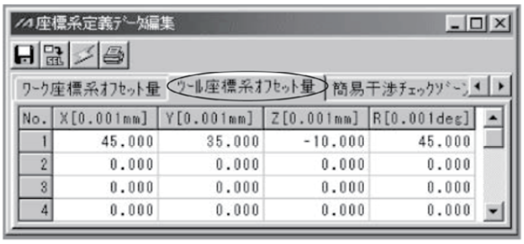
Program example

```
      :
SLWK    1    Select Work Coordinate System No. 1
SLTL    1    Select Tool Coordinate System No. 1
PTPR         Specify right arm of PTP target arm system
MOVP    5    Move to Position No. 5.
MOVP    6    Move to Position No. 6.
      :
```

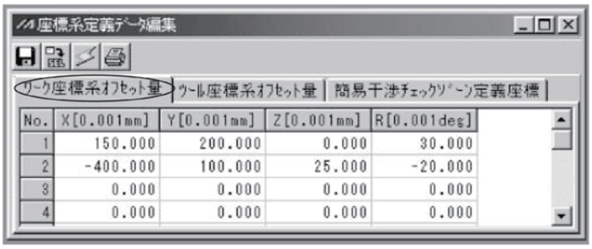
The Z-axis position at the tip of the tool is as follows:
Position No. 5 Zb = 0
Position No. 6 Zb = 20
The figure on the left is viewed from above.

1. Preparation in Advance

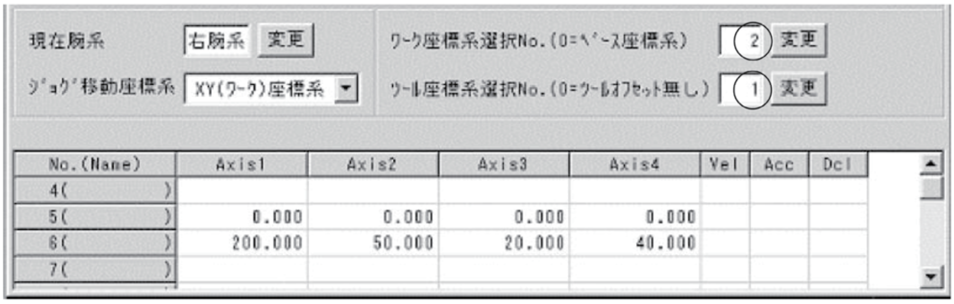
(Example 2) Position the tip of the tool on Tool Coordinate System No. 1 to Position No. 5 and No. 6 on Work Coordinate System No. 2 via PTP operation.



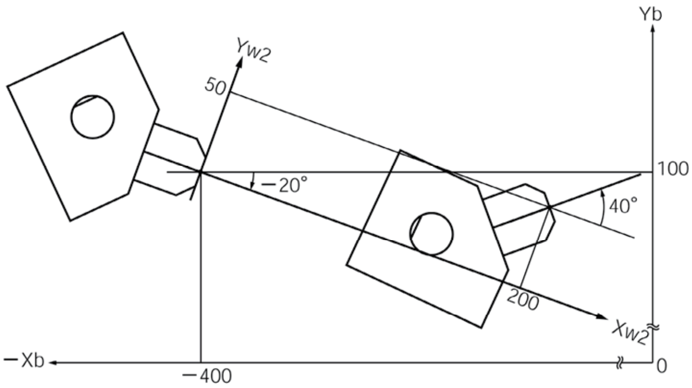
No.	X[0.001mm]	Y[0.001mm]	Z[0.001mm]	R[0.001deg]
1	45.000	35.000	-10.000	45.000
2	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000



No.	X[0.001mm]	Y[0.001mm]	Z[0.001mm]	R[0.001deg]
1	150.000	200.000	0.000	30.000
2	-400.000	100.000	25.000	-20.000
3	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000



No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
4( )							
5( )	0.000	0.000	0.000	0.000			
6( )	200.000	50.000	20.000	40.000			
7( )							



Program example

:

SLWK	2	Select Work Coordinate System No. 2
SLTL	1	Select Tool Coordinate System No. 1
PTPR		Specify right arm of PTP target arm system
MOVP	5	Move to Position No. 5.
MOVP	6	Move to Position No. 6.

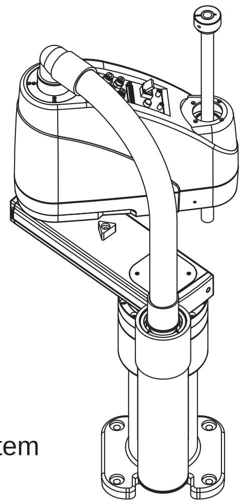
:

The Z-axis position at the tip of the tool is as follows:
Position No. 5 Zb = 25
Position No. 6 Zb = 45

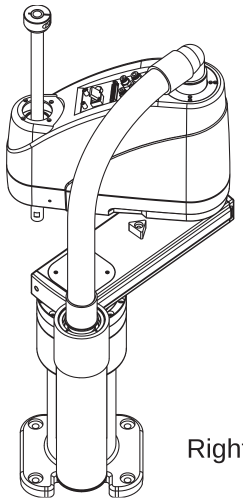
[4] Arm System

(1) Right arm system/left arm system

Robot postures are classified into two types: right arm system and left arm system.



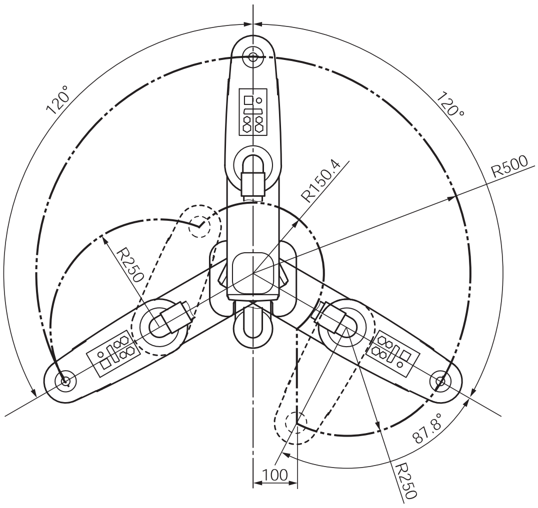
Left arm system



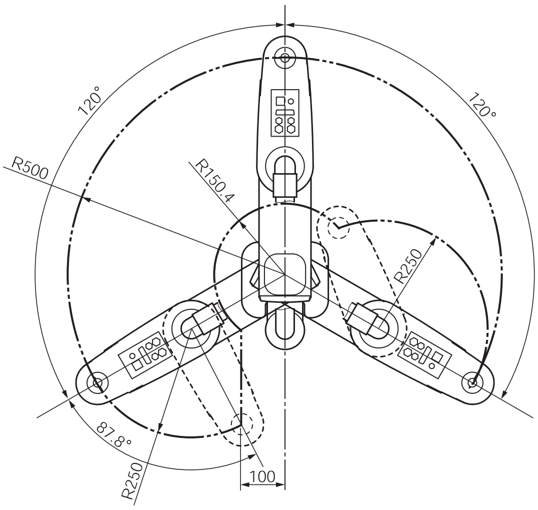
Right arm system

Right arm system : Condition where arms 1 and 2 extend straight and arm 2 is positioned in the CCW direction.
Left arm system : Condition where arms 1 and 2 extend straight and arm 2 is positioned in the CW direction.
The conditions of robot arms are expressed by assuming them as human arms.

The operation area is different between the right arm system and left arm system.
The figure below shows the operation area of each arm system of a robot whose arm length is 500mm.



Operation area of left arm system



Operation area of right arm system

(2) Arm control commands (dedicated SCARA commands)

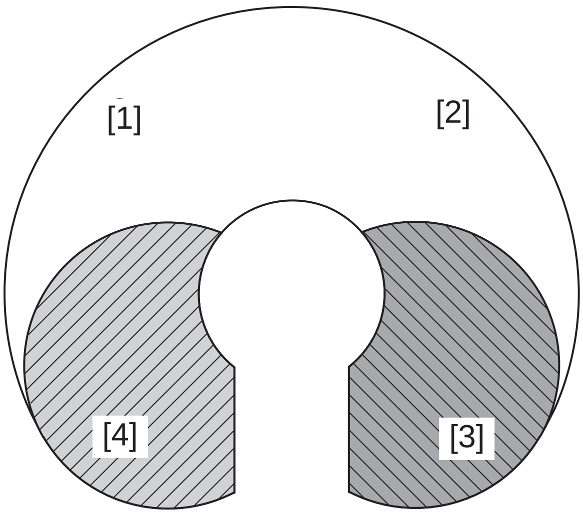
The left arm system is defined as “opposite arm system” of the right arm system, and vice versa.
The actual arm system currently used is defined as “current arm system”.
The arm system scheduled to be used for positioning to the target under a movement command is defined as “target arm system”.

Commands that are used to control the robot arm system include PTPD, PTPE, PTPR, PTPL, RIGH and LEFT.
PTPD, PTPE, PTPR and PTPL are control declarations for the target arm system of PTP operation, so they remain valid throughout the program once declared. In the case of CP operation where the arm system does not change, operation is performed based on the current arm system without being affected by the above commands.
Only one of PTPD, PTPE, PTPR and PTPL, whichever is executed last, remains valid.
RIGH and LEFT are control commands for the current arm system.

(3) Arm system control commands and arm system changes

Arm system commands and how the arm system changes during PTP operation as a result of their declaration are explained.

Set position No. 1 to 4 as shown below ([1] to [4]).
Try moving the actuator using a MOVP command (PTP operation) in the order of 1 → 2 → 3 → 2 → 1 → 4.
Move the robot while it is positioned at position No. 1.
Position No. 3 is inside the limited area for left arm system (positioning to this position is not possible if the right arm system is used).
Position No. 4 is inside the limited area for right arm system (positioning to this position is not possible if the left arm system is used).




```
:
:
:
Arm system control command
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4
EXIT
```

: Limited area for right arm system

: Limited area for left arm system

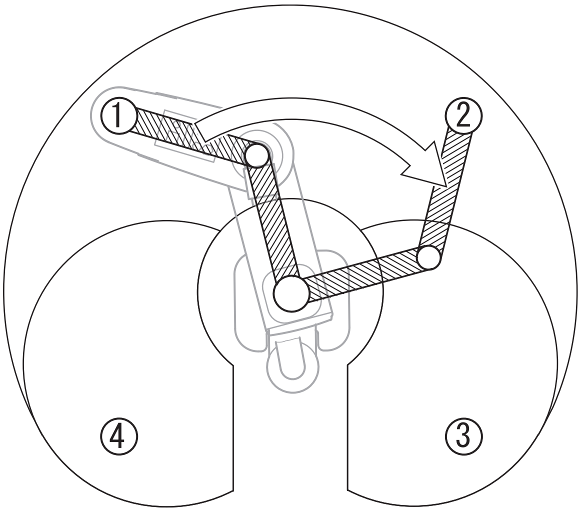
How the arm system changes with an arm system control command is explained for each command.

In the figure, the black arrows indicate movements where the arm system changes.
White arrows indicate movements where the arm system does not change.
The shaded arm represents the right arm system. The unshaded arm represents the left arm system.

[PTPD]

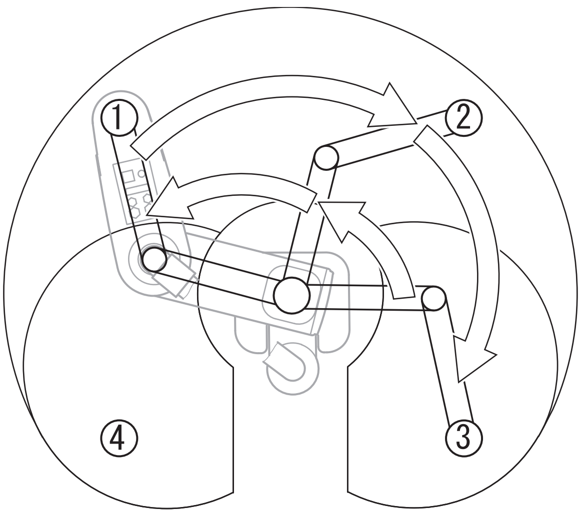
Following the execution of a PTPD command, the robot performs positioning by moving according to the current arm system. The PTPD command prohibits situations where the current arm system is opposite the target arm system. An attempt to move to an area to which positioning is impossible without changing to the opposite arm system generates an error "C73: Target path soft limit over error". Even when a PTPD command is not executed, this command is already effective on the robot when the program is started.

1) Starting from right arm system




```
:
:
:
PTPD
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

2) Starting from left arm system

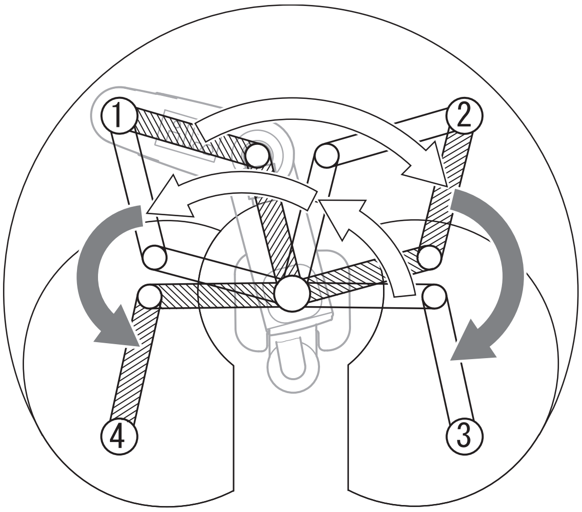



```
:
:
:
PTPD
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[PTPE]

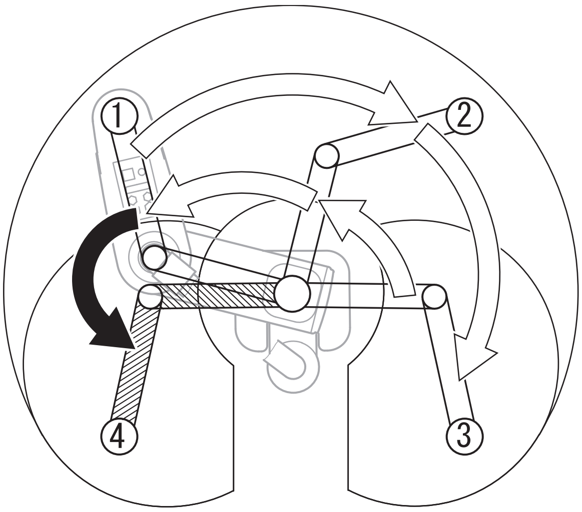
Following the execution of a PTPE command, the robot gives priority to the current arm system for movement and positioning. The PTPE command permits situations where the current arm system is opposite the target arm system. Therefore, it is permitted to move to an area to which positioning is impossible without changing to the opposite arm system.
To prohibit moving to the area for opposite arm system after permitting such movement, execute a PTPD command.

1) Starting from right arm system




```
:
:
:
PTPE
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4
EXIT
```

2) Starting from left arm system

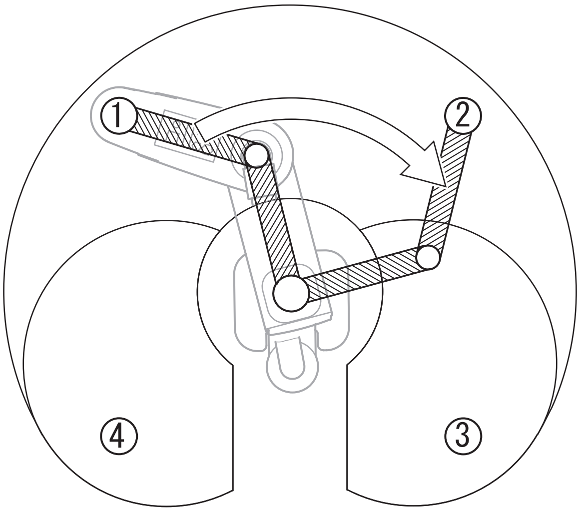



```
:
:
:
PTPE
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4
EXIT
```

[PTPR]

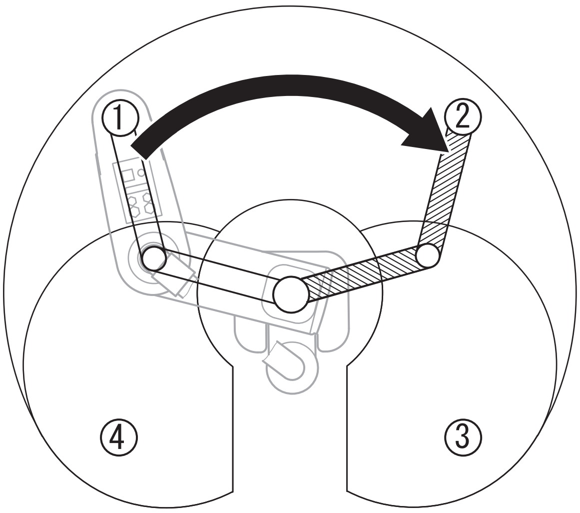
Following the execution of a PTPR command, the robot performs positioning according to the right arm system. The PTPR command limits the target arm system to the right arm system. Therefore, an attempt to move to an area to which positioning is impossible without changing to the left arm system generates a "C73: Target path soft limit over error".
Executing a PTPR command alone does not initiate any arm movement.
When a PTP movement command is executed following the execution of a PTPR command and while the current arm system is the left arm system, the robot moves as it changes from the left arm system to right arm system and performs positioning according to the right arm system.

1) Starting from right arm system



:
:
:
PTPR
MOVP 2
MOVP 3 ⇒ C73 error occurs.

2) Starting from left arm system

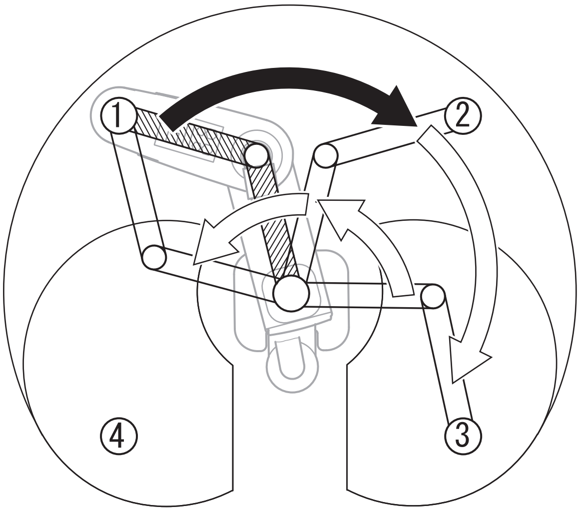


:
:
:
PTPR
MOVP 2
MOVP 3 ⇒ C73 error occurs.

[PTPL]

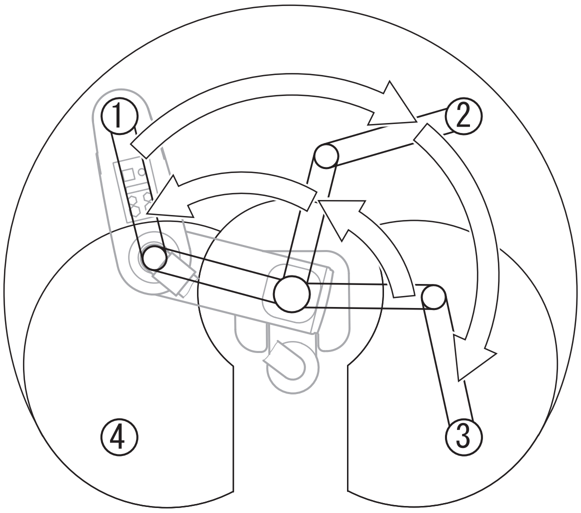
Following the execution of a PTPL command, the robot performs positioning according to the left arm system. The PTPL command limits the target arm system to the left arm system. Therefore, an attempt to move to an area to which positioning is impossible without changing to the right arm system generates a “C73: Target path soft limit over error”.
Executing a PTPL command alone does not initiate any arm movement.
When a PTP movement command is executed following the execution of a PTPL command and while the current arm system is the right arm system, the robot moves as it changes from the right arm system to left arm system and performs positioning according to the left arm system.

1) Starting from right arm system




```
:
:
:
PTPL
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

2) Starting from left arm system

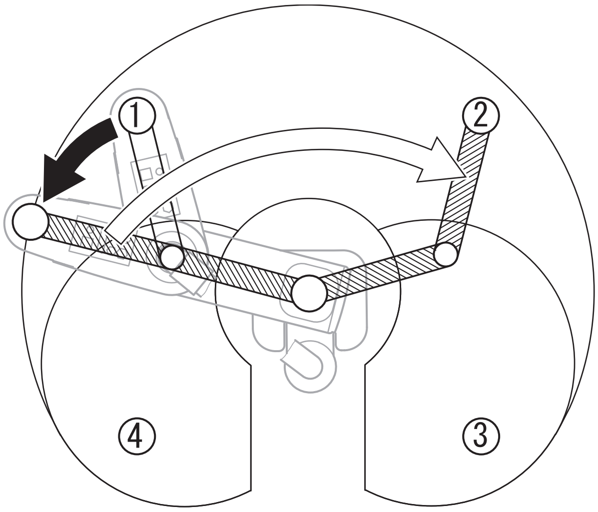



```
:
:
:
PTPL
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[RIGH]

The RIGH command changes the current arm system to the right arm system.
When a RIGH command is executed while the current arm system is the left arm system, arm 2 operates in such a way that both arms 1 and 2 form a straight line.
Executing a RIGH command while the current arm system is the right arm system does not initiate any arm movement.

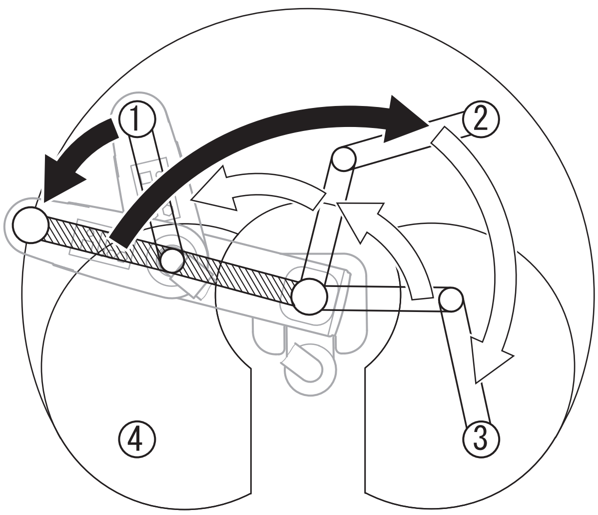
1) Starting from left arm system




```
:
:
:
RIGH
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

In the above example, the PTPD command is effective because no arm system control command other than RIGH is set.
The RIGH command only controls the current arm system. It does not limit positioning via PTP operation to the right arm system. The arm system used for positioning varies depending on the control declaration of target arm system (PTPD, PTPE, PTPR, PTPL).
Accordingly, the specific operation that takes place after the execution of a RIGH command varies depending on the control declaration of target arm system which is currently effective.

2) RIGH command at PTPL command execution

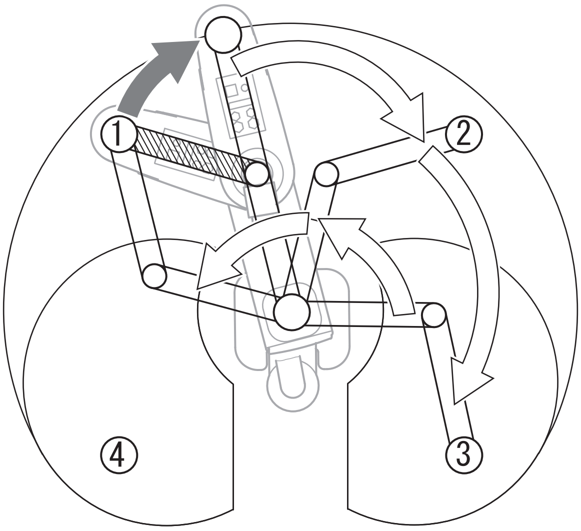



```
:
:
:
PTPL
:
:
:
RIGH
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[LEFT]

The LEFT command changes the current arm system to the left arm system.
When a LEFT command is executed while the current arm system is the right arm system, arm 2 operates in such a way that both arms 1 and 2 form a straight line.
Executing a LEFT command while the current arm system is the left arm system does not initiate any arm movement.

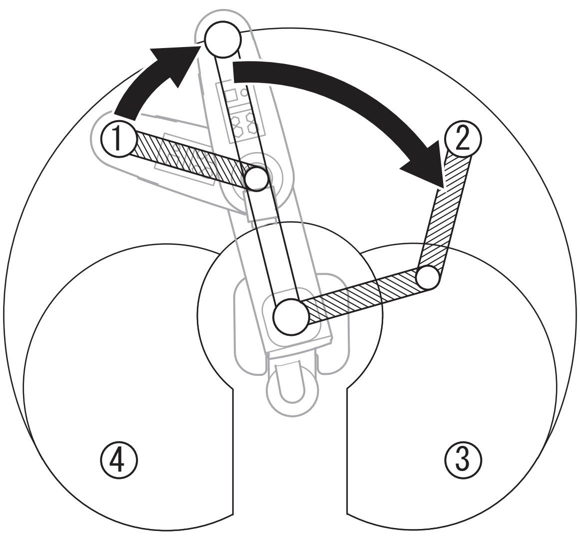
1) Starting from right arm system




```
:
:
:
LEFT
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

In the above example, the PTPD command is effective because no arm system control command other than LEFT is set.
The LEFT command only controls the current arm system. It does not limit positioning via PTP operation to the left arm system. The arm system used for positioning varies depending on the control declaration of target arm system (PTPD, PTPE, PTPR, PTPL).
Accordingly, the specific operation that takes place after the execution of a LEFT command varies depending on the control declaration of target arm system which is currently effective.

2) LEFT command at PTPR command execution




```
:
:
:
PTPR
:
:
:
LEFT
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

[4] PTP Acceleration/Deceleration Optimization Function

IX-***H Type and IXP Type operates in the optimum acceleration / deceleration speed during the PTP operation.

(Note) Those such as IX-NNN5020 do not operate in the optimum acceleration/deceleration speed during the PTP operation. For those models, the maximum acceleration/deceleration speed during the PTP operation relies on the values set in Axis Parameters No. 134 “Maximum PTP acceleration for SCARA axis” and No. 135 “Maximum PTP deceleration for SCARA axis”.
The acceleration for PTP operation corresponds to the ratio (%) set according to the ACCS command and DCLS command.

(1) Function overview

PTP optimum acceleration/deceleration is the automatic adjustment function to obtain the optimized acceleration and deceleration responding to the conditions of such facts as the tip load. In PTP optimum acceleration/deceleration, it is necessary to set the tip load mass with the WGHT Command as well as the acceleration and deceleration ratios with ACCS and DCLS Commands set for the existing models. Set an appropriate load mass according to the load to be transported, etc.
The calculation formats of PTP acceleration and deceleration in PTP optimum acceleration/deceleration are as shown below:

- PTP acceleration = Maximum acceleration determined by the load mass, etc. × ACCS command [%]
- PTP deceleration = Maximum deceleration determined by the load mass, etc. × DCLS command [%]

* The WGHT command is supported by controllers of main application Ver.0.45 or later. This command can be input in PC software of Ver.7.5.0.0 or later or on teaching pendants of Ver.1.11 or later.

⚠ Caution

- PTP optimum acceleration/deceleration would not work with the ideal acceleration or deceleration unless the setting of the WGHT Command according the actual robot tip load is conducted. Make sure to setup the tip load mass setting in the WGHT Command.
- PTP optimum acceleration/deceleration function is effective only for the PTP operation. It cannot be operated with the optimized acceleration or deceleration for CP operation or direct-movement axis.
- If an overload error occurs, lower the acceleration setting and/or deceleration setting as deemed appropriate or make other adjustment such as providing a stopping time after acceleration/deceleration to prevent an overload error from occurring.

[5] Horizontal Movement Optimizing Function responding to Z-axis position

SCARA Robot (IX-***H) can utilize the horizontal movement optimizing function.

(Note) Note that the horizontal movement Z-position optimization function is not available for those such as IX-NNN5020. (Using this function would generate a "D8A: Internal parameter error for acceleration/deceleration optimization or horizontal movement Z-position optimization function".)

(1) Function overview

Horizontal movement optimizing function by Z-axis is the function to optimize the horizontal movement condition based on Z-axis position and the tip load mass.
This function can be set effective/ineffective in the all axes parameter No. 51. When a parameter setting change is made, make sure to reset the software or reboot the system after the flash ROM writing is complete.
The tip load mass setting by the WGHT Command is required while the SCARA Z-axis position and horizontal movement optimizing function are effective. Set the load mass setting accordingly following the transporting work figure.

● All-axis common parameters

No.	Parameter name	Default value (reference)	Input range	Unit	Access privilege	Remarks
51	SCARA-axis control 1	0H	0H to FFFFFFFFH		F	Bits 8 to 11: Z-axis Position -> horizontal movement optimized (PTP) (0: Disable, 1: Enable) (Available only on high-speed SCARA robots with main application Ver.0.45 or later) Bits 12 to 15: Z-axis Position -> horizontal movement optimized (CP) (0: Disable, 1: Enable) * Disabling this function is recommended if the CP operation requires constant speed, path precision and attainment of specified speed. (Available only on high-speed SCARA robots with main application Ver.0.45 or later)

* The WGHT command is supported by controllers of main application Ver.0.45 or later. This command can be input in PC software of Ver.7.5.0.0 or later or on teaching pendants of Ver.1.11 or later.

Caution

- It is necessary to set the tip load mass with the WGHT Command while the horizontal movement optimizing function by Z position is activated. An appropriate result could not be gained unless the mass setting according the actual robot tip load is conducted.

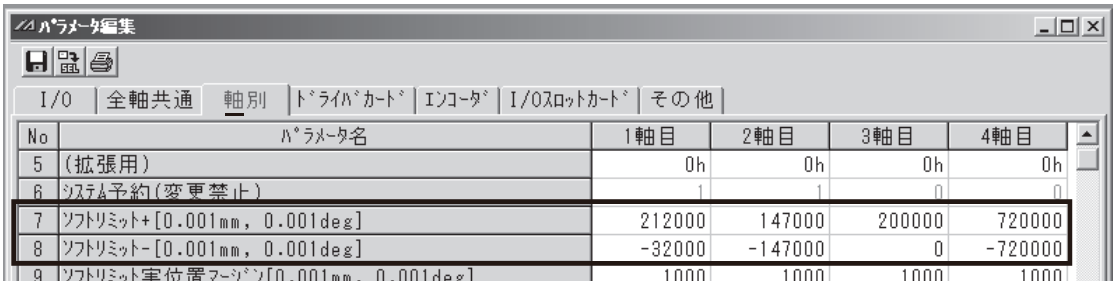
- When the horizontal movement optimizing function by Z position is activated, the speed may not reach the set speed due to the robot load mass or movement position. Make the horizontal movement optimization function invalid if it is required to reach the set speed.
  * When also it is indicated the Operation 1 = 0 (prioritized to reach set speed automatic division) for DIS (divide distance setting for spline movement) and DIG (arc angle setting), the horizontal movement optimization function should be prioritized and may not reach the set speed.

- When operating individually with the PATH, CIR, ARC, CIRS, ARCS or PSPL Command while the horizontal movement optimization (CP) by Z position is activated, the movement speed during the command may vary due to the robot load mass or the movement position. In a continuous operation with the continuous operation related commands (PATH, PSPL, CIR2, ARC2, CIRS, ARCS, CIRS, ARCD, ARCC, CIR, ARC Commands, etc.), the movement speed between the commands may vary due to the operational condition. Make the horizontal movement optimization (CP) invalid if an evenly paced speed is required.

- When the horizontal movement optimization (CP) by Z axis is activated, the track of CP operation may slightly vary due to the robot load mass and movement position.
  If accuracy in the track is required, make the horizontal movement optimization (CP) invalid.

[6] Soft Limit

The soft limit is set in axis-specific parameter No. 7 and 8. Below is an example of a screen showing the soft limits for IX5020 (arm length 500mm, Z-axis 200mm).


ﾊﾟﾗﾒｰﾀ編集

I/O | 全軸共通 | 軸別 | ﾄﾞﾗｲﾊﾞｶｰﾄﾞ | ｴﾝｺｰﾀﾞ | I/Oｽﾛｯﾄｶｰﾄﾞ | その他

No	ﾊﾟﾗﾒｰﾀ名	1軸目	2軸目	3軸目	4軸目
5	(拡張用)	0h	0h	0h	0h
6	ｼｽﾃﾑ予約(変更禁止)	1	1	0	0
7	ｿﾌﾄﾘﾐｯﾄ+[0.001mm, 0.001deg]	212000	147000	200000	720000
8	ｿﾌﾄﾘﾐｯﾄ-[0.001mm, 0.001deg]	-32000	-147000	0	-720000
9	ｿﾌﾄﾘﾐｯﾄ実位置ﾏｰｼﾞﾝ[0.001mm, 0.001deg]	1000	1000	1000	1000

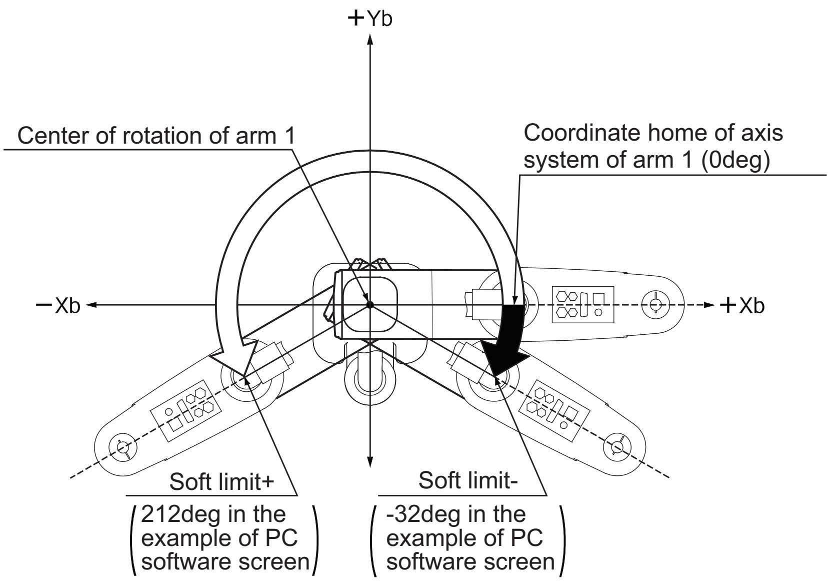
The soft limit parameters are set by coordinate values according to each axis system.
Axis 1 corresponds to arm 1, axis 2 corresponds to arm 2, axis 3 corresponds to Z-axis, and axis 4 corresponds to R-axis.
The setting units is [0.001deg] for arm 1, arm 2 and R-axis (rotational axis). The setting unit for Z-axis is [0.001mm].
The soft limits are used to limit the range of operation of arm 1, arm 2, Z-axis or R-axis from the coordinate home of each axis system. It is not affected by the work coordinates system or tool coordinate systems.

(Note) These parameters have been set to the maximum limits of range of operation at the factory. Do not set values that would enlarge the range of operation.

(1) Coordinates on each axis system and soft limits

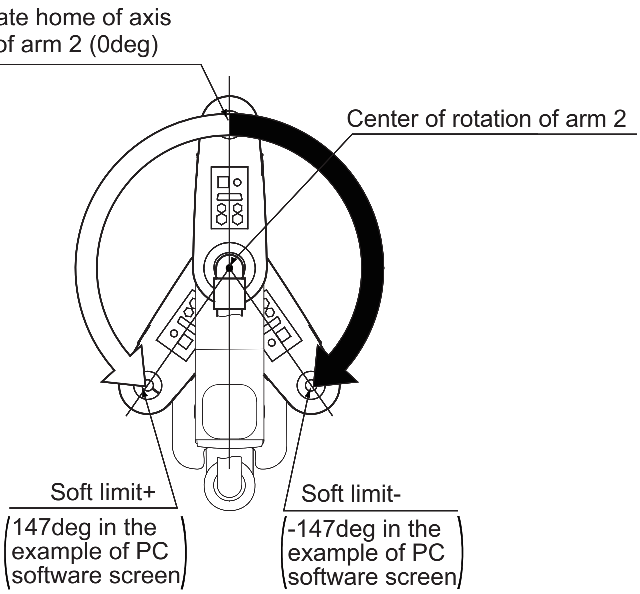
[Soft limits for arm 1]

The arm 1 position at which the arm faces the +Xb direction defines the coordinate home of the axis system of arm 1 (0deg).
This position is not affected by the arm 2 position.
Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 1 of axis-specific parameter No. 7).
Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 1 of axis-specific parameter No. 8).



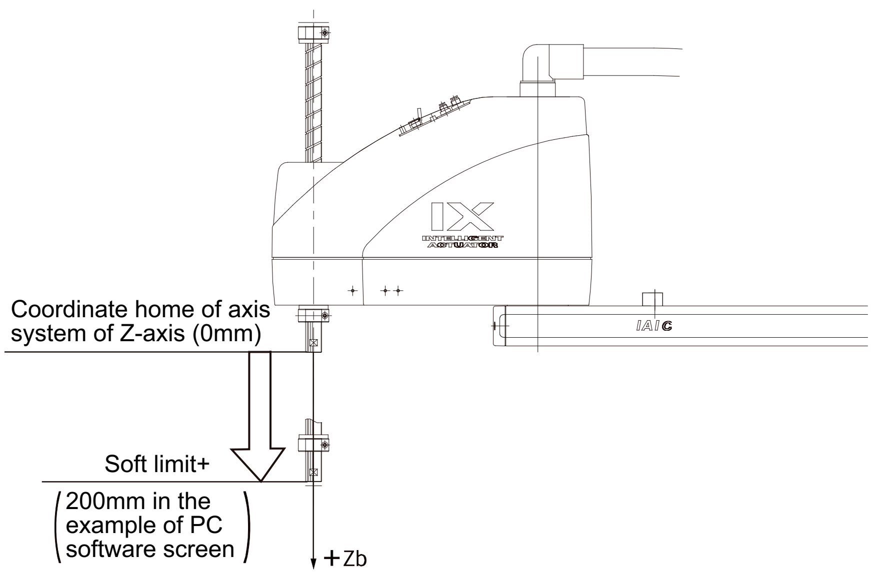
[Soft limits for arm 2]

The arm 2 position at which the arm forms a straight line with arm 1 defines the coordinate home of the axis system of arm 2 (0deg).
This position is not affected by the arm 1 position.
Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 2 of axis-specific parameter No. 7).
Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 2 of axis-specific parameter No. 8).



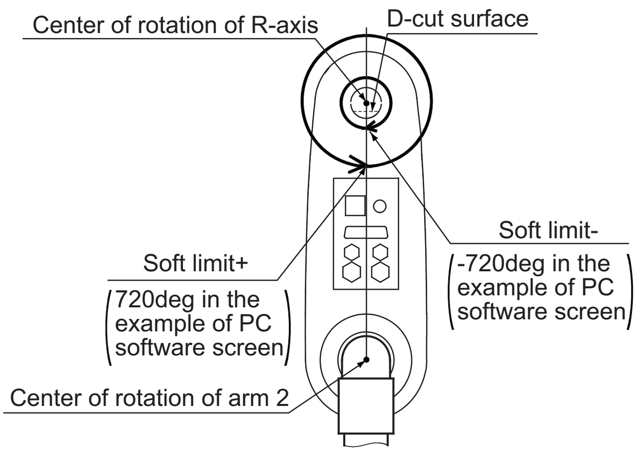
[Soft limits for Z-axis]

The Z-axis position at which the mechanical stopper attached to the Z-axis is approx. 5mm below the mechanical end at the bottom of arm 2 defines the coordinate home of the axis system of Z-axis (0mm). This position is the same as the Axis 3 = 0mm position on the base coordinate system. (On actuators of clean-room specification and dust-proof/splash-proof specification, this mechanical stopper is not visible because it is located inside bellows.)
Movements in the downward direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 3 of axis-specific parameter No. 7). Movements in the upward direction (negative direction) from this coordinate home of axis system are limited by the soft limit- (axis 3 of axis-specific parameter No. 8). (The relationship is reversed on actuators of inverse specification.)



[Soft limits for R-axis]

The R-axis position at which the D-cut surface at the tip of the axis faces the center of rotation of arm 2 defines the coordinate home of the axis system of R-axis (0deg). This position is not affected by the arm 1 or arm 2 position.



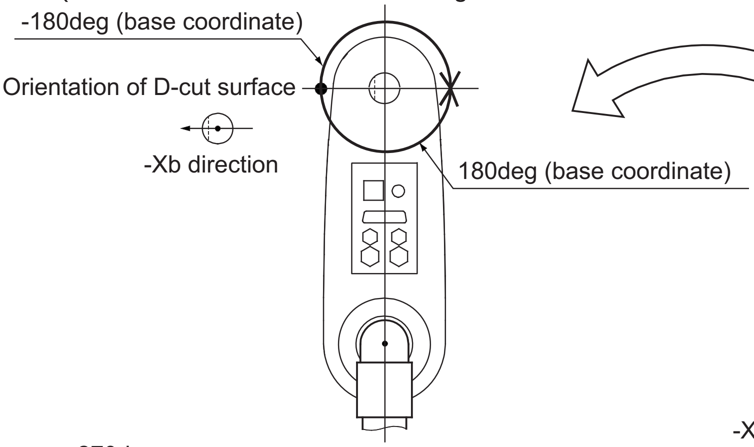
Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 4 of axis-specific parameter No. 7). Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 4 of axis-specific parameter No. 8).

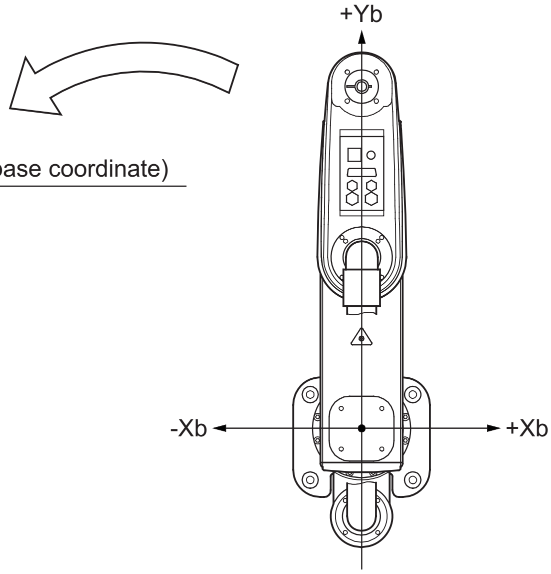
When limiting the operating range of the R-axis, you must pay attention to the difference between the base coordinate system and this axis system.

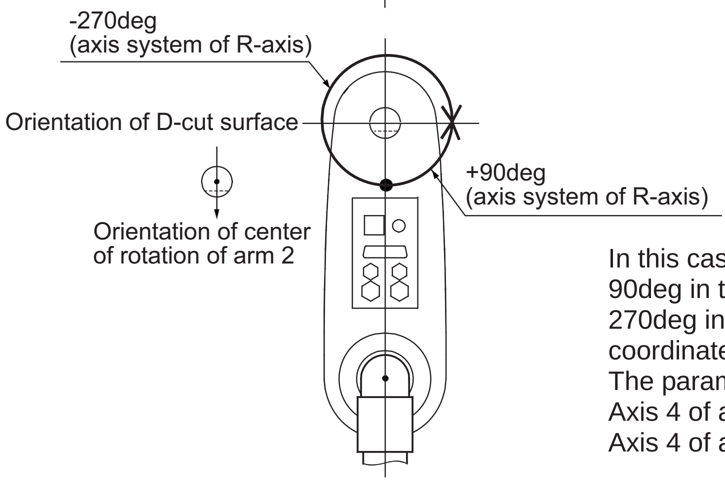
(Example)

Limit the range of operation of the R-axis to ±180 from the position shown below.

(Limit the R-axis to within ±180deg in the Axis 4 = 0 condition on the base coordinate system.)







In this case, the R-axis is limited to within 90deg in the positive direction and within 270deg in the negative direction from the coordinate home of the axis system of R-axis.

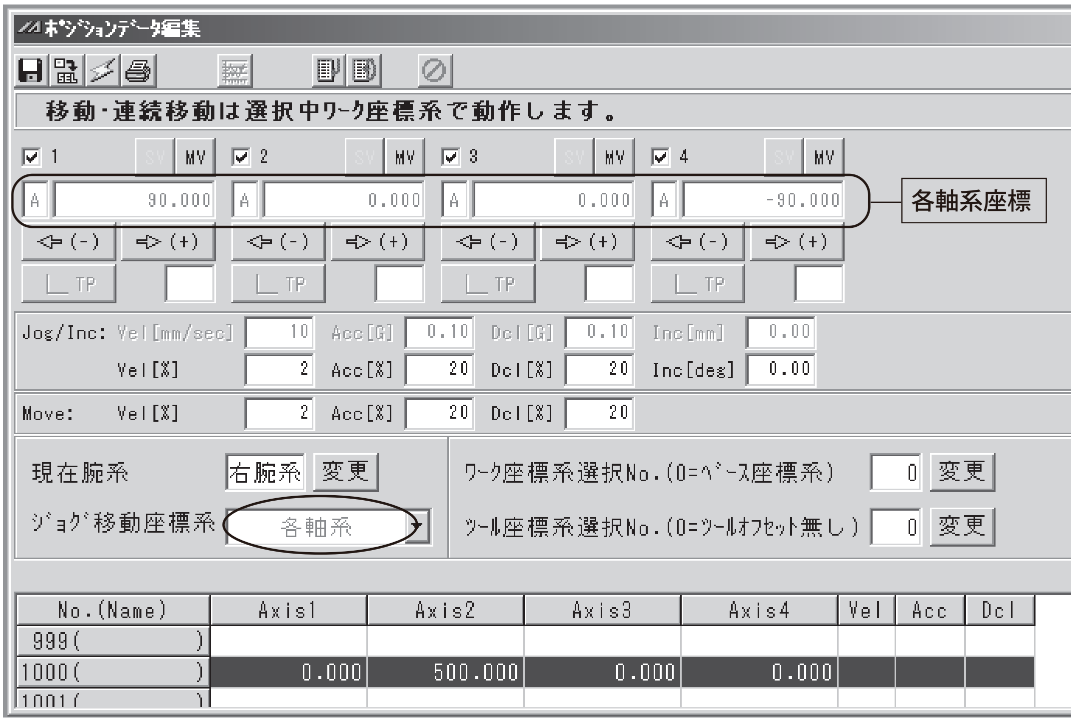
The parameters below are set as follows:

Axis 4 of axis-specific parameter No. 7 = 90000

Axis 4 of axis-specific parameter No. 8 = -270000

(2) Monitoring of axis system coordinates

You can use the PC software or teaching pendant to monitor axis system coordinates. Shown below is an example of a PC software screen. When the jog movement coordinate system is selected for each axis system in the position data edit window, the currently displayed position switch to a coordinate based on the selected axis system.



(An IX5020 (arm length 500mm, Z-axis 200mm) is located at the position of Axis 1 = 0, Axis 2 = 500, Axis 3 = 0, Axis 4 = 0 on the base coordinate system.)

(Note) Position data cannot be loaded in each axis system.

[For details on the specific operating procedure, refer to the Instruction Manual for your PC software or teaching pendant.]

[7] Simple Contact Check Zone

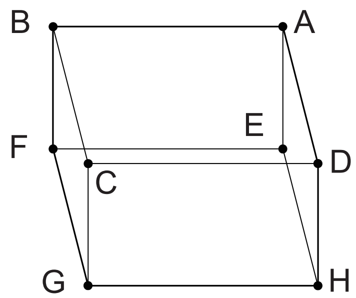
The simple contact check zone is an area you must set when checking for contact between the robot and nearby equipment.
When tool coordinate system No. 0 (= tool coordinate system offset 0) is selected, you can detect an entry into the simple contact check zone by the center position of the tool mounting surface. When any one of tool coordinate system No. 1 to 127 (= tool coordinate system offset enabled) is selected, you can detect a similar entry by the tool tip position.

[Notes on use of simple contact check zone]

- An entry into the simple contact check zone by the center position of the tool mounting surface (when tool coordinate system No. 0 is selected) or tool tip position (when any one of tool coordinate system No. 1 to 127 is selected) is detected. An entry by the outer periphery of the R-axis or any part of the tool other than its tip is not detected.
- This function does not prevent an entry into the simple contact check zone. It only detects an entry after it has occurred.
- An entry cannot be detected reliably unless the applicable position remains inside the simple contact check zone for 5msec or more. This function is intended to provide a means for simple check by low-speed operation.
- The path changes between high-speed operation (actual operation) and low-speed operation. Ensure a sufficient margin to avoid contact. (The robot tends to pass on the inner side of the path during high-speed operation compared to low-speed operation.)
- The coordinates defining the simple contact check zone are always recognized as data of the base coordinate system (work coordinate system selection No. 0). Take note that changing the work coordinate system does not change the position of the simple contact check zone.
  If the coordinates defining the simple contact check zone are changed, it will take 5msec before the check result according to the new coordinates is reflected.
- In PTP operation, the robot does not move along a fixed path. Conduct test operation at low speed to confirm absence of contact near an obstacle (including a part of the robot), and then gradually raise the speed to an appropriate level.
- If physical output port numbers or global flag numbers are duplicated, chattering occurs and operation results become unstable. Do not specify duplicate numbers.
- Use of the simple contact check zone consumes significant CPU power. When this function is not used, disable the function by setting 0 for the applicable “physical output port number/global flag number” and “error type”.
- The simple interference check zone becomes available after the home-operation complete or the absolute coordinate confirmation. Note that interference check cannot be held when home-return operation is incomplete or the absolute coordinate is unconfirmed.
- In Physical Output Port Number and Global Flag Number of Simple Interference Check Zone Definition, numbers to duplicate with those set in System Output Port / Flag Numbers (Output Function Select, Linear Axis Zone, etc.) Error No. 906 “Input and Output Port / Flag Number Error” generates if any duplicated number is indicated.

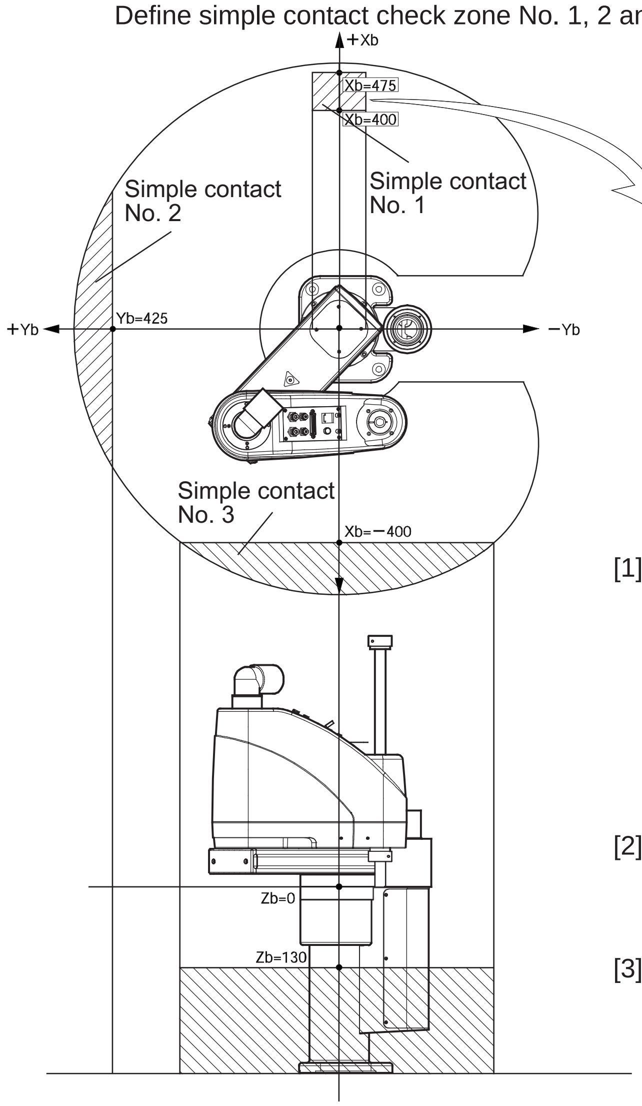
[Setting of simple contact check zone]

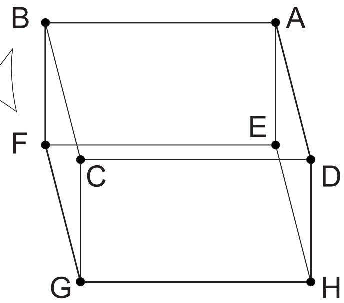
Set the simple contact check zone using position data of the base coordinate system.
Enter the maximum and minimum coordinate values of the simple contact check zone.
Set the boundary surfaces of the simple contact check zone in parallel with the base coordinate axes.



To set a rectangular solid like the one shown to the left as the simple contact check zone, enter the coordinate values of two points according to a combination of A-G, B-H, C-E or D-F.

(Example) Setting example of simple contact check zones
Define simple contact check zone No. 1, 2 and 3 as shown below.



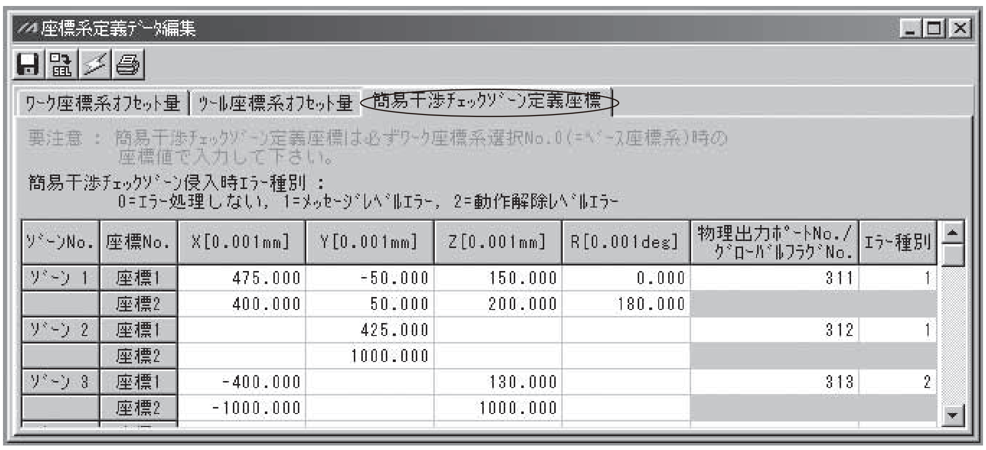


[1] For simple contact check zone No. 1, set a rectangular solid as the simple contact check zone.
A : Xb = 475, Yb = -50, Zb = 150, Rb = 0
G : Xb = 400, Yb = 50, Zb = 200, Rb = 180
An entry into this rectangular solid is not detected if Rb is outside the range of 0 to 180°.

[2] For simple contact check zone No. 2, set an area where Yb is 425mm or more as the simple contact check zone.

[3] For simple contact check zone No. 3, set an area where Xb is -400mm or less and Zb is 130mm or more as the simple contact check zone.

Shown below is the screenshot of the edit window for the simple contact check zone definition data in PC software dedicated for SCARA Robot assuming the simple contact check zones No. 1, No. 2 and No. 3 are set.


座標系定義ﾃﾞｰﾀ編集

ﾜｰｸ座標系ｵﾌｾｯﾄ量 | ﾂｰﾙ座標系ｵﾌｾｯﾄ量 | 簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

要注意 ： 簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標は必ずﾜｰｸ座標系選択No.0(=ﾍﾞｰｽ座標系)時の座標値で入力して下さい。

簡易干渉ﾁｪｯｸｿﾞｰﾝ侵入時ｴﾗｰ種別 ：
0=ｴﾗｰ処理しない, 1=ﾒｯｾｰｼﾞﾚﾍﾞﾙｴﾗｰ, 2=動作解除ﾚﾍﾞﾙｴﾗｰ

ｿﾞｰﾝNo.	座標No.	X[0.001mm]	Y[0.001mm]	Z[0.001mm]	R[0.001deg]	物理出力ﾎﾟｰﾄNo./ｸﾞﾛｰﾊﾞﾙﾌﾗｸﾞNo.	ｴﾗｰ種別
ｿﾞｰﾝ 1	座標1	475.000	-50.000	150.000	0.000	311	1
	座標2	400.000	50.000	200.000	180.000		
ｿﾞｰﾝ 2	座標1		425.000			312	1
	座標2		1000.000				
ｿﾞｰﾝ 3	座標1	-400.000		130.000		313	2
	座標2	-1000.000		1000.000			

- As for simple contact check zone No. 1, an entry into this rectangular solid is not detected if the Rb is outside the range of 0 to 180°. To detect an entry into this zone regardless of the R-axis coordinate value, leave the coordinate 1 and 2 fields for zone 1 and R blank.

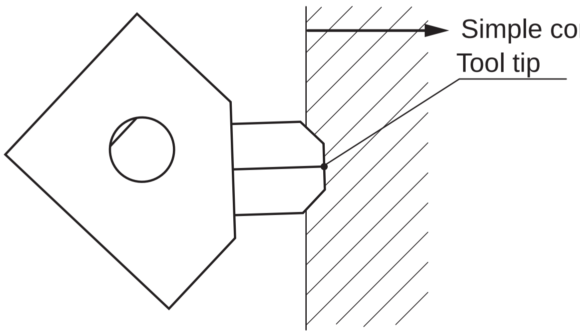
- If either the maximum value or minimum value is not limited, as is the case with simple contact check zone No. 2 and 3, enter a value outside the range of operation (such as 1000 for zone 2, and 1000 or -1000 for zone 3).

- The maximum value and minimum value can be set under either coordinate 1 or 2.

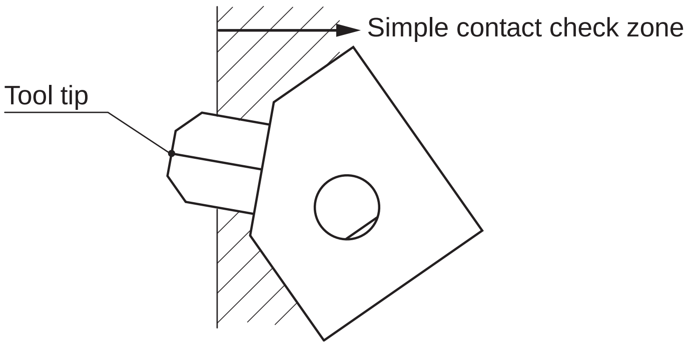
- According to the above settings, output port No. 311 turns ON upon entry into simple contact check zone No. 1, output port No. 312 turns ON upon entry into simple contact check zone No. 2, and output port No. 313 turns ON upon entry into simple contact check zone No. 3.

* Use a DFIF command if you want to set a simple contact check zone within the SEL program.

[Note on detection while tool coordinate system is selected]
While the tool coordinate system is selected, this function detects an entry of the tool tip, not the center of the mounting surface, into the simple contact check zone.



Depending on the movement path, a part of the tool other than its tip may enter the simple contact check zone, as shown below. Exercise due caution because in this case, the entry will not be detected until the tool tip enters the simple contact check zone.



● Caution

In X-SEL-RXD/SXD, the definitions of SCARA axes (Axes 1 to 4) are to be set to Axes 1 to 4 and SCARA axes (Axes 5 to 8) to Axes 5 to 8. SCARA axes (Axes 1 to 4) and SCARA axes (Axes 5 to 8) cannot be set in one zone number at the same time. (10 zone definitions are required in total for 2 units of SCARA.)

Each coordinate axis number expresses the meaning listed below.

- •Axis 1: Interference domain data of X-axis for SCARA axes (Axes 1 to 4)
- •Axis 2: Interference domain data of Y-axis for SCARA axes (Axes 1 to 4)
- •Axis 3: Interference domain data of A-axis for SCARA axes (Axes 1 to 4)
- •Axis 4: Interference domain data of R-axis for SCARA axes (Axes 1 to 4)
- •Axis 5: Interference domain data of X-axis for SCARA axes (Axes 5 to 8)
- •Axis 6: Interference domain data of Y-axis for SCARA axes (Axes 5 to 8)
- •Axis 7: Interference domain data of Z-axis for SCARA axes (Axes 5 to 8)
- •Axis 8: Interference domain data of R-axis for SCARA axes (Axes 5 to 8)

2. Connection with Host System

When transferring the data between the host system (PLC, etc.), it can be selected from the following methods(Note 1):

1) Use 24V DC I/O.
2) (For XSEL only) Use the serial communication (RS232C).
3) Use the Fieldbus communication(Note 2) (option). … This is able to control like I/O.

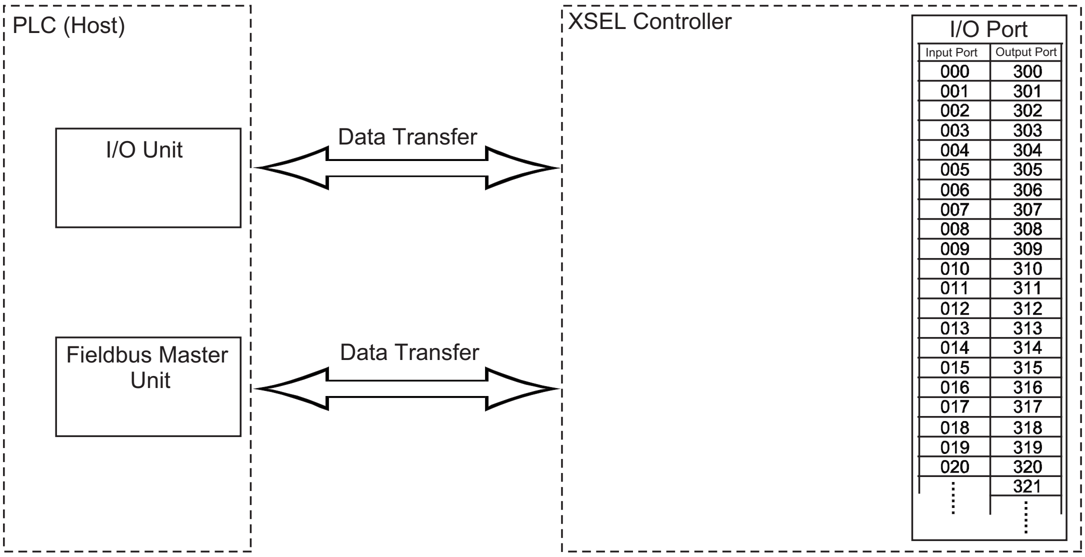
(Note 1) It includes the optional functions.

(Note 2) A dedicated PCB is required separately.
The types of applicable Fieldbus are CC-Link, DeviceNet, PROFIBUS, Ethernet (for XSEL only), EtherNet/IP (for XSEL-R*/S* only) and EtherCAT (for XSEL-R*/S* only).
[For details, refer to the Fieldbus Instruction Manual provided separately and the Instruction Manual for the host system.]

2.1 I/O Signal

There are 2 types of input and output signals as shown below.

1) Input and Output I/O Port
2) Virtual I/O Port



	Port No.	Function		Port No.	Function
Input	000	Program Start	Output	300	Alarm Output
	001	General-purpose Input		301	Ready Output
	002	General-purpose Input		302	Emergency Stop Output
	003	General-purpose Input		303	General-purpose Output
	004	General-purpose Input		304	General-purpose Output
	005	General-purpose Input		305	General-purpose Output
	006	General-purpose Input		306	General-purpose Output
	007	Program Specification (PRG No.1)		307	General-purpose Output
	008	Program Specification (PRG No.2)		308	General-purpose Output
	009	Program Specification (PRG No.4)		309	General-purpose Output
	010	Program Specification (PRG No.8)		310	General-purpose Output
	011	Program Specification (PRG No.10)		311	General-purpose Output
	012	Program Specification (PRG No.20)		312	General-purpose Output
	013	Program Specification (PRG No.40)		313	General-purpose Output
	014	General-purpose Input		314	General-purpose Output
	015	General-purpose Input		315	General-purpose Output
	⋮	⋮		⋮	⋮

(Note) The numbers of I/O ports are:

Input: 000 to 299 (300 points max.)
Output: 300 to 599 (300 points max.)

2.1.1 XSEL-J/K Type Controllers

XSEL-J/K type controllers
- XSEL-J/K/KE/KT/KET
- XSEL-JX/KX/KETX

[1] Input and Output I/O Port
With XSEL-J/K type controllers, the assignments of input and output functions to I/O ports are fixed and cannot be changed.

I/O Signal Table

Input

Pin No.	Wire color	Port No.	Standard (factory) setting Can be changed by I/O parameter	I/O parameter	
1	Brown-1		K, KX types : Cannot be connected. J, JX types : +24V input		
2	Red-1	000	Program start	No. 30	0: General-purpose input 1: Program start (BCD specification) 2: Program start (binary specification)
3	Orange-1	001	General-purpose input	No. 31	0: General-purpose input 1: Soft reset signal
4	Yellow-1	002	General-purpose input	No. 32	0: General-purpose input 1: Servo ON signal
5	Green-1	003	General-purpose input	No. 33	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
6	Blue-1	004	General-purpose input	No. 34	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
7	Purple-1	005	General-purpose input	No. 35	0: General-purpose input 1: Operation pause cancellation input (ON edge)
8	Gray-1	006	General-purpose input	No. 36	0: General-purpose input 1: Operation pause signal (OFF level)
9	White-1	007	Program number specification (MSB)	No. 37	0: General-purpose input 1: Program number specification (MSB)
10	Black-1	008	Program number specification (bit 2)	No. 38	0: General-purpose input 1: Program number specification (bit 2)
11	Brown-2	009	Program number specification (bit 3)	No. 39	0: General-purpose input 1: Program number specification (bit 3)
12	Red-2	010	Program number specification (bit 4)	No. 40	0: General-purpose input 1: Program number specification (bit 4)
13	Orange-2	011	Program number specification (bit 5)	No. 41	0: General-purpose input 1: Program number specification (bit 5)
14	Yellow-2	012	Program number specification (bit 6)	No. 42	0: General-purpose input 1: Program number specification (bit 6)
15	Green-2	013	Program number specification (LSB: bit 7)	No. 43	0: General-purpose input 1: Program number specification (LSB: bit 7)
16	Blue-2	014	General-purpose input	No. 44	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
17	Purple-2	015	General-purpose input	No. 45	0: General-purpose input The following settings are effective only with XSEL-J/K: 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)
18	Gray-2	016	General-purpose input		
19	White-2	017	General-purpose input		
20	Black-2	018	General-purpose input		
21	Brown-3	019	General-purpose input		
22	Red-3	020	General-purpose input		
23	Orange-3	021	General-purpose input		
24	Yellow-3	022	General-purpose input		
25	Green-3	023	General-purpose input		
26	Blue-3	024	General-purpose input		
27	Purple-3	025	General-purpose input		
28	Gray-3	026	General-purpose input		
29	White-3	027	General-purpose input		
30	Black-3	028	General-purpose input		
31	Brown-4	029	General-purpose input		
32	Red-4	030	General-purpose input		
33	Orange-4	031	General-purpose input		

Output

Pin No.	Wire color	Port No.	Standard (factory) setting Can be changed by I/O parameter	I/O parameter	
34	Yellow-4	300	Output of operation-cancellation level or higher error (OFF)	No. 46	0: General-purpose output 1: Output of operation-cancellation level or higher error (ON) 2: Output of operation-cancellation level or higher error (OFF) 3: Output of operation-cancellation level or higher error + Emergency stop output (ON) 4: Output of operation-cancellation level or higher error + Emergency stop output (OFF)
35	Green-4	301	READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) (Main application version 0.20 or later)	No. 47	0: General-purpose output 1: READY output (PIO-trigger program operation enabled) 2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) (Main application Ver.0.20 or later) 3: READY output (PIO-trigger program operation enabled and no cold-start level or higher error) (Main application Ver.0.20 or later)
36	Blue-4	302	Emergency stop output (OFF)	No. 48	0: General-purpose output 2: Emergency stop output (ON) 3: Emergency stop output (OFF)
37	Purple-4	303	General-purpose output	No. 49	0: General-purpose output 1: AUTO mode output 2: Auto operation output (When other parameter No. 12 is set to ‘1’)
38	Gray-4	304	General-purpose output	No. 50	0: General-purpose output The following settings are effective only with XSEL-J/K: 1: Output when all effective axes are home (= 0) 2: Output when all effective axes have completed home return 3: Output when all effective axes are at home preset coordinate (Main application Ver.0.21 or later) * To move an actuator of absolute encoder specification to coordinate 0 or the home preset coordinate, use a MOVP command instead of HOME command.
39	White-4	305	General-purpose output	No. 51	0: General-purpose output 2: Axis 1 servo ON output (Main application Ver.0.44 or later)
40	Black-4	306	General-purpose output	No. 52	0: General-purpose output 2: Axis 2 servo ON output (Main application Ver.0.44 or later)
41	Brown-5	307	General-purpose output	No. 53	0: General-purpose output 2: Axis 3 servo ON output (Main application Ver.0.44 or later)
42	Red-5	308	General-purpose output	No. 54	0: General-purpose output 2: Axis 4 servo ON output (Main application Ver.0.44 or later)
43	Orange-5	309	General-purpose output	No. 55	
44	Yellow-5	310	General-purpose output	No. 56	
45	Green-5	311	General-purpose output	No. 57	
46	Blue-5	312	General-purpose output	No. 58	
47	Purple-5	313	General-purpose output	No. 59	0: General-purpose output 1: System-memory backup battery voltage low alarm level or lower
48	Gray-5	314	General-purpose output	No. 60	0: General-purpose output 1: Absolute-battery backup battery voltage low alarm level or lower (OR check of all axes. If an error level is detected, this output is retained until power-ON reset or software reset.) (Main application Ver.0.28 or later)
49	White-5	315	General-purpose output	No. 61	
50	Black-5		K, KX types : Need not be connected. J, JX types : 0V input		

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.

[2] Virtual I/O Ports

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-J/K Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	Voltage low warning for system-memory backup battery
7003	Abnormal voltage of system-memory backup battery
7004	(For future expansion = Use strictly prohibited)
7005	(For future expansion = Use strictly prohibited)
7006	Top-level system error = Message level error is present
7007	Top-level system error = Operation-cancellation level error is present
7008	Top-level system error = Cold-start level error is present
7009	(For future expansion = Use strictly prohibited)
7010	Drive-source cutoff factor is present (including when waiting for cutoff reset input)
7011	Latch signal indicating that all-operation-cancellation factor is present (latch signal for recognizing 1-shot cancellation factor; latch is cancelled by 7300-ON)
7012	All-operation-pause factor is present (including when waiting for restart switch signal) (Valid only during automatic operation recognition)
7013	All-servo-axis-interlock factor is present (all-operation-pause factor + interlock input-port factor)
7014	(For future expansion = Use strictly prohibited)
7015	Voltage low warning for axis-1 absolute-data backup battery (main application version 0.28 or later)
7016	Abnormal voltage of axis-1 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later)
7017	Voltage low warning for axis-2 absolute-data backup battery (main application version 0.28 or later)
7018	Abnormal voltage of axis-2 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later)
7019	Voltage low warning for axis-3 absolute-data backup battery (main application version 0.28 or later)
7020	Abnormal voltage of axis-3 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later)
7021	Voltage low warning for axis-4 absolute-data backup battery (main application version 0.28 or later)
7022	Abnormal voltage of axis-4 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later)
7023 to 7030	For future expansion = Use strictly prohibited
7031	Reading SIO CH1 (standard SIO) (reception ready) (*OFF if used for PC/TP connection) (main application versions 0.41 or later)
7032	Reading SIO CH2 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7033	Reading SIO CH3 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7034	Reading SIO CH4 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7035	Reading SIO CH5 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7036	Reading SIO CH6 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7037	Reading SIO CH7 (expanded SIO) (reception ready) (main application versions 0.41 or later)
7038 to 7040	(For future expansion = Use strictly prohibited)
7041 to 7070	(For future expansion = Use strictly prohibited)
7071	In AUTO mode (main application version 0.87 or later)
7072	During automatic operation (main application version 0.87 or later)
7073 to 7100	(For future expansion = Use strictly prohibited)
7101	Running program No. 01 (including during pause)
~	~
7164	Running program No. 64 (including during pause)
7165 to 7299	(For future expansion = Use strictly prohibited)

XSEL-J/K Virtual Output Ports (Internal Flags)

Port No.	Function
7300	Latch cancellation output for a latch signal indicating that all-operation-cancellation factor is present (7011) (latch is cancelled only when operation-cancellation factor is no longer present) (7300 will be turned OFF following an attempt to cancel latch.)
7301 to 7380	(For future expansion = Use strictly prohibited)
7381 to 7399	(For future expansion = Use strictly prohibited)
7400 to 7599	(For future expansion = Use strictly prohibited)

XSEL-JX/KX Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	System-memory backup battery voltage low warning
7003	System-memory backup battery voltage error
7004	(Reserved by the system = Use is strictly prohibited)
7005	(Reserved by the system = Use is strictly prohibited)
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	(Reserved by the system = Use is strictly prohibited)
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014	(Reserved by the system = Use is strictly prohibited)
7015	Axis 1 absolute-data backup battery voltage low warning
7016	Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7017	Axis 2 absolute-data backup battery voltage low warning
7018	Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7019	Axis 3 absolute-data backup battery voltage low warning
7020	Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7021	Axis 4 absolute-data backup battery voltage low warning
7022	Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7023 to 7030	(For future expansion = Use is strictly prohibited)
7031	Reading SIO CH1 (standard SIO) (Receive ready) (*OFF if a PC/TP is connected)
7032	Reading SIO CH2 (standard SIO) (Receive ready)
7033	Reading SIO CH3 (standard SIO) (Receive ready)
7034	Reading SIO CH4 (standard SIO) (Receive ready)
7035	Reading SIO CH5 (standard SIO) (Receive ready)
7036	Reading SIO CH6 (standard SIO) (Receive ready)
7037	Reading SIO CH7 (standard SIO) (Receive ready)
7038 to 7070	(Reserved by the system = Use is strictly prohibited)
7071	In the AUTO mode (Main application Ver.0.34 or later)
7072	During auto operation (Main application Ver.0.34 or later)
7073 to 7100	(Reserved by the system = Use is strictly prohibited)
7101	Program No. 01 is being executed (or paused).
~	~
7164	Program No. 64 is being executed (or paused).
7165 to 7299	(For future expansion = Use is strictly prohibited)

XSEL-JX/KX Virtual Output Ports (Internal Flags)

Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	(For future expansion = Use is strictly prohibited)
7381 to 7399	(Reserved by the system = Use is strictly prohibited)
7400 to 7599	(For future expansion = Use is strictly prohibited)

2.1.2 XSEL-P/Q/PCT/QCT Controllers

[1] Input and Output I/O Port

With XSEL-P/Q/PCT/QCT controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

Pin No.	Wire color	Port No.	Standard (factory-set) function
1	Brown-1		+24V input
2	Red-1	000	Program start
3	Orange-1	001	General-purpose input
4	Yellow-1	002	General-purpose input
5	Green-1	003	General-purpose input
6	Blue-1	004	General-purpose input
7	Purple-1	005	General-purpose input
8	Gray-1	006	General-purpose input
9	White-1	007	Program number specification (MSB)
10	Black-1	008	Program number specification (bit 2)
11	Brown-2	009	Program number specification (bit 3)
12	Red-2	010	Program number specification (bit 4)
13	Orange-2	011	Program number specification (bit 5)
14	Yellow-2	012	Program number specification (bit 6)
15	Green-2	013	Program number specification (LSB: bit 7)
16	Blue-2	014	General-purpose input
17	Purple-2	015	General-purpose input
18	Gray-2	016	General-purpose input
19	White-2	017	General-purpose input
20	Black-2	018	General-purpose input
21	Brown-3	019	General-purpose input
22	Red-3	020	General-purpose input
23	Orange-3	021	General-purpose input
24	Yellow-3	022	General-purpose input
25	Green-3	023	General-purpose input
26	Blue-3	024	General-purpose input
27	Purple-3	025	General-purpose input
28	Gray-3	026	General-purpose input
29	White-3	027	General-purpose input
30	Black-3	028	General-purpose input
31	Brown-4	029	General-purpose input
32	Red-4	030	General-purpose input
33	Orange-4	031	General-purpose input

Remarks		
Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters.		
Parameter No.	Parameter name	Function
No. 30	Input function selection 000	0: General-purpose input 1: Program start (input ports 007 to 013, BCD specification) 2: Program start (input ports 007 to 013, binary specification) 3: Program start (input ports 008 to 014, BCD specification) 4: Program start (input ports 008 to 014, binary specification)
No. 31	Input function selection 001	0: General-purpose input 1: Soft reset signal
No. 32	Input function selection 002	0: General-purpose input 1: Servo ON signal
No. 33	Input function selection 003	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
No. 34	Input function selection 004	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
No. 35	Input function selection 005	0: General-purpose input 1: Operation pause cancellation input (ON edge)
No. 36	Input function selection 006	0: General-purpose input 1: Operation pause signal (OFF level)
No. 37	Input function selection 007	0: General-purpose input 1: Program number specification (MSB)
No. 38	Input function selection 008	0: General-purpose input 1: Program number specification (bit 2)
No. 39	Input function selection 009	0: General-purpose input 1: Program number specification (bit 3)
No. 40	Input function selection 010	0: General-purpose input 1: Program number specification (bit 4)
No. 41	Input function selection 011	0: General-purpose input 1: Program number specification (bit 5)
No. 42	Input function selection 012	0: General-purpose input 1: Program number specification (bit 6)
No. 43	Input function selection 013	0: General-purpose input 1: Program number specification (LSB: bit 7)
No. 44	Input function selection 014	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
No. 45	Input function selection 015	0: General-purpose input 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)

Output

Pin No.	Wire color	Port No.	Standard Setting (in the delivery) Function	Parameter No. *1	Parameter Name	Function
				Remarks When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting.		
34	YW-4	300	Error Output at the Operation Cancellation Level or more (OFF)	No.46 No.331	Output Function Selection 300 Output Function Selection 300 (Area 2)	0: Universal Output 1: Error Output at the Operation Cancellation Level or more (ON) 2: Error Output at the Operation Cancellation Level or more (OFF) 3: Error Output at the Operation Cancellation Level or more + Emergency-stop output (ON) 4: Error Output at the Operation Cancellation Level or more + Emergency-stop output (OFF)
35	GN-4	301	READY Output (PIO Trigger Program Operation Available and without occurrence of any error at the cold start level or more) (Main Application Ver. 0.20 or later)	No.47 No.332	Output Function Selection 301 Output Function Selection 301 (Area 2)	0: Universal Input 1: READY Output (PIO Trigger Program Operation Available) 2: READY Output (PIO Trigger Program Operation Available)and without occurrence of any error at the operation cancellation level or more 3: READY Output (PIO Trigger Program Operation Available)and READY Output (PIO Trigger Program Operation Available, and without occurrence of any error at the cold start level or more or more level or more
36	BL-4	302	Emergency-stop output (OFF)	No.48 No.333	Output Function Selection 302 Output Function Selection 302 (Area 2)	0: Universal Input 2: Emergency-stop output (ON) 3: Emergency-stop output (OFF)
37	PL-4	303	Universal Output	No.49 No.334	Output Function Selection 303 Output Function Selection 303 (Area 2)	0: Universal Output 1: AUTO Mode Output 2: Output during the Automatic Operation (In addition, when the parameter No. 12 is set to "1")
38	GY-4	304	Universal Output	No.50 No.335	Output Function Selection 304 Output Function Selection 304 (Area 2)	0: Universal Output 1: Output at the time of "All Effective Axes Homing (=0)" 2: Output when all the effective axes homing is completed 3: Output when all the effective axes home preset coordinates are set * When the actuator applicable to the absolute encoder is moved to the coordinates "0" or home preset coordinates, use "MOVE" order, not "HOME" order.
39	WT-4	305	Universal Output	No.51 No.336	Output Function Selection 305 Output Function Selection 305 (Area 2)	0: Universal Output 1: Axis 1 in-position output (turned OFF when pressing missed) 2: Output during the Axis 1 servo ON
40	BK-4	306	Universal Output	No.52 No.337	Output Function Selection 306 Output Function Selection 306 (Area 2)	0: Universal Output 1: Axis 2 in-position output (turned OFF when pressing missed) 2: Output during the Axis 2 servo ON
41	BR-5	307	Universal Output	No.53 No.338	Output Function Selection 307 Output Function Selection 307 (Area 2)	0: Universal Output 1: Axis 3 in-position output (turned OFF when pressing missed) 2: Output during the Axis 3 servo ON
42	RD-5	308	Universal Output	No.54 No.339	Output Function Selection 308 Output Function Selection 308 (Area 2)	0: Universal Output 1: Axis 4 in-position output (turned OFF when pressing missed) 2: Output during the Axis 4 servo ON
43	OR-5	309	Universal Output	No.55 No.340	Output Function Selection 309 Output Function Selection 309 (Area 2)	0: Universal Output 1: Axis 5 in-position output (turned OFF when pressing missed) 2: Output during the Axis 5 servo ON
44	YW-5	310	Universal Output	No.56 No.341	Output Function Selection 310 Output Function Selection 310 (Area 2)	0: Universal Output 1: Axis 6 in-position output (turned OFF when pressing missed) 2: Output during the Axis 6 servo ON
45	GN-5	311	Universal Output	No.57 No.342	Output Function Selection 311 Output Function Selection 311 (Area 2)	
46	BL-5	312	Universal Output	No.58 No.343	Output Function Selection 312 Output Function Selection 312 (Area 2)	

Pin No.	Wire color	Port No.	Standard Setting (in the delivery) Function	Remarks: When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting. Parameter No. *1	Parameter Name	Function
47	PL-5	313	Universal Output	No.59 No.344	Output Function Selection 313 Output Function Selection 313 (Area 2)	0: Universal Output 1: System Memory Backup Battery Low Voltage Alarm Level or less
48	GY-5	314	Universal Output	No.60 No.345	Output Function Selection 314 Output Function Selection 314 (Area 2)	0: Universal Output 1: Absolute Battery Backup Battery Low Voltage Alarm Level or less (All axes OR check: Error level detection is maintained until power ON reset and software reset)
49	WT-5	315	Universal Output	No.61 No.346	Output Function Selection 315 Output Function Selection 315 (Area 2)	
50	BK-5		0V Output			

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.

[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-P/Q/PCT/QCT Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	System-memory backup battery voltage low warning
7003	System-memory backup battery voltage error
7004	(Reserved by the system = Use is strictly prohibited)
7005	(Reserved by the system = Use is strictly prohibited)
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	(Reserved by the system = Use is strictly prohibited)
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014	(Reserved by the system = Use is strictly prohibited)
7015	Axis 1 absolute-data backup battery voltage low warning
7016	Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7017	Axis 2 absolute-data backup battery voltage low warning (Main application version 0.28 or later)
7018	Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7019	Axis 3 absolute-data backup battery voltage low warning
7020	Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7021	Axis 4 absolute-data backup battery voltage low warning
7022	Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7023	Axis 5 absolute-data backup battery voltage low warning (Effective only with 6-axis types)
7024	Axis 5 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) (Effective only with 6-axis types)
7025	Axis 6 absolute-data backup battery voltage low warning (Effective only with 6-axis types)
7026	Axis 6 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) (Effective only with 6-axis types)
7027 to 7040	(Reserved by the system = Use is strictly prohibited)
7041, 7042	(For future expansion = Use is strictly prohibited)
7043	Axis 1 home return completion
7044	Axis 2 home return completion
7045	Axis 3 home return completion
7046	Axis 4 home return completion
7047	Axis 5 home return completion
7048	Axis 6 home return completion
7049 to 7070	(For future expansion = Use is strictly prohibited)
7071	In the AUTO mode
7072	During auto operation
7073 to 7100	(Reserved by the system = Use is strictly prohibited)
7101	Program No. 01 is being executed (or paused).
~	~
7164	Program No. 64 is being executed (or paused).

XSEL-P/Q/PCT/QCT Virtual Input Ports (Internal Flags)

Port No.	Function
7165	Program No. 65 is being executed (or paused). (Controller with increased memory capacity (with gateway function) only)
~	~
7228	Program No. 128 is being executed (or paused). (Controller with increased memory capacity (with gateway function) only)
7229 to 7299	(For future expansion = Use is strictly prohibited)

XSEL-P/Q/PCT/QCT Virtual Output Ports (Internal Flags)

Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	(For future expansion = Use is strictly prohibited)
7381 to 7399	(Reserved by the system = Use is strictly prohibited)
7400 to 7599	(For future expansion = Use is strictly prohibited)

2.1.3 XSEL-PX/QX Controllers

[1] Input and Output I/O Port

With XSEL-PX/QX type controllers, the assignments of input and output functions to I/O ports are fixed and cannot be changed.

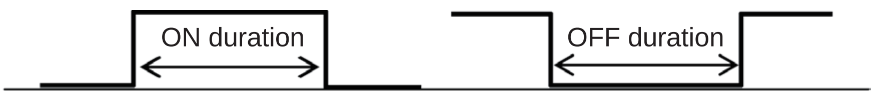
Input

Pin No.	Wire color	Port No.	Standard (factory-set) function	I/O parameter Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters.	
1	Brown-1		+24V input		
2	Red-1	000	Program start	No. 30	0: General-purpose input 1: Program start (input ports 007 to 013, BCD specification) 2: Program start (input ports 007 to 013, binary specification) 3: Program start (input ports 008 to 014, BCD specification) 4: Program start (input ports 008 to 014, binary specification)
3	Orange-1	001	General-purpose input	No. 31	0: General-purpose input 1: Soft reset signal
4	Yellow-1	002	General-purpose input	No. 32	0: General-purpose input 1: Servo ON signal
5	Green-1	003	General-purpose input	No. 33	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
6	Blue-1	004	General-purpose input	No. 34	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
7	Purple-1	005	General-purpose input	No. 35	0: General-purpose input 1: Operation pause cancellation input (ON edge)
8	Gray-1	006	General-purpose input	No. 36	0: General-purpose input 1: Operation pause signal (OFF level)
9	White-1	007	Program number specification (MSB)	No. 37	0: General-purpose input 1: Program number specification (MSB)
10	Black-1	008	Program number specification (bit 2)	No. 38	0: General-purpose input 1: Program number specification (bit 2)
11	Brown-2	009	Program number specification (bit 3)	No. 39	0: General-purpose input 1: Program number specification (bit 3)
12	Red-2	010	Program number specification (bit 4)	No. 40	0: General-purpose input 1: Program number specification (bit 4)
13	Orange-2	011	Program number specification (bit 5)	No. 41	0: General-purpose input 1: Program number specification (bit 5)
14	Yellow-2	012	Program number specification (bit 6)	No. 42	0: General-purpose input 1: Program number specification (bit 6)
15	Green-2	013	Program number specification (LSB: bit 7)	No. 43	0: General-purpose input 1: Program number specification (LSB: bit 7)
16	Blue-2	014	General-purpose input	No. 44	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
17	Purple-2	015	General-purpose input	No. 45	0: General-purpose input 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)
18	Gray-2	016	General-purpose input		
19	White-2	017	General-purpose input		
20	Black-2	018	General-purpose input		
21	Brown-3	019	General-purpose input		
22	Red-3	020	General-purpose input		
23	Orange-3	021	General-purpose input		
24	Yellow-3	022	General-purpose input		
25	Green-3	023	General-purpose input		
26	Blue-3	024	General-purpose input		
27	Purple-3	025	General-purpose input		
28	Gray-3	026	General-purpose input		
29	White-3	027	General-purpose input		
30	Black-3	028	General-purpose input		
31	Brown-4	029	General-purpose input		
32	Red-4	030	General-purpose input		
33	Orange-4	031	General-purpose input		

Output

Pin No.	Wire color	Port No.	Standard (factory-set) function		
34	Yellow-4	300	Output of operation-cancellation level or higher error (OFF)	No. 46	0: General-purpose output 1: Output of operation-cancellation level or higher error (ON) 2: Output of operation-cancellation level or higher error (OFF) 3: Output of operation-cancellation level or higher error + Emergency stop output (ON) 4: Output of operation-cancellation level or higher error + Emergency stop output (OFF)
35	Green-4	301	READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) (Main application Ver.0.20 or later)	No. 47	0: General-purpose output 1: READY output (PIO-trigger program operation enabled) 2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) 3: READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)
36	Blue-4	302	Emergency stop output (OFF)	No. 48	0: General-purpose output 2: Emergency stop output (ON) 3: Emergency stop output (OFF)
37	Purple-4	303	General-purpose output	No. 49	0: General-purpose output 1: AUTO mode output 2: Auto operation output (When other parameter No. 12 is set to ‘1’)
38	Gray-4	304	General-purpose output	No. 50	0: General-purpose output The following settings are effective only with XSEL-J/K: 1: Output when all effective axes are home (= 0) 2: Output when all effective axes have completed home return 3: Output when all effective axes are at home preset coordinate * To move an actuator of absolute encoder specification to coordinate 0 or the home preset coordinate, use a MOVP command instead of HOME command.
39	White-4	305	General-purpose output	No. 51	0: General-purpose output 2: Axis 1 servo ON output
40	Black-4	306	General-purpose output	No. 52	0: General-purpose output 2: Axis 2 servo ON output
41	Brown-5	307	General-purpose output	No. 53	0: General-purpose output 2: Axis 3 servo ON output
42	Red-5	308	General-purpose output	No. 54	0: General-purpose output 2: Axis 4 servo ON output
43	Orange-5	309	General-purpose output	No. 55	0: General-purpose output 2: Axis 5 servo ON output
44	Yellow-5	310	General-purpose output	No. 56	0: General-purpose output 2: Axis 6 servo ON output
45	Green-5	311	General-purpose output	No. 57	
46	Blue-5	312	General-purpose output	No. 58	
47	Purple-5	313	General-purpose output	No. 59	0: General-purpose output 1: System-memory backup battery voltage low alarm level or lower
48	Gray-5	314	General-purpose output	No. 60	0: General-purpose output 1: Absolute-battery backup battery voltage low alarm level or lower (OR check of all axes. If an error level is detected, this output is retained until power-ON reset or software reset.)
49	White-5	315	General-purpose output	No. 61	
50	Black-5		0V input		

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port
Should be the same as XSEL-P/Q/PCT/QCT [Refer to 2.1.2 XSEL-P/Q/PCT/QCT]

XSEL-PX/QX Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	Voltage low warning for system memory backup battery
7003	Abnormal voltage of system memory backup battery
7004	(For future expansion = Use strictly prohibited)
7005	(For future expansion = Use strictly prohibited)
7006	Top level system error = Message level error is present
7007	Top level system error = Operation cancellation level error is present
7008	Top level system error = Cold start level error is present
7009	(For future expansion = Use strictly prohibited)
7010	Drive source cutoff factor is present (including when waiting for cutoff reset input)
7011	Latch signal indicating that all operation cancellation factor is present (latch signal for recognizing 1-shot cancellation factor; latch is cancelled by 7300 being ON)
7012	All operation pause factor is present (including when waiting for restart switch signal. Valid only during automatic operation recognition)
7013	All servo axis interlock factor is present (all operation pause factor + interlock input port factor)
7014	(For future expansion = Use strictly prohibited)
7015	Voltage low warning for axis 1 absolute data backup battery
7016	Abnormal voltage of axis 1 absolute data backup battery (latched until power on reset or software reset)
7017	Voltage low warning for axis 2 absolute data backup battery (main application version 0.28 or later)
7018	Abnormal voltage of axis 2 absolute data backup battery (latched until power on reset or software reset)
7019	Voltage low warning for axis 3 absolute data backup battery
7020	Abnormal voltage of axis 3 absolute data backup battery (latched until power on reset or software reset)
7021	Voltage low warning for axis 4 absolute data backup battery
7022	Abnormal voltage of axis 4 absolute data backup battery (latched until power on reset or software reset)
7023	Voltage low warning for axis 5 absolute data backup battery (valid only when the controller supports up to 6 axes)
7024	Abnormal voltage of axis 5 absolute data backup battery (latched until power on reset or software reset. Valid only when the controller supports up to 6 axes)
7025	Voltage low warning for axis 6 absolute data backup battery (valid only when the controller supports up to 6 axes)
7026	Abnormal voltage of axis 6 absolute data backup battery (latched until power on reset or software reset. Valid only when the controller supports up to 6 axes)
7027 to 7040	(For future expansion = Use strictly prohibited)
7041 to 7070	(For future expansion = Use strictly prohibited)
7071	In AUTO mode
7072	During automatic operation
7073 to 7100	(For future expansion = Use strictly prohibited)
7101	Running program No. 01 (including during pause)
~	~
7164	Running program No. 64 (including during pause)
7165	Running program No. 65 (including during pause) (Controller with increased memory size (with gateway function) only)
~	~
7228	Running program No. 128 (including during pause) (Controller with increased memory size (with gateway function) only)
7229 to 7299	(For future expansion = Use strictly prohibited)

XSEL-PX/QX Virtual Output Ports (Internal Flags)

Port No.	Function
7300	Latch cancellation output for a latch signal indicating that all operation cancellation factor is present (port 7011. The latch is cancelled only when operation cancellation factor is no longer present. 7300 will be turned OFF following an attempt to cancel latch)
7301 to 7380	(For future expansion = Use strictly prohibited)
7381 to 7399	(For future expansion = Use strictly prohibited)
7400 to 7599	(For future expansion = Use strictly prohibited)

2.1.4 XSEL-R/S Controllers

[1] Input and Output I/O Port

With XSEL-R/S controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

Pin No.	Wire color	Port No.	Standard (factory-set) function
1	Brown-1		+24V input
2	Red-1	000	Program start
3	Orange-1	001	General-purpose input
4	Yellow-1	002	General-purpose input
5	Green-1	003	General-purpose input
6	Blue-1	004	General-purpose input
7	Purple-1	005	General-purpose input
8	Gray-1	006	General-purpose input
9	White-1	007	Program number specification (MSB)
10	Black-1	008	Program number specification (bit 2)
11	Brown-2	009	Program number specification (bit 3)
12	Red-2	010	Program number specification (bit 4)
13	Orange-2	011	Program number specification (bit 5)
14	Yellow-2	012	Program number specification (bit 6)
15	Green-2	013	Program number specification (LSB: bit 7)
16	Blue-2	014	General-purpose input
17	Purple-2	015	General-purpose input
18	Gray-2	016	General-purpose input
19	White-2	017	General-purpose input
20	Black-2	018	General-purpose input
21	Brown-3	019	General-purpose input
22	Red-3	020	General-purpose input
23	Orange-3	021	General-purpose input
24	Yellow-3	022	General-purpose input
25	Green-3	023	General-purpose input
26	Blue-3	024	General-purpose input
27	Purple-3	025	General-purpose input
28	Gray-3	026	General-purpose input
29	White-3	027	General-purpose input
30	Black-3	028	General-purpose input
31	Brown-4	029	General-purpose input
32	Red-4	030	General-purpose input
33	Orange-4	031	General-purpose input

Remarks		
Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters.		
Parameter No.	Parameter name	Function
No. 30	Input function selection 000	0: General-purpose input 1: Program start (input ports 007 to 013, BCD specification) 2: Program start (input ports 007 to 013, binary specification) 3: Program start (input ports 008 to 014, BCD specification) 4: Program start (input ports 008 to 014, binary specification)
No. 31	Input function selection 001	0: General-purpose input 1: Soft reset signal
No. 32	Input function selection 002	0: General-purpose input 1: Servo ON signal
No. 33	Input function selection 003	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
No. 34	Input function selection 004	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
No. 35	Input function selection 005	0: General-purpose input 1: Operation pause cancellation input (ON edge)
No. 36	Input function selection 006	0: General-purpose input 1: Operation pause signal (OFF level)
No. 37	Input function selection 007	0: General-purpose input 1: Program number specification (MSB)
No. 38	Input function selection 008	0: General-purpose input 1: Program number specification (bit 2)
No. 39	Input function selection 009	0: General-purpose input 1: Program number specification (bit 3)
No. 40	Input function selection 010	0: General-purpose input 1: Program number specification (bit 4)
No. 41	Input function selection 011	0: General-purpose input 1: Program number specification (bit 5)
No. 42	Input function selection 012	0: General-purpose input 1: Program number specification (bit 6)
No. 43	Input function selection 013	0: General-purpose input 1: Program number specification (LSB: bit 7)
No. 44	Input function selection 014	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
No. 45	Input function selection 015	0: General-purpose input 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)

Output

Pin No.	Wire color	Port No.	Standard Setting (in the delivery) Function	Remarks		
				When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting.		
				Parameter No. *1	Parameter Name	Function
34	YW-4	300	Error Output at the Operation Cancellation Level or more (OFF)	No.46 No.331	Output Function Selection 300 Output Function Selection 300 (Area 2)	0: Universal Output 1: Error Output at the Operation Cancellation Level or more (ON) 2: Error Output at the Operation Cancellation Level or more (OFF) 3: Error Output at the Operation Cancellation Level or more + Emergency-stop output (ON) 4: Error Output at the Operation Cancellation Level or more + Emergency-stop output (OFF)
35	GN-4	301	READY Output (PIO Trigger Program Operation Available and without occurrence of any error at the cold start level or more) (Main Application Ver. 0.20 or later)	No.47 No.332	Output Function Selection 301 Output Function Selection 301 (Area 2)	0: Universal Input 1: READY Output (PIO Trigger Program Operation Available) 2: READY Output (PIO Trigger Program Operation Available)and without occurrence of any error at the operation cancellation level or more 3: READY Output (PIO Trigger Program Operation Available)and READY Output (PIO Trigger Program Operation Available, and without occurrence of any error at the cold start level or more or more level or more
36	BL-4	302	Emergency-stop output (OFF)	No.48 No.333	Output Function Selection 302 Output Function Selection 302 (Area 2)	0: Universal Input 2: Emergency-stop output (ON) 3: Emergency-stop output (OFF)
37	PL-4	303	Universal Output	No.49 No.334	Output Function Selection 303 Output Function Selection 303 (Area 2)	0: Universal Output 1: AUTO Mode Output 2: Output during the Automatic Operation (In addition, when the parameter No. 12 is set to “1”)
38	GY-4	304	Universal Output	No.50 No.335	Output Function Selection 304 Output Function Selection 304 (Area 2)	0: Universal Output 1: Output at the time of “All Effective Axes Homing (=0)” 2: Output when all the effective axes homing is completed (Coordinates determined) 3: Output when all the effective axes home preset coordinates are set
39	WT-4	305	Universal Output	No.51 No.336	Output Function Selection 305 Output Function Selection 305 (Area 2)	0: Universal Output 1: Axis 1 in-position output (turned OFF when pressing missed) 2: Output during the Axis 1 servo ON
40	BK-4	306	Universal Output	No.52 No.337	Output Function Selection 306 Output Function Selection 306 (Area 2)	0: Universal Output 1: Axis 2 in-position output (turned OFF when pressing missed) 2: Output during the Axis 2 servo ON
41	BR-5	307	Universal Output	No.53 No.338	Output Function Selection 307 Output Function Selection 307 (Area 2)	0: Universal Output 1: Axis 3 in-position output (turned OFF when pressing missed) 2: Output during the Axis 3 servo ON
42	RD-5	308	Universal Output	No.54 No.339	Output Function Selection 308 Output Function Selection 308 (Area 2)	0: Universal Output 1: Axis 4 in-position output (turned OFF when pressing missed) 2: Output during the Axis 4 servo ON
43	OR-5	309	Universal Output	No.55 No.340	Output Function Selection 309 Output Function Selection 309 (Area 2)	0: Universal Output 1: Axis 5 in-position output (turned OFF when pressing missed) 2: Output during the Axis 5 servo ON
44	YW-5	310	Universal Output	No.56 No.341	Output Function Selection 310 Output Function Selection 310 (Area 2)	0: Universal Output 1: Axis 6 in-position output (turned OFF when pressing missed) 2: Output during the Axis 6 servo ON
45	GN-5	311	Universal Output	No.57 No.342	Output Function Selection 311 Output Function Selection 311 (Area 2)	0: Universal Output 1: Axis 7 in-position output (turned OFF when pressing missed) 2: Output during the Axis 7 servo ON (system monitoring task output)
46	BL-5	312	Universal Output	No.58 No.343	Output Function Selection 312 Output Function Selection 312 (Area 2)	0: Universal Output 1: Axis 8 in-position output (turned OFF when pressing missed) 2: Output during the Axis 8 servo ON (system monitoring task output)

Pin No.	Wire color	Port No.	Standard Setting (in the delivery) Function
47	PL-5	313	Universal Output
48	GY-5	314	Universal Output
49	WT-5	315	Universal Output
50	BK-5		0V Output

Remarks		
When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting.		
Parameter No. *1	Parameter Name	Function
No.59 No.344	Output Function Selection 313 Output Function Selection 313 (Area 2)	0: Universal Output 1: System Memory Backup Battery Low Voltage Alarm Level or less
No.60 No.345	Output Function Selection 314 Output Function Selection 314 (Area 2)	0: Universal Output
No.61 No.346	Output Function Selection 315 Output Function Selection 315 (Area 2)	0: Universal Output

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.

[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-R/S/RX/SX/RXD/SXD Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002 to 7005	(Reserved by the system = Use is strictly prohibited)
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	(Reserved by the system = Use is strictly prohibited)
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014	(Reserved by the system = Use is strictly prohibited)
7015	Axis 1 absolute-data backup battery voltage low warning
7016	Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7017	Axis 2 absolute-data backup battery voltage low warning
7018	Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7019	Axis 3 absolute-data backup battery voltage low warning
7020	Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7021	Axis 4 absolute-data backup battery voltage low warning
7022	Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7023	Axis 5 absolute-data backup battery voltage low warning
7024	Axis 5 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7025	Axis 6 absolute-data backup battery voltage low warning
7025	Axis 6 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7027	Axis 7 absolute-data backup battery voltage low warning
7028	Axis 7 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7029	Axis 8 absolute-data backup battery voltage low warning
7030	Axis 8 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7031 to 7040	(Reserved by the system = Use is strictly prohibited)
7041 to 7042	(For future expansion = Use is strictly prohibited)
7043	Axis 1 home return completion
7044	Axis 2 home return completion
~	~
7048	Axis 6 home return completion
7049	Axis 7 home return completion
7050	Axis 8 home return completion
7051 to 7069	(For future expansion = Use is strictly prohibited)
7070	(Reserved by the system = Use is strictly prohibited)
7071	In the AUTO mode
7072	During auto operation
7073 to 7074	(For future expansion = Use is strictly prohibited)
7075	During Tracking Conveyor Speed Drop Detection

XSEL-R/S/RX/SX/RXD/SXD Virtual Input Ports (Internal Flags)

Port No.	Function
7076	Tracking Operation Complete Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”)
7077	In Tracking Conveyor Tracking Complete Range
7078	Tracking Reversed Operation Detected Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”)
7079	During Tracking Mode (Work detection valid) (for SCARA controller only)
7080	During Tracking Operation (including a pause in tracking operation)
7081 to 7100	(Reserved by the system = Use is strictly prohibited)
7101	Program No. 01 is being executed (or paused).
7102	Program No. 02 is being executed (or paused).
7103	Program No. 03 is being executed (or paused).
~	~
7227	Program No. 127 is being executed (or paused).
7228	Program No. 128 is being executed (or paused).
7229 to 7299	(Reserved by the system = Use is strictly prohibited)

XSEL-R/S/RX/SX/RXD/SXD Virtual Output Ports (Internal Flags)

Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	(For future expansion = Use is strictly prohibited)
7381 to 7399	(Reserved by the system = Use is strictly prohibited)
7400 to 7599	(For future expansion = Use is strictly prohibited)

2.1.5 XSEL-RX/SX/RXD/SXD Controllers

[1] Input and Output I/O Port

With XSEL-RX/SX/RXD/SXD type controllers, the assignments of input and output functions to I/O ports are fixed and cannot be changed.

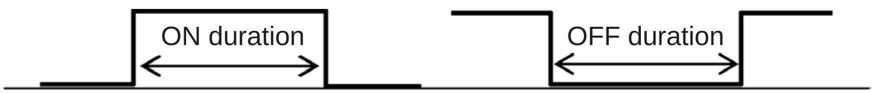
Input

Pin No.	Wire color	Port No.	Standard (factory-set) function	I/O parameter Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters.	
1	Brown-1		+24V input		
2	Red-1	000	Program start	No. 30	0: General-purpose input 1: Program start (input ports 007 to 013, BCD specification) 2: Program start (input ports 007 to 013, binary specification) 3: Program start (input ports 008 to 014, BCD specification) 4: Program start (input ports 008 to 014, binary specification)
3	Orange-1	001	General-purpose input	No. 31	0: General-purpose input 1: Soft reset signal
4	Yellow-1	002	General-purpose input	No. 32	0: General-purpose input 1: Servo ON signal
5	Green-1	003	General-purpose input	No. 33	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
6	Blue-1	004	General-purpose input	No. 34	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
7	Purple-1	005	General-purpose input	No. 35	0: General-purpose input 1: Operation pause cancellation input (ON edge)
8	Gray-1	006	General-purpose input	No. 36	0: General-purpose input 1: Operation pause signal (OFF level)
9	White-1	007	Program number specification (MSB)	No. 37	0: General-purpose input 1: Program number specification (MSB)
10	Black-1	008	Program number specification (bit 2)	No. 38	0: General-purpose input 1: Program number specification (bit 2)
11	Brown-2	009	Program number specification (bit 3)	No. 39	0: General-purpose input 1: Program number specification (bit 3)
12	Red-2	010	Program number specification (bit 4)	No. 40	0: General-purpose input 1: Program number specification (bit 4)
13	Orange-2	011	Program number specification (bit 5)	No. 41	0: General-purpose input 1: Program number specification (bit 5)
14	Yellow-2	012	Program number specification (bit 6)	No. 42	0: General-purpose input 1: Program number specification (bit 6)
15	Green-2	013	Program number specification (LSB: bit 7)	No. 43	0: General-purpose input 1: Program number specification (LSB: bit 7)
16	Blue-2	014	General-purpose input	No. 44	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
17	Purple-2	015	General-purpose input	No. 45	0: General-purpose input 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)
18	Gray-2	016	General-purpose input		
19	White-2	017	General-purpose input		
20	Black-2	018	General-purpose input		
21	Brown-3	019	General-purpose input		
22	Red-3	020	General-purpose input		
23	Orange-3	021	General-purpose input		
24	Yellow-3	022	General-purpose input		
25	Green-3	023	General-purpose input		
26	Blue-3	024	General-purpose input		
27	Purple-3	025	General-purpose input		
28	Gray-3	026	General-purpose input		
29	White-3	027	General-purpose input		
30	Black-3	028	General-purpose input		
31	Brown-4	029	General-purpose input		
32	Red-4	030	General-purpose input		
33	Orange-4	031	General-purpose input		

Output

Pin No.	Wire color	Port No.	Standard (factory-set) function		
34	Yellow-4	300	Output of operation-cancellation level or higher error (OFF)	No. 46	0: General-purpose output 1: Output of operation-cancellation level or higher error (ON) 2: Output of operation-cancellation level or higher error (OFF) 3: Output of operation-cancellation level or higher error + Emergency stop output (ON) 4: Output of operation-cancellation level or higher error + Emergency stop output (OFF) 5: Error output of cold start level or more (ON) 6: Error output of cold start level or more (OFF)
35	Green-4	301	READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) (Main application Ver.0.20 or later)	No. 47	0: General-purpose output 1: READY output (PIO-trigger program operation enabled) 2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) 3: READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)
36	Blue-4	302	Emergency stop output (OFF)	No. 48	0: General-purpose output 2: Emergency stop output (ON) 3: Emergency stop output (OFF)
37	Purple-4	303	General-purpose output	No. 49	0: General-purpose output 1: AUTO mode output 2: Auto operation output (When other parameter No. 12 is set to ‘1’)
38	Gray-4	304	General-purpose output	No. 50	0: General-purpose output 1: Output at the time of “All Effective Axes Homing (=0)” 2: Output when all effective linear drive axis home-return operation is complete (coordinate is established) 3: Output when all the effective axes home preset coordinates are set * Use MOVP Command, not HOME Command, if moving the ABS encoder axes to the coordinate 0 or home preset coordinate.
39	White-4	305	General-purpose output	No. 51	0: General-purpose output 1: Axis 1 in-position output (turned OFF when pressing missed) 2: Output during the Axis 1 servo ON
40	Black-4	306	General-purpose output	No. 52	0: General-purpose output 1: Axis 2 in-position output (turned OFF when pressing missed) 2: Output during the Axis 2 servo ON
41	Brown-5	307	General-purpose output	No. 53	0: General-purpose output 1: Axis 3 in-position output (turned OFF when pressing missed) 2: Output during the Axis 3 servo ON
42	Red-5	308	General-purpose output	No. 54	0: General-purpose output 1: Axis 4 in-position output (turned OFF when pressing missed) 2: Output during the Axis 4 servo ON
43	Orange-5	309	General-purpose output	No. 55	0: General-purpose output 1: Axis 5 in-position output (turned OFF when pressing missed) 2: Output during the Axis 5 servo ON
44	Yellow-5	310	General-purpose output	No. 56	0: General-purpose output 1: Axis 6 in-position output (turned OFF when pressing missed) 2: Output during the Axis 6 servo ON
45	Green-5	311	General-purpose output	No. 57	0: General-purpose output 1: Axis 7 in-position output (turned OFF when pressing missed) 2: Output during the Axis 7 servo ON (system monitoring task output)
46	Blue-5	312	General-purpose output	No. 58	0: General-purpose output 1: Axis 8 in-position output (turned OFF when pressing missed) 2: Output during the Axis 8 servo ON (system monitoring task output)
47	Purple-5	313	General-purpose output	No. 59	0: General-purpose output 1: System-memory backup battery voltage low alarm level or lower
48	Gray-5	314	General-purpose output	No. 60	0: General-purpose output 1: Absolute-battery backup battery voltage low alarm level or lower (OR check of all axes. If an error level is detected, this output is retained until power-ON reset or software reset.)
49	White-5	315	General-purpose output	No. 61	
50	Black-5		0V input		

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port

Should be the same as XSEL-R/S. [Refer to 2.1.4 XSEL-R/S]

2.1.6 SSEL, ASEL, PSEL Controllers

[1] Input and Output I/O Port

With SSEL, ASEL and PSEL controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input function setting values (0 to 23) in input function selections 000 to 015 (I/O parameters 30 to 45) corresponding to port No. 000 to 015 or input function selections 016 to 023 (I/O parameters 251 to 258) corresponding to port No. 16 to 23, and the set functions will be assigned.
For output ports, set output function setting values (0 to 17, 24, 25) in output function selections 300 to 307 (I/O parameters 46 to 53) corresponding to port No. 300 to 307, and the set functions will be assigned.

Program mode
Input

Pin No.	Wire color	Port No.	Standard (factory-set) function	Paramet er No.	Parameter name	Input function setting value (factory setting)
1A	Brown 1	-	I/O power supply +24 V	-	-	-
1B	Red 1	016	Program No. 1 selection (MSB)	251	Input function selection 016	9
2A	Orange 1	017	Program No. 2 selection (bit 2)	252	Input function selection 017	10
2B	Yellow 1	018	Program No. 4 selection (bit 3)	253	Input function selection 018	11
3A	Green 1	019	Program No. 8 selection (bit 4)	254	Input function selection 019	12
3B	Blue 1	020	Program No. 10 selection (bit 5)	255	Input function selection 020	13
4A	Purple 1	021	Program No. 20 selection (bit 6)	256	Input function selection 021	14
4B	Gray 1	022	Program No. 40 selection (LSB: bit 7)	257	Input function selection 022	15
5A	White 1	023	Software reset	258	Input function selection 023	3
5B	Black 1	000	Program start	30	Input function selection 000	1
6A	Brown 2	001	General-purpose input	31	Input function selection 001	0
6B	Red 2	002	General-purpose input	32	Input function selection 002	0
7A	Orange 2	003	General-purpose input	33	Input function selection 003	0
7B	Yellow 2	004	General-purpose input	34	Input function selection 004	0
8A	Green 2	005	General-purpose input	35	Input function selection 005	0
8B	Blue 2	006	General-purpose input	36	Input function selection 006	0
9A	Purple 2	007	General-purpose input	37	Input function selection 007	0
9B	Gray 2	008	General-purpose input	38	Input function selection 008	0
10A	White 2	009	General-purpose input	39	Input function selection 009	0
10B	Black 2	010	General-purpose input	40	Input function selection 010	0
11A	Brown 3	011	General-purpose input	41	Input function selection 011	0
11B	Red 3	012	General-purpose input	42	Input function selection 012	0
12A	Orange 3	013	General-purpose input	43	Input function selection 013	0
12B	Yellow 3	014	General-purpose input	44	Input function selection 014	0
13A	Green 3	015	General-purpose input	45	Input function selection 015	0

Input function setting value	Function
0	General-purpose input
1	Program start (BCD) (ON edge) signal
2	Program start (BIN) (ON edge) signal
3	Soft reset signal (ON for 1sec)
4	Servo ON signal (ON edge)
5	Auto program start signal (ON edge)
6	All-servo-axis software interlock (OFF level)
7	Operation pause cancellation input (ON edge)
8	Operation pause signal (OFF level)
9	Program number specification (MSB)
10	Program number specification (bit 2)
11	Program number specification (bit 3)
12	Program number specification (bit 4)
13	Program number specification (bit 5)
14	Program number specification (bit 6)
15	Program number specification (LSB: bit 7)
16	Error reset (ON edge)
17	Drive-source cutoff cancellation input (ON edge)
18	All-effective-axis home return command signal (ON edge)
19	All-effective-incremental-axis home return (ON edge)
20	PC/teaching pendant servo movement command acceptance input
21	Remote mode control input
22	Axis 1 forced brake release input
23	Axis 2 forced brake release input
24 to 27	Reserved by the system
24	Program number specification (bit 8)
25	Program number specification (bit 9)

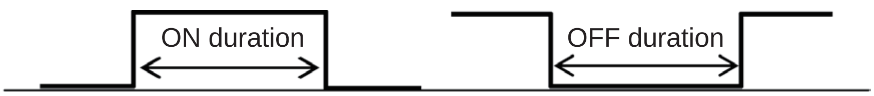
Program mode
Output

Pin No.	Wire color	Port No.	Standard (factory-set) function	Parameter No.	Parameter name	Input function setting value (factory setting)
13B	Blue 3	300	Alarm output	46	Output function selection 300	2
14A	Purple 3	301	Ready output	47	Output function selection 301	7
14B	Gray 3	302	General-purpose output	48	Output function selection 302	0
15A	White 3	303	General-purpose output	49	Output function selection 303	0
15B	Black 3	304	General-purpose output	50	Output function selection 304	0
16A	Brown 4	305	General-purpose output	51	Output function selection 305	0
16B	Red 4	306	General-purpose output	52	Output function selection 306	0
17A	Orange 4	307	General-purpose output	53	Output function selection 307	0
17B	Yellow 4	N	I/O power supply 0V	-	-	-

Input function setting value	Function
0	General-purpose input
1	Output of operation-cancellation level or higher error (ON)
2	Output of operation-cancellation level or higher error (OFF)
3	Output of operation-cancellation level or higher error + Emergency stop output (ON)
4	Output of operation-cancellation level or higher error + Emergency stop output (OFF)
5	READY output (PIO-trigger program operation enabled)
6	READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error)
7	READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)
8	Emergency stop output (ON)
9	Emergency stop output (OFF)
10	AUTO mode output
11	Auto operation output
12	Output when all effective axes are home (= 0)
13	Output when all effective axes have completed home return
14	Output when all effective axes are at home preset coordinate
15	System-memory backup battery (optional) voltage low warning output
16	Absolute-data backup battery (optional) voltage low warning output
17	Drive-source cutoff (SDN) notification output
24	Axis 1 servo ON output
25	Axis 2 servo ON output

*1 Output function setting values 1, 2, 3 and 4 cannot be assigned at the same time.
*2 Output function setting values 5, 6 and 7 cannot be assigned at the same time.

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

ASEL/PSEL/SSEL Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	System-memory backup battery voltage low warning
7003	System-memory backup battery voltage error
7004	(Reserved by the system = Use is strictly prohibited)
7005	(Reserved by the system = Use is strictly prohibited)
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	(Reserved by the system = Use is strictly prohibited)
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014	(Reserved by the system = Use is strictly prohibited)
7015	Axis 1 absolute-data backup battery voltage low warning
7016	Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7017	Axis 2 absolute-data backup battery voltage low warning
7018	Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)
7019 to 7026	(For future expansion = Use is strictly prohibited)
7027 to 7040	(Reserved by the system = Use is strictly prohibited)
7041, 7042	(For future expansion = Use is strictly prohibited)
7043	Axis 1 home return completion
7044	Axis 2 home return completion
7045 to 7070	(For future expansion = Use is strictly prohibited)
7071	In the AUTO mode
7072	During auto operation
7073 to 7100	(Reserved by the system = Use is strictly prohibited)
7101	Program No. 01 is being executed (or paused).
~	~
7164	Program No. 64 is being executed (or paused).
7165	Program No. 65 is being executed (or paused). ··· Dedicated only for SSEL with expanded memory capacity
~	~
7228	Program No. 128 is being executed (or paused). Dedicated only for SSEL with expanded memory capacity
7229 to 7299	(For future expansion = Use is strictly prohibited)

ASEL/PSEL/SSEL Virtual Output Ports (Internal Flags)

Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	(For future expansion = Use is strictly prohibited)
7381 to 7399	(Reserved by the system = Use is strictly prohibited)
7400 to 7599	(For future expansion = Use is strictly prohibited)

2.1.7 Tabletop Robot TT/TTA

[1] Input and Output I/O Port

With the tabletop robot TT, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

Pin No.	Wire color	Port No.	Standard (factory-set) function
1	Brown 1	-	I/O power supply +24V
2	Red 1	016	General-purpose input
3	Orange 1	017	General-purpose input
4	Yellow 1	018	General-purpose input
5	Green 1	019	General-purpose input
6	Blue 1	020	General-purpose input
7	Purple 1	021	General-purpose input
8	Gray 1	022	General-purpose input
9	White 1	023	General-purpose input
10	Black 1	024	General-purpose input
11	Brown 2	025	General-purpose input
12	Red 2	026	General-purpose input
13	Orange 2	027	General-purpose input
14	Yellow 2	028	General-purpose input
15	Green 2	029	General-purpose input
16	Blue 2	030	General-purpose input
17	Purple 2	031	General-purpose input

Remarks		
Inputs are set as general-purpose inputs, but you can change these input functions by setting applicable I/O parameters.		
Parameter No.	Parameter name	Function
30	Input function selection 000[*1]	1: Program start
31	Input function selection 001	0: General-purpose input 1: Soft reset signal
32	Input function selection 002	0: General-purpose input 1: Soft reset signal
33	Input function selection 003	0: General-purpose input 1: Auto program start upon power-ON reset or software reset in AUTO mode 2: Auto program start signal
34	Input function selection 004	0: General-purpose input 1: Software interlock of all servo axes (OFF level)
35	Input function selection 005	0: General-purpose input 1: Operation pause cancellation input (ON edge)
36	Input function selection 006	0: General-purpose input 1: Operation pause signal (OFF level)
37	Input function selection 007[*2]	0: General-purpose input 1: Program number specification (LSB)
38	Input function selection 008[*2]	0: General-purpose input 1: Program number specification (bit 2)
39	Input function selection 009[*2]	0: General-purpose input 1: Program number specification (bit 3)
40	Input function selection 010[*2]	0: General-purpose input 1: Program number specification (bit 4)
41	Input function selection 011[*2]	0: General-purpose input 1: Program number specification (bit 5)
42	Input function selection 012[*2]	0: General-purpose input 1: Program number specification (bit 6)
43	Input function selection 013[*2]	0: General-purpose input 1: Program number specification (MSB: bit 7) 2: Error reset (ON edge)
44	Input function selection 014	0: General-purpose input 1: Drive-source cutoff cancellation (ON edge)
45	Input function selection 015	0: General-purpose input 1: Home return of all effective axes (ON edge) 2: Home return of all effective incremental axes (ON edge)

*1 If input function selection 000 (program start) is assigned to a port other than No. 000, the start switch on the front panel is disabled.

*2 If input function selections 007 to 013 (program selection switches) are assigned to ports other than No. 007 to 013, the program selection switches on the front panel are disabled.

Output

Pin No.	Wire color	Port No.	Standard (factory-set) function
18	Gray 2	316	General-purpose output
19	White 2	317	General-purpose output
20	Black 2	318	General-purpose output
21	Brown 3	319	General-purpose output
22	Red 3	320	General-purpose output
23	Orange 3	321	General-purpose output
24	Yellow 3	322	General-purpose output
25	Green 3	323	General-purpose output
26	Blue 3	324	General-purpose output
27	Purple 3	325	General-purpose output
28	Gray 3	326	General-purpose output
29	White 3	327	General-purpose output
30	Black 3	328	General-purpose output
31	Brown 4	329	General-purpose output
32	Red 4	330	General-purpose output
33	Orange 4	331	General-purpose output
34	Yellow 4	-	I/O power supply 0 V

Remarks		
Outputs are set as general-purpose outputs, but you can change these output functions by setting applicable I/O parameters.		
Parameter No.	Parameter name	Function
46 331	Output function selection 300*3 Output function selection 300 (area 2)	0: General-purpose output 1: Output of operation-cancellation level or higher error (ON) 2: Output of operation-cancellation level or higher error (OFF) 3: Output of operation-cancellation level or higher error + Emergency stop output (ON) 4: Output of operation-cancellation level or higher error + Emergency stop output (OFF)
47 332	Output function selection 301*3 Output function selection 301 (area 2)	0: General-purpose output 1: READY output (PIO-trigger program operation enabled) 2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) 3: READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)
48 333	Output function selection 302*3 Output function selection 302 (area 2)	0: General-purpose output 1: Emergency stop output (ON) 2: Emergency stop output (OFF)
49 334	Output function selection 303*3 Output function selection 303 (area 2)	0: General-purpose output 1: AUTO mode output 2: Auto operation output (When other parameter No. 12 is set to '1')
50 335	Output function selection 304*3 Output function selection 304 (area 2)	0: General-purpose output 1: Output when all effective axes are home (= 0) 2: Output when all effective axes have completed home return 3: Output when all effective axes are at home preset coordinate
51 336	Output function selection 305 Output function selection 305 (area 2)	0: General-purpose output 2: Axis 1 servo ON output
52 337	Output function selection 306 Output function selection 306 (area 2)	0: General-purpose output 2: Axis 2 servo ON output
53 338	Output function selection 307 Output function selection 307 (area 2)	0: General-purpose input 2: Axis 3 servo ON output
54 339	Output function selection 308 Output function selection 308 (area 2)	0: General-purpose output
55 340	Output function selection 309 Output function selection 309 (area 2)	0: General-purpose output
56 341	Output function selection 310 Output function selection 310 (area 2)	0: General-purpose output
57 342	Output function selection 311 Output function selection 311 (area 2)	0: General-purpose output
58 343	Output function selection 312 Output function selection 312 (area 2)	0: General-purpose output
59 344	Output function selection 313 Output function selection 313 (area 2)	0: General-purpose output
60 345	Output function selection 314 Output function selection 314 (area 2)	0: General-purpose output
61 346	Output function selection 315 Output function selection 315 (area 2)	0: General-purpose output

*3 Since output function selections 300 to 304 are assigned to LEDs in the panel window, the LEDs are disabled if parameters 46 to 50 are set as general-purpose outputs or port number assignments are changed using parameters 299 to 303.
If you want to output system signals to the I/O shown in the tables above, use output function selection area 2.

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".

[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

TT/TTA Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	System-memory backup battery voltage low warning (Note TTA is for system reservation)
7003	System-memory backup battery voltage error (Note TTA is for system reservation)
7004	Reserved by the system = Use is prohibited
7005	Reserved by the system = Use is prohibited
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	Reserved by the system = Use is prohibited
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014 to 7050	Reserved by the system = Use is prohibited
7051 to 7070	For future expansion = Use is prohibited
7071	In the AUTO mode
7072	During auto operation
7073 to 7100	Reserved by the system = Use is prohibited
7101	Program No. 01 is being executed (or paused).
~	~ (Note TTA : Port No.7101 to 7228, program No.1 to 128 are being executed.)
7164	Program No. 64 is being executed (or paused).
7165 to 7299	For future expansion = Use is prohibited

TT Virtual Output Ports (Internal Flags)

Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	(For future expansion = Use is strictly prohibited)
7381 to 7399	(Reserved by the system = Use is strictly prohibited)
7400 to 7599	(For future expansion = Use is strictly prohibited) (Note TTA : Port No.7401 to 7527, Program No.129 to 255 are being executed.)

2.1.8 MSEL Controller

[1] Input and Output I/O Port

In Input Function Select No. 000 to 015 and Output Function Select No. 300 to 315, dedicated functions can be set, and they can be assigned to desired input and output ports.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
For other I/O port Nos., they can be used freely as the universal I/O port.

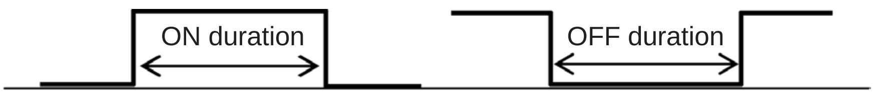
(1) Input Port Function Assignment

Input function selection No.	Parameter No.		Setting values	Input Signal Functions	Setting at delivery
000	No.30		0	General-purpose input	
			1	Program start (Indicated input port = 007 to 014, Start of program number indicated in BCD ... ON edge)	○
			2	Program start (Indicated input port = 007 to 014, Start of program number indicated in binary ... ON edge)	
001	No.31		0	General-purpose input	○
			1	Software reset signal (Kept on continuously for 1sec)	
002	No.32		0	General-purpose input	○
			1	Servo ON signal (Edge input)	
003	No.33		0	General-purpose input	
			1	Indicated program automatically starts by power-on reset / software reset in AUTO Mode	○
			2	Start of Auto Start Program (kept on continuously for 100ms)	
004	No.34		0	General-purpose input	○
			1	Software interlock on all servo axes (level signal with always being on)	
005	No.35		0	General-purpose input	○
			1	Operation pause cancellation input (On-edge process signal)	
006	No.36		0	General-purpose input	○
			1	Operation pause cancellation input (level signal with always being on)	
007	No.37	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 0 bit	○
		When Input Function Select 000 = 2			
008	No.38	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 1 bit	○
		When Input Function Select 000 = 2			
009	No.39	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 2 bit	○
		When Input Function Select 000 = 2			
010	No.40	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 3 bit	○
		When Input Function Select 000 = 2			
011	No.41	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 4 bit	○
		When Input Function Select 000 = 2			
012	No.42	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 5 bit	○
		When Input Function Select 000 = 2			
013	No.43	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 6 bit	○
		When Input Function Select 000 = 2			
014	No.44	Independent from value in Input Function Select 000	0	General-purpose input	
		When Input Function Select 000 = 1	1	Program number specification 7 bit	○
		When Input Function Select 000 = 2			
015	No.45		0	General-purpose input	○

(2) Output Port Function Assignment

Input function selection No.	Parameter No.	Setting values	Input Signal Functions	Setting at delivery
300	No.46	0	General-purpose output	
		1	Output of operation-cancellation level or higher error (ON)	
		2	Output of operation-cancellation level or higher error (OFF)	○
		3	Output of operation-cancellation level or higher error + Emergency stop output (ON)	
		4	Output of operation-cancellation level or higher error + Emergency stop output (OFF)	
		5	Error output of cold start level or more (ON)	
		6	Error output of cold start level or more (OFF)	
		7	Message level related to maintenance information alarm function (ON) for error output of (Error No. 231 to 232)	
		8	Message level related to maintenance information alarm function (OFF) for error output of (Error No. 231 to 232)	
301	No.47	0	General-purpose output	
		1	READY output (PIO-trigger program operation enabled)	
		2	READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error)	
		3	READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)(ON)	○
302	No.48	0	General-purpose output	
		1	Emergency stop output（ON）	
		2	Emergency stop output（OFF）	○
303	No.49	0	General-purpose output	○
304	No.50	0	General-purpose output	○
305	No.51	0	General-purpose output	○
306	No.52	0	General-purpose output	○
307	No.53	0	General-purpose output	○
308	No.54	0	General-purpose output	○
309	No.55	0	General-purpose output	○
310	No.56	0	General-purpose output	○
311	No.57	0	General-purpose output	○
312	No.58	0	General-purpose output	○
313	No.59	0	General-purpose output	○
314	No.60	0	General-purpose output	○
315	No.61	0	General-purpose output	○

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

MSEL Virtual Input Ports (Internal Flags)

Port No.	Function
7000	Always OFF
7001	Always ON
7002	Reserved by the system
7003	Reserved by the system
7004	Reserved by the system
7005	Reserved by the system
7006	Critical system error = A message level error is present.
7007	Critical system error = An operation-cancellation level error is present.
7008	Critical system error = A cold-start level error is present.
7009	Reserved by the system
7010	A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input).
7011	A latch signal indicating that a cause of all-operation cancellation is present. (This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON)
7012	A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed). (Effective only in the auto operation recognition mode)
7013	A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port)
7014 to 7050	Reserved by the system
7051 to 7070	For future expansion
7071	In the AUTO mode
7072	During auto operation
7073 to 7100	Reserved by the system
7101	Program No. 01 is being executed (or paused).
~	~
7228	Program No. 128 is being executed (or paused).
7229 to 7299	For future expansion

MSEL Virtual Output Ports (Internal Flags)

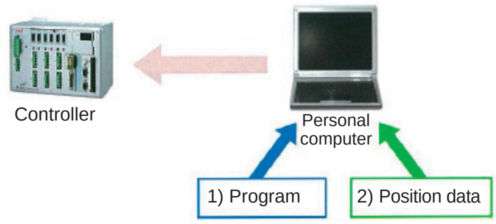
Port No.	Function
7300	A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.)
7301 to 7380	For future expansion
7381 to 7399	Reserved by the system
7400	For future expansion
7401	Program No. 129 is being executed (or paused).
~	
7527	Program No. 255 is being executed (or paused).
7528 to 7599	For future expansion

3. Program

3.1 Position Table and Program Format

To run the robot,
- Program
- Position data
  (information of positions for the robot to move)

it is necessary to create 2 types of data as shown below and input them to the controller with using PC.



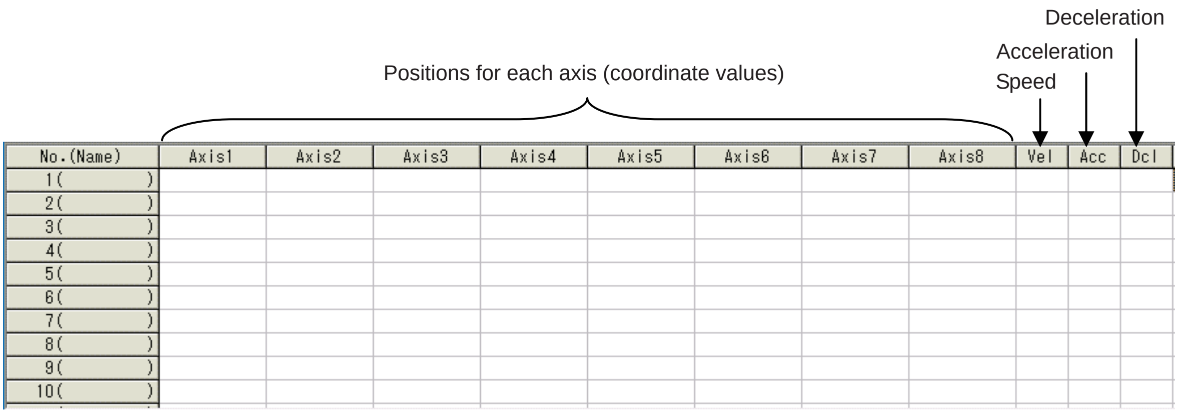
3.1.1 Position Table

The position data is to be input in the position table provided in the PC software.

(Note) • The edit window differs depending on the type of robot.
• The total number of settable positions differ depending on the controller

Controller	Position Total Number
XSEL-P/Q/PCT/QCT, PX/QX	20000
XSEL-R/S/RX/SX/RXD/SXD	53332 (1-axis specification) 40000 (2-axis specification) 32000 (3-axis specification) 26666 (4-axis specification) 22856 (5-axis specification) 20000 (6-axis specification) 17776 (7-axis specification) 16000 (8-axis specification)
XSEL-J/K/KE/KT/KET JX/KX/KETX	3000
SSEL	20000
ASEL/PSEL	1500
TT	3000
TTA	30000
MSEL	30000

[1] Single/rectangular axes, TT robots
Set positions (coordinate values), speeds, accelerations and decelerations in the position table and store the table in the controller.



- Axes 1 to 8: Position (coordinate value)
  Set the positions (coordinates) for all the connected axes (8 axes at maximum).
  The setting range varies depending on the actuator.
  The maximum range is from -99999.999 to 99999.999.

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Single/rectangular axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8

- Comment Column (shown on the right side of the position data)
  - Input is available up to 32 words with half width characters and 16 with full width at the maximum.
  - Comment can be added to Positions No. 1 to 10000.
  - Comment needs to be written to flash ROM after the data is transferred.

(Note 1) When having the software reset conducted or the power turned OFF without writing the position data to the flash ROM after a change is made to a comment, 22B "Position Data Comment Lost Error" will occur and the comment that a change was made will be deleted.
It is only the comment which is lost, and the program operation is available.

(Note 2) Comment is applicable also for PCLR (Position Data Clear) Command and PCPY (Position Data Copy) Command in SEL Program.
Therefore, in case that the position data with a comment being set is cleared by PCLR Command, or that a copy is made to the position with a comment being set in the area to copy from or area to copy to with PCPY Command, if the software reset is conducted or the power is turned OFF without the position data being written to the flash ROM, 22B "Position Data Comment Lost Error" will occur.
Please understand this well when using PCLR or PCPY.

- Vel (speed), Acc (acceleration), Dcl (deceleration)
  If Vel (speed), Acc (acceleration) and Dcl (deceleration) are set in the position data table, the values set in the position data table are given priority over the corresponding data set in the program.
  To make the Vel (speed), Acc (acceleration) and Dcl (deceleration) settings in the program effective, leave these fields blank.

The effective values of speed and acceleration are determined based on the priorities specified below.

Priority	Speed	Acceleration (Deceleration)
1	Value set in the position data table	Value set in the position data table
2	Value set in the VEL command	Value set in the ACC (DCL) command
3		All-axis parameter No. 11, "Default acceleration" (All-axis parameter No. 12, "Default deceleration")

The setting ranges vary depending on the actuator.

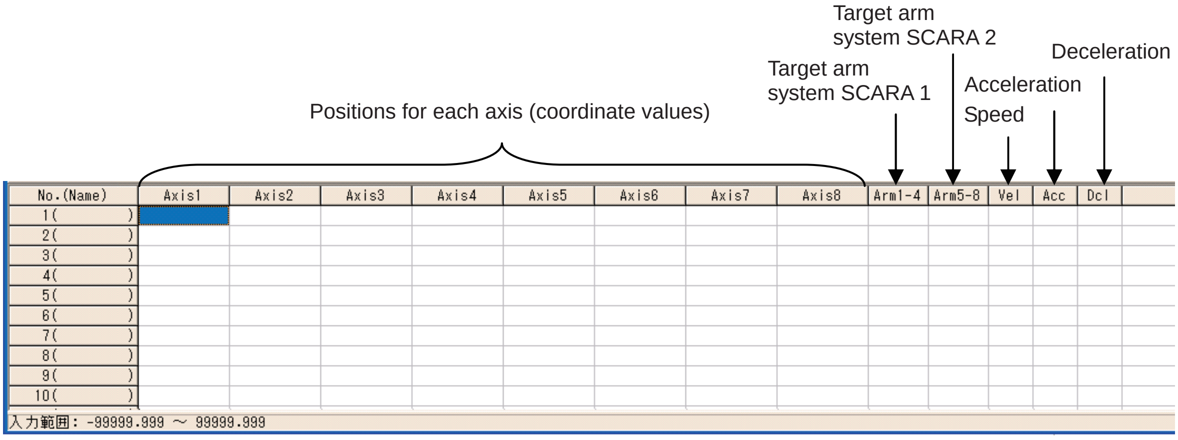
- Speed: 1 to the value of all-axis common parameter No. 21, “Maximum operating speed”
- Acceleration: 0.01 to the value of all-axis common parameter No. 22, “Maximum acceleration”
- Deceleration: 0.01 to the value of all-axis common parameter No. 23, “Maximum deceleration”

With rotational axes, values in mm are handled in degrees.
If Axis-specific parameter No. 1, “Axis operation type” is set to 1 (Rotational movement axis (angle control)), all values indicated in mm (including parameters, etc.) are converted to angles (in degrees).
These angles (in degrees) indicate angles (in degrees) of a rotating body at the end, as long as the gear ratio parameters (Axis-specific parameter No. 50, 51) are set correctly.

Example) Distance 1mm → 1deg
Speed 1mm/sec → 1deg/sec
Acceleration/deceleration 1G = 9807mm/sec^2
→ 9807deg/sec^2

[2] SCARA robots

Set positions (coordinate values), target arm system indications, speeds, accelerations and decelerations in the position table and store the table in the controller.



- Axes 1 to 8: Position (coordinate value)
  Set the positions (coordinates) for all the connected SCARA robots (8 axes at maximum).
  The setting range varies depending on the actuator.
  The maximum range is from -99999.999 to 99999.999.

SCARA robots 1				SCARA robots 2			
Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
X-axis	Y-axis	Z-axis	R-axis	X-axis	Y-axis	Z-axis	R-axis

- Arm1-4, Arm5-8: Target arm system indications (for XSEL-RX/SX/RXD/SXD, MSEL-PCX/PGX (Arm1-4) only)
  Indicate the arm system at PTP movement. (R: right arm system, L: left arm system)
  The indications become valid when:
  1) an operation is made with “Move” in the position data edit window or “Continuous Move” button, or
  2) when SEL program PTP movement command is executed using the position data

(Note 1) Unless otherwise the target arm system is indicated, the operation will be made with the current arm system.

(Note 2) The priority is put in the order as shown below to the indication of the target arm systems when PTP movement command is executed by SEL program.
1) Position data settings
2) SEL command (Settings for PTPR, PTPL, PTPD and PTPE)

(Note 3) Error No. B4D “Arm System Setting Error” would be issued if the current arm system and the target arm system are different at CP movement.

- Vel (speed), Acc (acceleration), Dcl (deceleration)
  If Vel (speed), Acc (acceleration) and Dcl (deceleration) are set in the position data table, the values set in the position data table are given priority over the corresponding data set in the program.
  To make the Vel (speed), Acc (acceleration) and Dcl (deceleration) settings in the program effective, leave these fields blank.
  Only for CP operation (*1) the speeds, accelerations and decelerations set in the position table become effective.
  (*1) CP operation is a type of operation in which the axes interpolates with one another to move to the target position. CP operation may be performed in the form of linear movement, arc movement, etc. If the axes do not interpolate with one another, such operation is called "PTP operation".

The effective values of speed and acceleration are determined based on the priorities specified below.

Priority	Speed	Acceleration (Deceleration)
1	Value set in the position data table	Value set in the position data table
2	Value set in the VEL command	Value set in the ACC (DCL) command
3		All-axis parameter No. 11, "Default acceleration" (All-axis parameter No. 12, "Default deceleration")

The setting ranges vary depending on the actuator.

- Speed: 1 to the value of all-axis common parameter No. 21, "Maximum operating speed"
- Acceleration: 0.01 to the value of all-axis common parameter No. 22, "Maximum acceleration"
- Deceleration: 0.01 to the value of all-axis common parameter No. 23, "Maximum deceleration"

With rotational axes, values in mm are handled in degrees.
If Axis-specific parameter No. 1, "Axis operation type" is set to 1 (Rotational movement axis (angle control)), all values indicated in mm (including parameters, etc.) are converted to angles (in degrees).
These angles (in degrees) indicate angles (in degrees) of a rotating body at the end, as long as the gear ratio parameters (Axis-specific parameter No. 50, 51) are set correctly.

Example)	Distance	1mm	→	1deg
	Speed	1mm/sec	→	1deg/sec
	Acceleration/deceleration	1G = 9807mm/sec^2		
			→	9807deg/sec^2

3.2 Program

Create a program using the “SEL Language” which is a proprietary language by IAI.

(Note) The number of programs and total number steps vary depending on the controller.

Controller	Number of programs	Total number of program steps
XSEL-P/Q/PCT/QCT/PX/QX/ R/S/RX/SX/RXD/SXD	128	9999
XSEL-J/K/KE/KT/KET JX/KX/KETX	64	6000
SSEL	128	9999
ASEL/PSEL	64	2000
TT	64	6000
TTA	256	9999
MSEL	256	9999

3.3 Program Format

[1] Program Edit

The program is to be input to the program edit window in the PC software.
The created program is to be transferred to the controller to be activated.

SEL language is translated by a step number (1 line) to another for the operation. Thus, it is unnecessary to compile (translate into the computer language).

No.	B	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
3					VEL	100			
4					ACC	0.3			
5					TAG	1			
6					EXSR	5			
7					MOVP	10			
8					MOVP	11			
9					TIMW	0.3			
10					EXSR	5			
11					MOVP	15			
12					EXSR	6			
13					TIMW	0.2			
14					MOVP	16			
15					VEL	300			
16					EXSR	1			
17					MOVP	20			

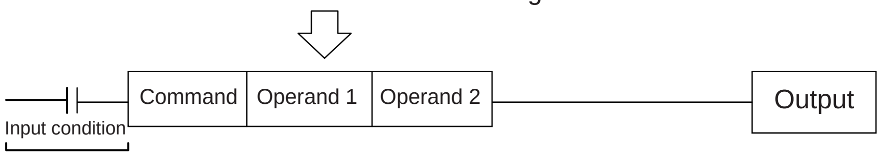
Step No.
Expansion Condition
Input Condition
Input Condition
Command and Declaration
Operand 1
Operand 2
Output

The program edit window is created in a program format (table format) and you are only to input the information for position, command, etc. to the appropriate areas.
In the following, explains about the program format.

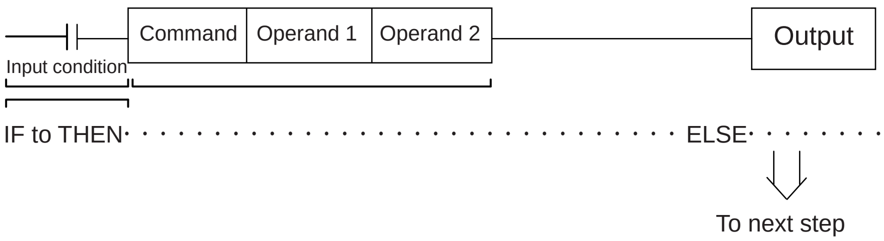
[2] Program Format

Extension condition (AND, OR)	Input condition (I/O, flag)	Command, declaration			Output (Output port, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand1	Operand2	Pst

The above is illustrated as follows in a ladder diagram.

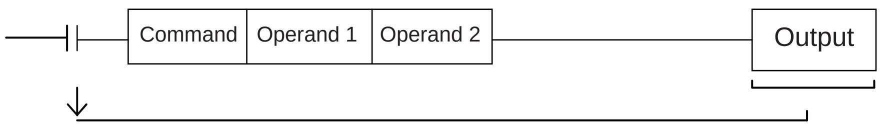


(1) The conditions in front of the command are equivalent to “IF to THEN” for the BASIC language.



1) The command will be executed if the input conditions meet the requirement. If there is an indication for the output, it turns the output port on, and if the input conditions do not meet the requirement, it moves to the next step.
2) If no condition is set, the command is executed unconditionally.
3) To use the condition in reverse (using the so-called contact b logic), add “N” (NOT) to the condition.
4) You can use an input port, output port or flag for the input condition.
5) Operands 1 and 2, and the output, can be specified indirectly.

(2) Operation of the output, specified after the command and operands 1 and 2, is explained below.



1) In the case of an actuator operation control command, etc., the output turns OFF the moment the command execution is started and turns ON when the execution is completed. In the case of a calculation command, etc., the output turns ON when the result becomes a certain value and turns OFF with other values.
2) You can use an output port or flag for the output.

[Application] Extended condition

You can combine extended conditions in a complex manner using the AND gate and OR gate.

(Example)

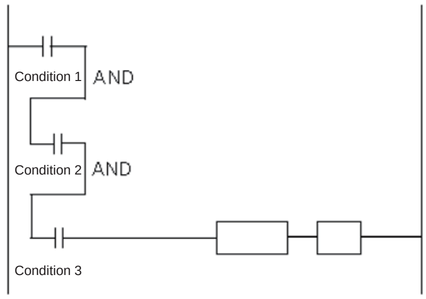
[Extension by AND]

Combination of A (AND) and A (AND)

(SEL language)

Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
A	Condition 2				
A	Condition 3	Command	Operand 1	Operand 2	

(Ladder diagram)



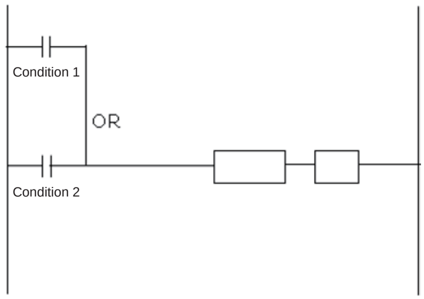
[Extension by OR]

Combination of O (OR) and O (OR)

(SEL language)

Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
○	Condition 2	Command	Operand 1	Operand 2	

(Ladder diagram)



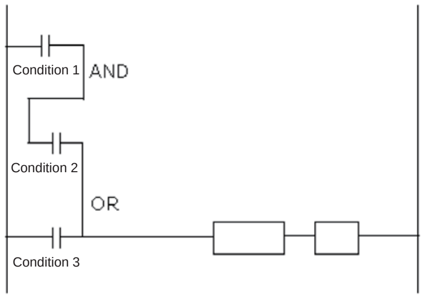
[Extension by AND and OR]

Combination of A (AND) and O (OR)

(SEL language)

Extension condition	Input condition	Command			Output
		Command	Operand 1	Operand 2	
	Condition 1				
A	Condition 2				
○	Condition 3	Command	Operand 1	Operand 2	

(Ladder diagram)



3.4 Relationship of program and position table

In the case of a movement command such as MOVL, set a position number in operand 1. Some commands such as ARCH (arch motion) require a position number to be set in operand 2, as well.
The position corresponding to the position number set in the position table is referenced and the actuator moves to the applicable position.

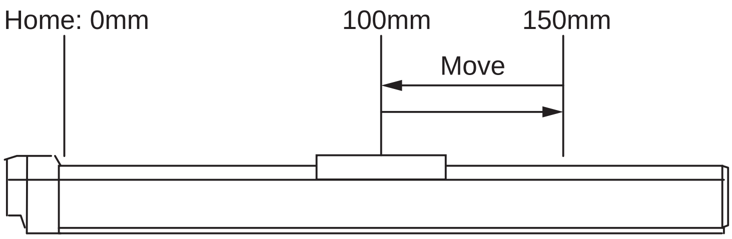
Program format

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					HOME	1			原点復帰
2					VEL	100			速度設定100mm/sec
3					TAG	1			
4					MOVL	1			P1移動
5					MOVL	2			P2移動
6					GOTO	1			
7									

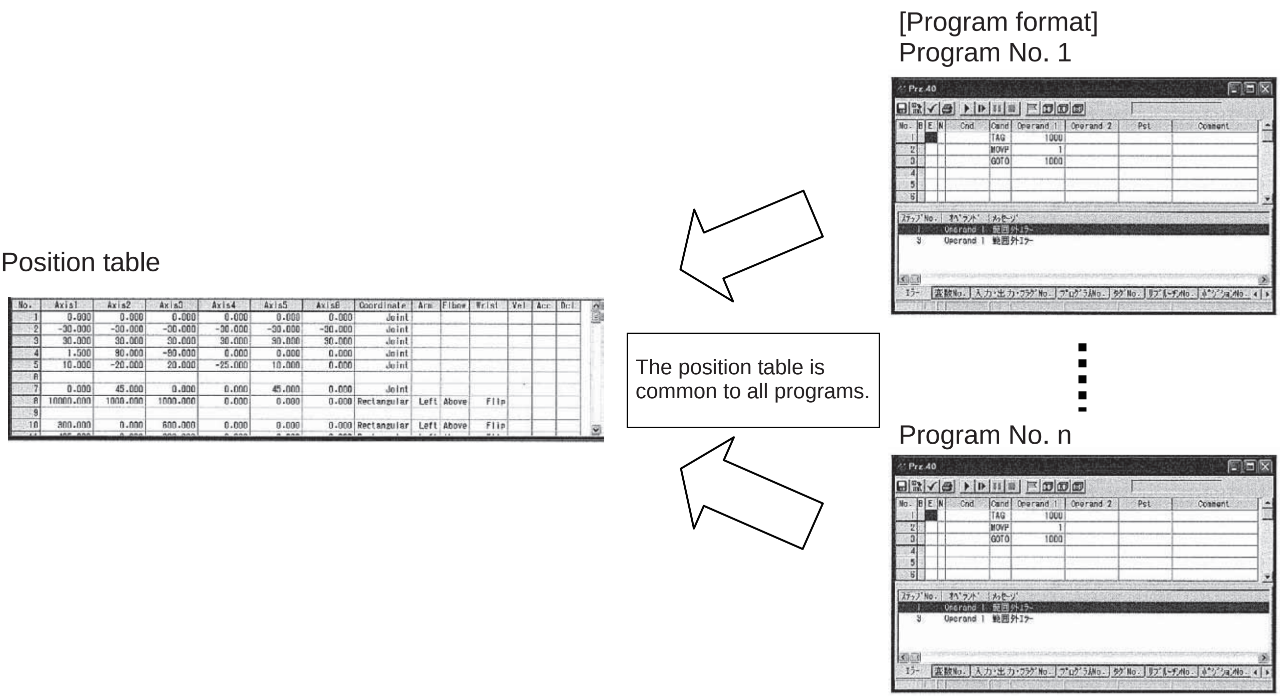
Position table

No.	Axis1
1	100.000
2	150.000
3	
4	
5	
6	
7	

In the above example, the actuator moves to the positions at 100mm corresponding to position No. 1 and 150mm corresponding to position No. 2.



The position table is a single table that can be referenced from all programs. In the example below, the standard position table is used. A different table is used if the controller has a gateway function.
(Note) For RC Gateway Function, a different position table is to be used.
[Refer to the Instruction Manual for XSEL Controller P/Q/PX/QX RC Gateway Function.]



No.	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Coordinate	Arm	Elbow	Wrist	Vel	Acc	Dcl
1	0.000	0.000	0.000	0.000	0.000	0.000	Joint						
2	-30.000	-30.000	-30.000	-30.000	-30.000	-30.000	Joint						
3	30.000	30.000	30.000	30.000	30.000	30.000	Joint						
4	1.500	80.000	-80.000	0.000	0.000	0.000	Joint						
5	10.000	-20.000	20.000	-25.000	10.000	0.000	Joint						
6													
7	0.000	45.000	0.000	0.000	45.000	0.000	Joint						
8	10000.000	1000.000	1000.000	0.000	0.000	0.000	Rectangular	Left	Above	Flip			
9													
10	300.000	0.000	600.000	0.000	0.000	0.000	Rectangular	Left	Above	Flip			

3.5 Basic Stage (Program creation and position table creation)

In this section, explains how to create a program for the basic operation patterns.

3.5.1 Home Return and Home Return Completion Signal

[1] Description

Output a signal to confirm completion of homing (incremental specification).
With the controller, a home return completion signal can be output using an I/O parameter.
However, the following explains how to output a home return completion signal within a program using a general-purpose output.
Once turned ON, a general-purpose output will remain ON even after the current program ends or other program is started. (It will turn OFF upon emergency stop, etc., but the ON status can be maintained using an I/O parameter (I/O parameter No. 70 and 71).)
(Note) SCARA robots do not require home return operation.

[2] Example of Use

a. Output a home return completion signal.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			HOME	11		
			BTON	303		

Execute homing. (1st and 2nd axes)
Output 303 is turned ON when home-return operation is complete

b. Use a home return completion signal to make sure the actuator will not perform homing if it has already been performed.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
	N	303	HOME	11		
			BTON	303		

If output 303 is OFF (NOT);

Execute homing if output 303 is OFF.
Home-return complete output
(turn Output 303 ON)

c. Use the output field instead of a BTON command.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
	N	303	HOME	11		303

Output section

The same process as Example b. is proceeded.

[3] Reference

Output port No. 304 can be used as a home return completion output (dedicated output) by setting I/O parameter No. 50 to “2”.

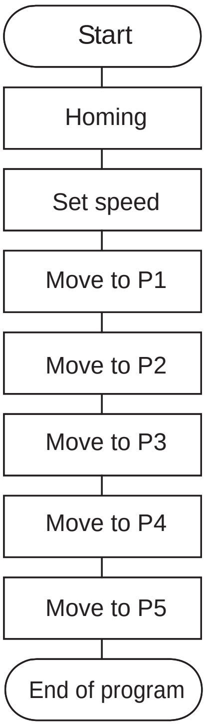
3.5.2 Positioning Operation (Moving position)

[1] Description

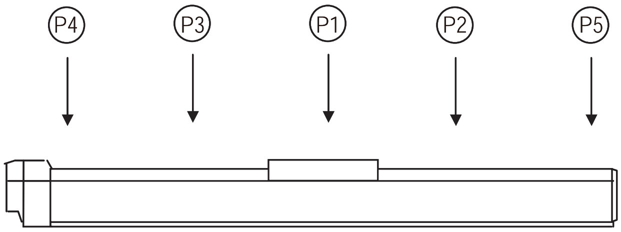
Move the actuator to positions 1 through 5 at a speed of 100mm/sec after homing.

[2] Example of Use

Flowchart



- Homing must be performed and a speed must be set, before the actuator can be operated.
- The actuator moves to the position data coordinates specified by the respective move commands.
- With the absolute specification, homing (HOME command) is not required.



Program (Example)

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					HOME	1			1軸原点復帰
2					VEL	100			速度設定100mm/sec
3					MOVL	1			ポジションデータNo.1に移動
4					MOVL	2			ポジションデータNo.2に移動
5					MOVL	3			ポジションデータNo.3に移動
6					MOVL	4			ポジションデータNo.4に移動
7					MOVL	5			ポジションデータNo.5に移動
8					EXIT				ﾌﾟﾛｸﾞﾗﾑ終了
9									

Position data (Example)

No.	Axis1
1	100.000
2	150.000
3	50.000
4	0.000
5	200.000
6	
7	
8	
9	

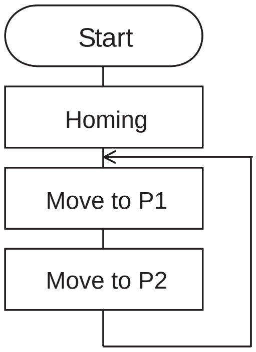
3.5.3 Moving Back and Forth between Two Points

[1] Description

Moves back and forth between two points.

[2] Example of Use

Flowchart



- The actuator moves back and forth between P1 and P2 indefinitely.
- Use of only 1 axis is assumed.
- Enter TAG in the first of the steps to be repeated, and enter GOTO in the last of the steps to be repeated.

Program (Example)

No.	B	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1					HOME	1			原点復帰
2					VEL	100			速度設定100mm/sec
3					TAG	1			
4					MOVL	1			ポジションデータNo.1に移動
5					MOVL	2			ポジションデータNo.2に移動
6					GOTO	1			TAG1 にジャンプ
7									

Position data (Example)

No.	Axis1
1	100.000
2	150.000
3	
4	
5	
6	
7	

3.5.4 Repeated Operation

[1] Description

Use GOTO and TAG commands to repeat the same operation within the program or to jump to a desired step if a condition is satisfied. A TAG command can be written in a step either before or after a GOTO command.

[2] Example of Use

Example 1

Repeat the same operation. (Move to Position 1 → to position 2 → to position 1 …)

```
        :
        TAG   1
Repeated.  MOVP  1  } Steps to be repeated
        MOVP  2
        GOTO  1
```

Example 2

Skip steps.

```
        :
        GOTO  2
Jump.   MOVP  1  } Steps to be ignored
        MOVP  2
        TAG   2
```

3.5.5 PATH Operation (Continuous operation among multiple positions)

[1] Description

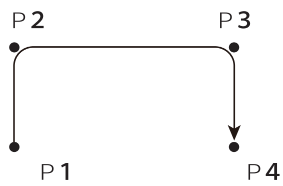
This function moves the robot continuously among 4 arbitrary points. (PATH movement)

[2] Example of Use

The actuator moves along the path shown at right, without stopping at P2 and P3.
Compared with MOVP and MOVL, this command does not require the actuator to position exactly at P2 and P3, and thus the movement tact time can be reduced.
Assume the following command is executed when the actuator is stopped at P1:

PATH 2 4

The actuator will move from P1 to P4 by passing points near P2 and P3.

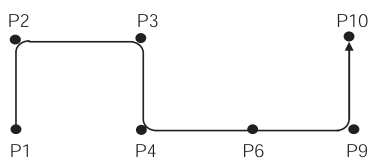


Even if “PATH 2 3” and “PATH 4 4” are input successively, the actuator will still move in the same way as when “PATH 2 4” is input.

If “PATH 4 1” is executed while the actuator is stopped at P4, the actuator will move along the same path in the opposite direction (P4 → P3 → P2 → P1).

It is possible to move through discontinuous positions or move continuously by passing the same position.

PATH	1	4	
PATH	6	6	discontinuous position
PATH	9	10	



As shown above, specify the number corresponding to the discontinuous position, or No. 6, for both the start position number and end position number in the PATH command. The actuator moves continuously in the sequence of position Nos. P1 → P2 → P3 → P4 → P6 → P9 → P10.

[3] Example of Use

Refer to the page for “PATH” Command in [12] Actuator Control Command for each command language for the caution in use.

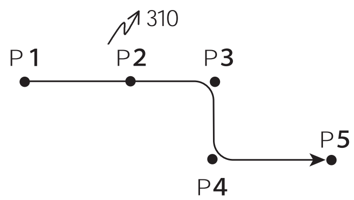
3.5.6 External Signal Output during Path Movement

[1] Description
Output signals while the actuator is moving with a PATH command.

[2] Example of Use
Before executing a PATH command, declare a POTP command to specify signal output during movement.
If a given output or global flag is specified in the output field of the PATH command, the output or flag specified in the output field will turn ON as the actuator approaches, via path movement, the position specified in the PATH command.

Example 1

The actuator moves from P1 to P5 along the positions shown at right, without stopping. As the actuator approaches P2, output port 310 turns ON.



Cmnd	Operand 1	Operand 2	Pst
VEL	100		
POTP	1		
PATH	1	1	
PATH	2	2	310
PATH	3	5	

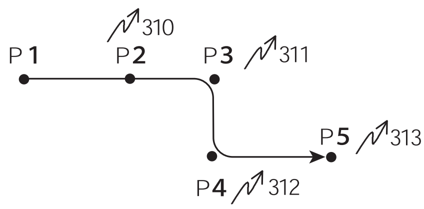
← A declaration command to specify signal output during path movement.
← 310 turns ON when the actuator approaches P2 specified in this step.

Example 2

Outputs 310 to 313 can be turned ON sequentially at the respective points of P2 to P5.

Cmnd	Operand 1	Operand 2	Pst
VEL	100		
POTP	1		
PATH	1	1	
PATH	2	5	310

← A declaration command to specify signal output during path movement.
← 310 to 313 turn ON sequentially at P2 to P5 specified in this step.



(Note) This command is able only to output and to turn the flag ON. The output or flag that was turned ON during path operation must be turned OFF (using a BTOF command) after the operation is completed.

[3] Example of Use
Refer to the page for "PATH" Command in [12] Actuator Control Command for each command language for the caution in use.

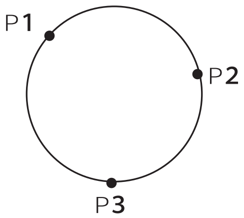
3.5.7 Circle/Arc Operation

[1] Description
The actuator moves along a two-dimensional circle or arc.

[2] Example of Use
To specify a circle, specify three points the actuator will pass. To specify an arc, specify the starting point, passing point and end point.

Example 1

Circle



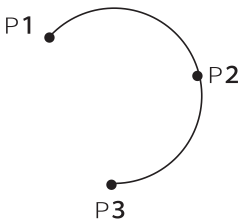
- Specify "CIR2 2 3" after the actuator has moved to P1.
- If "CIR2 2 3" is specified in the figure shown at left, the actuator will move along this circle clockwise.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			VEL	100		
			MOVP	1		
			CIR2	2	3	

- To cause the actuator to move counterclockwise, specify "CIR2 3 2".

Example 2

Arc



- Specify "ARC2 2 3" after the actuator has moved to P1.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			VEL	100		
			MOVP	1		
			ARC2	2	3	

[3] Reference
1) Some circle and arc commands can be executed not only two-dimensionally (between two actuator axes) but also three dimensionally (among three actuator axes).

   CIRS ·········· Move along circle three-dimensionally
   ARCS ·········· Move along arc three-dimensionally

2) Refer to the page for "CIR2, ARC2, CIRS, ARCS" Command in [12] Actuator Control Command for each command language for the caution in use.

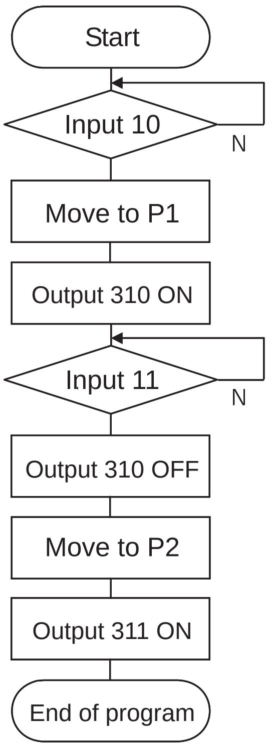
3.5.8 Axis Movement by External Signals and Output of Completion Signal to External Device

[1] Description

This is a function to enable the axes movement with an external signal input and to output the complete signal to an external device.

[2] Example of Use

Flowchart



Wait for the input port (external signal) 10 to turn ON, and then move to P1 (Position Data No. 1).
Wait for the input port (external signal) 11 to turn ON, and then move to P2 (Position Data No. 2).
The movement complete signal for P1 is output to the output port 310, and P2 complete signal to port 311.

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			速度100mm/sec設定
			WTON	10			入力10 ON待ち
			MOVP	1			P1移動
			BTON	310			出力310 ON
			WTON	11			入力11 ON待ち
			BTOF	310			出力310 OFF
			MOVP	2			P2移動
			BTON	311			出力311 ON
			EXIT				ﾌﾟﾛｸﾞﾗﾑ終了

3.5.9 Changing the Moving Speed

[1] Description
Change the moving speed.

[2] Example of Use
The speed can be set using the following two methods:
a: Use a VEL command within the program
b: Use a speed setting in the position table

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			MOVP	1		P1移動
			VEL	1000		速度1000mm/sec設定
			MOVP	2		P2移動
			MOVP	3		P3移動
			VEL	50		速度50mm/sec設定
			MOVP	4		P4移動

Position data (Example)

No.	Axis1	Vel	Acc	Dcl
1	100.000	100		
2	200.000	500		
3	300.000			
4	400.000			

Each Position (Position Data No. 1 (P1) to No. 4 (P4))

Moving speeds in the above program
Position at 100mm (P1) … The actuator moves at 100mm/sec.
Position at 200mm (P2) … The actuator moves at 500mm/sec.
Position at 300mm (P3) … The actuator moves at 1000mm/sec.
Position at 400mm (P4) … The actuator moves at 50mm/sec.

If a speed is specified in the position data table, this speed takes precedence over the speed specified in the application program, as shown above.

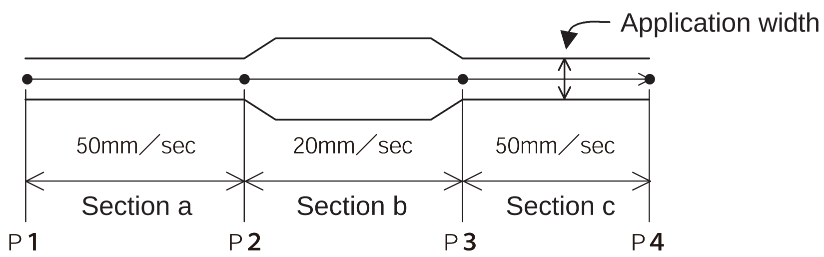
3.5.10 Speed Setting Change during PATH (Continuous) Operation

[1] Description

You can change the speed of the actuator without stopping it, by using a PATH command and VEL fields of the position table.
For example, this command is useful in a paint dispensing application where the application volume changes in the middle.

[2] Example of Use

The actuator moves through linear sections a, b and c at 50mm/sec, 20mm/sec and 50mm/sec, respectively, without stopping (PATH operation).



Position data (Example)

No.	Axis1	Vel	Acc	Dcl
1	0.000	50		
2	100.000	50		
3	200.000	20		
4	300.000	50		

Program (Example)
"PATH 1 4" is the only movement command required.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			PATH	1	4	

[3] Reference

The speed can also be changed from other program using a CHVL (speed change) command.

3.5.11 Variables and Flags [Global/Local]

[1] Description

The internal variables and flags used in the SEL language are classified into local and global types.
The data range used commonly by all programs is called the global range, while the data range used only by each program is called the local range. When multi-tasking programs are run simultaneously, the global range must be used to synchronize the programs and allow cross-referencing of variables among the programs.

[2] Example of Use

Program handshake

Program A

Cmnd	Operand 1	
MOVL	1	Turn ON Output 600 →
BTON	600	
WTON	601	Waiting for Output 601 to turn ON
MOVL	3	

Program B

Cmnd	Operand 1	
WTON	600	Waiting for Output 600 to turn ON
MOVL	2	
BTON	601	Turn ON Output 601

Use of global flags with the above two programs permits handshake between the programs, and the actuator moves per “MOVL 1” in program A, moves per “MOVL 2” in program B, and then move per “MOVL 3” in program A, for example.

Backup in Battery

The XSEL controller (Note 1) has a built-in battery for retaining variables and flags used in the programs. The data is retained in the global domain for both variables and flags even if the power is turned OFF.
The variables and flags in the local range are cleared when the program is started (the variables are reset to “0”, while the flags turn OFF).

Note 1: XSEL-R/S/RX/SX/RXD/SXD do not have a system memory backup battery since they possess the global domain in the non-volatile memory.

The system-memory backup battery is optional for ASEL, PSEL and SSEL controllers.
No system-memory backup battery is available for TT robots.
If the system memory is not backed up with a battery, global areas will be cleared once the power is cut off.

3.5.12 How to Use Subroutines

[1] Description

A subroutine is a group of steps that are called and executed several times within a program. Subroutines are used to reduce the number of program steps and make the program easy to read. Up to 99 subroutines can be used in one program. Up to 15 subroutine calls can be nested.

[2] Example of Use

Declare/call subroutines using the following commands:

EXSR: Call a subroutine
BGSR: Declare the start of a subroutine (start of a group of steps)
EDSR: Declare the end of a subroutine (end of a group of steps)

Cmnd	Operand 1
VEL	100
MOVL	1
BTON	310
WTON	20
BTOF	310
MOVL	2
BTON	310
WTON	20
BTOF	310
MOVL	3
BTON	310
WTON	20
BTOF	310
EXIT	

The same tasks are consolidated into a single location.

Cmnd	Operand 1
VEL	100
MOVL	1
EXSR	1
MOVL	2
EXSR	1
MOVL	3
EXSR	1
EXIT	
BGSR	1
BTON	310
WTON	20
BTOF	310
EDSR	

Subroutine

[3] Note

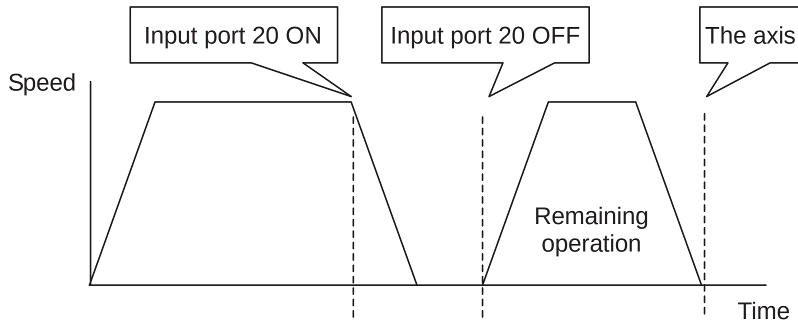
Jumping from within a subroutine to a TAG position outside the subroutine using a GOTO command is prohibited.

3.5.13 Pausing the Operation

[1] Description
Use a declaration command HOLD to pause the moving axis temporarily via external input.

[2] Example of Use
A pause interruption operation can be executed to a moving axis (to decelerate the axis to a stop) by declaring a HOLD command within the program.
While HOLD is input, the actuator pauses (decelerates to a stop, if currently moving) against all moving commands in the same program.

HOLD 20 A declaration to execute pause if general-purpose input 20 turns ON.



Application

You can specify a global flag, instead of an input port, in Operand 1 of the HOLD command.
Use of a global flag allows the actuator to be paused from other program.
The input signal pattern and stop action can be selected using Operand 2.

0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)
2 = Contact b (Decelerates to a stop, and then servo OFF
⇒ The drive power is not cut off.)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			HOLD	20	2		SVOF入力20 B接点

[3] Note
If the actuator is paused during homing, it will start the homing sequence from the beginning upon restart.

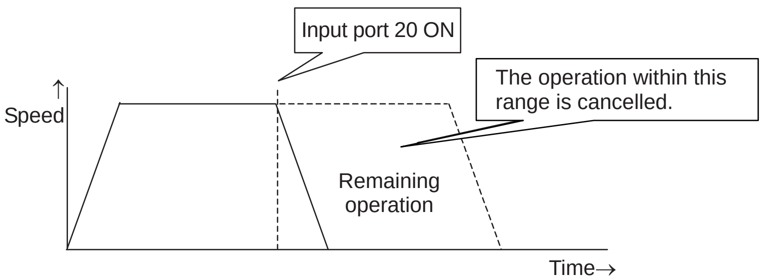
3.5.14 Canceling the Operation

[1] Description
Use a declaration command CANC to decelerate the moving axis to a stop and cancel the remaining operation.

[2] Example of Use
While CANC is input, all movement commands in the same program are cancelled.

CANC command
```
CANC   20   Cancel the movement commands if input port 20 turns ON (declaration).
    :
MOVP    1
MOVP    2
    :
WTON   21
    :
```

* Declare this command in a step before the movement commands you want to cancel.
* While CANC is input, all operation commands are cancelled sequentially, while tasks other than operation commands (such as I/O processing and calculation processing) are executed sequentially.



Application

A desired input signal pattern can be selected for a CANC command using Operand 2.
0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			CANC	20	1		ｷｬﾝｾﾙ入力20 B接点

[3] Note
It is recommended that you use a WTON command to create an input waiting step, because otherwise you cannot specify which of the program steps the actuator is currently executing.

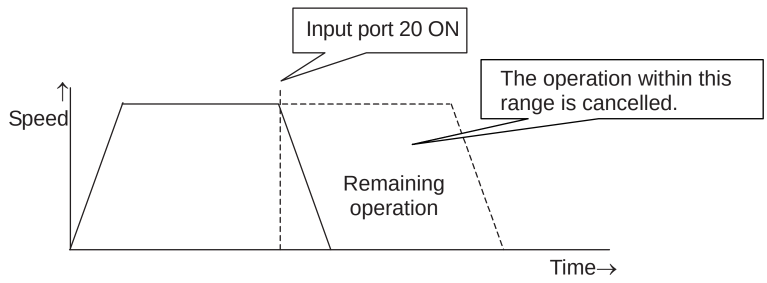
3.5.15 Aborting from Other Program

[1] Description

Decelerate the moving axis to a stop and cancel the remaining operation. (STOP)

[2] Example of Use

Execute a STOP command from other program to forcibly stop the operation (in the multi-tasking mode). Specify the axis you want to stop using an axis pattern.



Example 1

STOP command

Main program

```
EXPG    n    The stop program starts.
  :
MOVL    1
MOVL    2
  :
```

Stop control program

```
WTON    20    Wait for stop input.
STOP    11    Axes 1 and 2 stop.
```

If “STOP 11” is executed while “MOVL 1” is being executed, “MOVL 1” will be cancelled and the actuator will continue its operation from “MOVL 2”.

Example 2

Main program

```
EXPG    n    The stop program starts.
  :
MOVP    1
MOVP    2
  :
```

Stop control program

```
WTON    20    Wait for stop input.
STOP    10    Axis 2 stops.
```

If “STOP 10” is executed while “MOVL 1” is being executed, only the axis 2 part of “MOVL 1” will be cancelled. Both axes 1 and 2 will operate under “MOVL 2”.

[3] Note

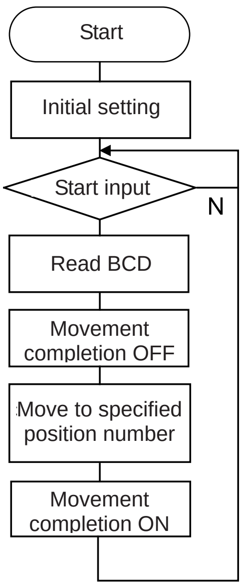
If a STOP command is executed during a CP operation (interpolation operation) initiated by MOVL, etc., the operations of all axes will be cancelled regardless of the axis pattern specified in the STOP command.

3.5.16 Operation by Position Number Specification via External Signals and Output of Completion Signal to External Device

[1] Description
Load externally input BCD codes as position numbers to execute movements.

[2] Example of Use
Use an INB command to load a position number as a BCD code from an input port.
A position number can be specified using a value consisting of up to three digits.

Flowchart



Input assignment*1

Port	Description
1	Start input
15	Position specification 1
16	Position specification 2
17	Position specification 4
18	Position specification 8
19	Position specification 10
20	Position specification 20
21	Position specification 40
22	Position specification 80
23	Position specification 100
24	Position specification 200
25	Position specification 400
26	Position specification 800

Output

303 Movement completion

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			速度設定
			TAG	1			GOTOの飛び先
			WTON	1			ｽﾀｰﾄ入力待ち
			INB	15	3		ﾎﾟｼﾞｼｮﾝNo．読取り
			BTOF	303			移動完了信号OFF
			MOVL	*99			ﾎﾟｼﾞｼｮﾝNo．へ移動
			BTON	303			移動完了信号ON
			GOTO	1			TAG1へｼﾞｬﾝﾌﾟ

*1 Shown above is an example of port assignment for XSEL, ASEL, PSEL and SSEL controllers.
An example for TT robots is shown below.

Input assignment

Port	Description
28	Start input
16	Position specification 1
17	Position specification 2
18	Position specification 4
19	Position specification 8
20	Position specification 10
21	Position specification 20
22	Position specification 40
23	Position specification 80
24	Position specification 100
25	Position specification 200
26	Position specification 400
27	Position specification 800

3.5.17 Operation by Coordinate Value Input via External Signals and Output of Completion Signal to External Device

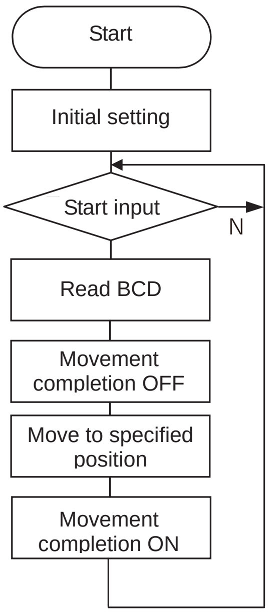
[1] Description

Receive target position data as absolute values from a host device to execute movements.

[2] Example of Use

Use an INB command to load position data as a BCD code from an input port.
Each BCD value should consist of four digits, with the last digit indicating a decimal place.
The moving axis is axis 1.
Example: If a BCD of “1234” is received, the axis will move to the position at 123.4mm.

Flowchart



Input assignment

Port	Description
1	Start input
15	0.1mm
16	0.2mm
17	0.4mm
18	0.8mm
19	1mm
20	2mm
21	4mm
22	8mm
23	10mm
24	20mm
25	40mm
26	80mm
27	100mm
28	200mm
29	400mm
30	800mm

Output

303 Movement completion

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	コメント
			HOME	11			原点復帰
			VEL	100			速度設定
			TAG	1			GOTOの飛び先
			WTON	1			スタート入力待ち
			INB	15	4		移動位置読み取り
			LET	199	*99		小数点付けの為実数変数にコピー
			DIV	199	10		小数点付けの為10で割る
			PPUT	1	1000		ポジションナンバ1000の1軸目にデータ代入
			BTOF	303			移動完了信号オフ
			MOVL	1000			代入された位置に移動
			BTON	303			移動完了信号オン
			GOTO	1			TAG1にジャンプ

[3] Note

With TT robots, 16 general-purpose inputs from 016 to 0.31 can be input. However, if coordinate values from 400mm [maximum TT stroke] to 0.0mm are inputs in 0.1mm increments and a start signal is used to start the movement, as in the example, all of the 16 general-purpose inputs are used. Take note that in this case you can no longer use general-purpose inputs for other purposes.

3.5.18 Output of Current Position Coordinate Value to External Device

[1] Description

Read the current actuator coordinate in real time and output the coordinate from an output port as BCD data.

[2] Example of Use

Use a PRDQ command to read the current coordinate value of axis 1.
Output the current coordinate data of axis 1 every 0.2sec as BCD output.
The output range is from 0.00 to 999.99mm.

BCD output assignment

Output port No.	Description	Output port No.	Description
324	0.01	337	20
325	0.02	338	40
326	0.04	339	80
327	0.08	340	100
328	0.1	341	200
329	0.2	342	400
330	0.4	343	800
331	0.8		
332	1		
333	2		
334	4		
335	8		

Unit: mm

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	コメント
			TAG	1			
			PRDQ	1	101		1軸現在位置を変数101へ
			MULT	101	100		少数点以下第三位目以下切り捨て
			LET	99	*101		整数変数にコピー
			OUTB	324	5		BCDで5桁分出力
			TIMW	0.2			サンプリングタイム
			GOTO	1			

* The current position coordinate is written to variable 101 according to the PRDQ command.
Since the value that has been read into the variable is in the xxx.xxx format, move the unused digits to below the decimal point so that the result can be output as BCD data.
In this example, the third and subsequent decimal places are not required and thus the value is multiplied by 100 to obtain the data xxxxx.x.
Next, the BCD output data is copied to dedicated variable 99.
The digit below the decimal point is rounded off at this time.
Then, the final value is output to an external device via an OUTB command.
This program is run in the multi-tasking mode as a sub-program.

[3] Note

The unit of output data may have to be changed as deemed appropriate depending on the moving range of the actuator and number of available output ports.
If coordinate values from 0mm to 400mm [maximum TT stroke] are output in 0.01mm increments, as in the example, 19 general-purpose outputs are needed. However, TT robots only have 16 general-purpose outputs of 316 to 331. Accordingly, you must take an appropriate action such as changing the unit of output data to 0.1mm.

3.5.19 Conditional Jump

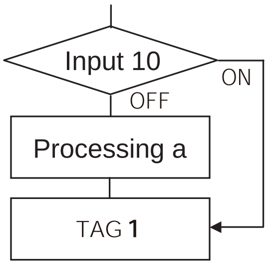
[1] Description

Select the destination to jump to via GOTO using the external input, output and/or internal flag statuses as a condition.
Process is switched over for each input.

[2] Example of Use

Example 1

If input 10 turns ON, the actuator will jump to TAG 1. If it turns OFF, the actuator will proceed to the next processing.



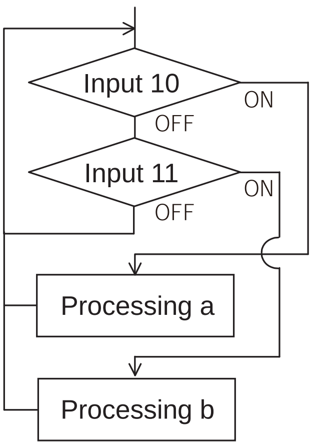
E	N	Cnd	Cmnd	Operand 1
		10	GOTO	1
Processing a				
			TAG	1
Processing b				

Execute GOTO 1 if input 10 turns ON.

* If input 10 turns ON, the actuator will skip processing a and perform processing b.
If input 10 turns OFF, the actuator will perform processing a, and then perform processing b.

Example 2

Wait for the input to the two ports 10 and 11, and if Input 10 becomes ON, proceed to the processing a, and proceed to the processing b if Input 11 becomes ON.



E	N	Cnd	Cmnd	Operand 1
			TAG	1
		10	GOTO	2
		11	GOTO	3
			GOTO	1
			TAG	2
Processing a				
			GOTO	1
			TAG	3
Processing b				
			GOTO	1

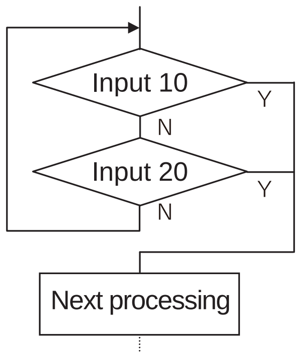
No input.
Input 10 turns ON.
Input 11 turns ON.

If both inputs 10 and 11 turn ON, the actuator will perform processing a.

3.5.20 How to Pause and Then Resume Program after Output Signal Input

[1] Description
The controller waits for multiple different inputs and performs processing upon reception of any of these inputs.

[2] Example of Use
Inputs 10 and 20 are monitored, and the actuator will proceed to the next step when either input is received (OR logic).



Program a

E	N	Cnd	Cmnd	Operand 1
			TAG	1
		10		
O		20	GOTO	2
			GOTO	1
			TAG	2

Next processing

Program b

E	N	Cnd	Cmnd	Operand 1
			TAG	1
	N	10		
A	N	20	GOTO	1

Next processing

* Both programs a and b perform the same processing.

As shown in the sample, the controller waits for input without using a WTON command.
This method can also be used when multiple input conditions must be combined.

[3] Note
With a WTON command, the program cannot wait for multiple inputs because processing will resume upon receipt of one of the specified inputs.

3.5.21 How to Use Offset

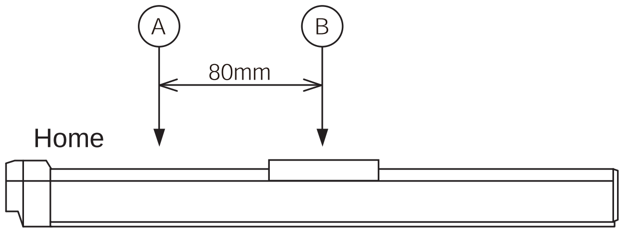
[1] Description

With an OFST command, an offset can be specified for position data when you want to shift (offset) all teaching points by several millimeters because the actuator was not installed exactly in the specified position or for other reasons.

[2] Example of Use

Move the actuator from point A to point B, which is offset by 80mm from point A.

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			
			MOVP	1			A点へ移動
			OFST	1	80		1軸目80mmｵﾌｾｯﾄ
			MOVP	1			B点へ移動



[3] Note

Once an offset has been set, the offset applies to all movement commands executed thereafter. To cancel the offset, execute an offset command again by specifying 0mm. An offset does not apply to other programs (even in the multi-tasking mode). If a given offset must be applied to all programs, it must be set for all programs individually.

3.5.22 How to Repeat Specified Operation Multiple Times

[1] Description
Execute a specific operation n times.

[2] Example of Use
The actuator moves back and forth between P1 and P2 ten times, and then the program ends.
Use a CPEQ command to compare the number of times the movement has been actually repeated, against 10.
It is assumed that homing has been completed.

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			VEL	100			速度設定
			LET	1	0		変数ｸﾘｱ
			TAG	1			
			MOVP	1			P1へ移動
			MOVP	2			P2へ移動
			ADD	1	1		変数1に1加算
			CPEQ	1	10	900	繰返し回数確認
	N	900	GOTO	1			10回未満ならTAG1へ
			EXIT				ﾌﾟﾛｸﾞﾗﾑ終了

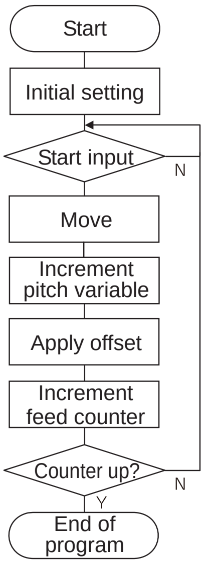
[3] Reference
The same operation can also be performed using a DWEQ command.

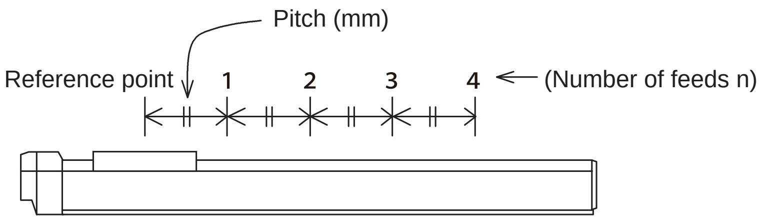
3.5.23 Constant Feed Operation [Pitch Feed]

[1] Description
Feed the actuator by a specified pitch n times from a reference point.
The pitch and number of repetitions are specified by variables in advance.

[2] Example of Use
Use an OFST command to perform pitch feed.
The number of times the actuator has been fed is counted by a counter variable.
The X-axis is fed in the positive direction.

Flowchart





Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
			LET	1	4		送り回数(n=4)代入
			LET	100	80		ﾋﾟｯﾁ(80mm)代入
			LET	2	0		変数ｸﾘｱ(ｶｳﾝﾀ)
			LET	101	0		変数ｸﾘｱ(ｵﾌｾｯﾄ値)
			HOME	1			原点復帰
			VEL	100			速度設定
			TAG	1			
			WTON	1			ｽﾀｰﾄ入力待ち
			MOVP	1			移動
			ADD	101	*100		ｵﾌｾｯﾄ値にﾋﾟｯﾁ加算
			OFST	1	*101		X軸ｵﾌｾｯﾄ処理
			ADD	2	1		ｶｳﾝﾀ用変数に＋１
			CPGT	2	*1	900	送り終了確認
	N	900	GOTO	1			未完了なら繰返す
			EXIT				ﾌﾟﾛｸﾞﾗﾑ終了

[3] Note
An OFST command applies to movement commands.
Executing an OFST command alone does not move the axis.

[4] Reference
Pitch feed can also be performed with MVPI and MVLI Commands.

3.5.24 How to JOG via External Signal Input

[1] Description

The slider moves forward or backward while an input is ON or OFF.
Instead of an input, an output or global flag can be used as a cue.
The slider will move directly to the next step if the specified input does not satisfy the condition when the command is executed.
Regardless of the input status, the slider will stop upon reaching the soft limit, and the command in the next step will be executed.

[2] Example of Use

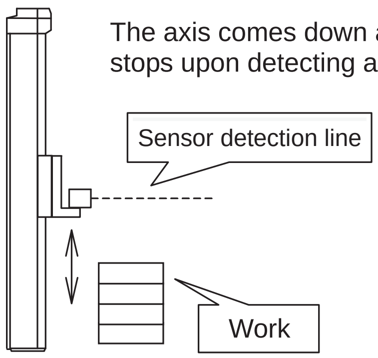
- Explanation of commands

JFWN	1	20	Axis 1 moves forward while input 20 is ON.
JFWF	1	21	Axis 1 moves forward while input 21 is OFF.
JBWN	10	22	Axis 2 moves backward while input 22 is ON.
JBWF	10	23	Axis 2 moves backward while input 23 is OFF.

Example 1

- Stop the axis when a sensor input is received.

The axis comes down and stops upon detecting a work.



:

VEL	50		Specify a low speed.
JFWF	1	20	Move until a sensor input (20) is received.
EXIT			The program ends.

Example 2

- Cause the actuator to jog just like in teaching pendant operation (2 axes are operated).

Program (Example)

E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
			TAG	1		
			JFWN	1	20	
			JBWN	1	21	
			JFWN	10	22	
			JBWN	10	23	
	N	24	GOTO	1		
			EXIT			

[3] Note

HOLD, STOP and CANC commands remain valid while the actuators are jogging.

3.5.25 Switching Programs

[1] Description
Use EXPG/ABPG commands to switch programs using a program.

[2] Example of Use

Example 1

Start program 2 once the processing of program 1 is completed, and then end program 1.

Program 1	Program 2
:	:
EXPG 2	:
EXIT	

Example 2

Start a program via an external signal, and then end the other program.

Program 1	Program 2
ABPG 2	ABPG 1
:	:

If program 2 is started while program 1 is running, program 1 will be aborted.
If program 1 is started while program 2 is running, program 2 will be aborted.

Application

If a program number is specified in operand 2, the programs from the one corresponding to the program number in operand 1 to the other corresponding to the program number in operand 2 can be started (EXPG) or ended (ABPG) simultaneously.

[3] Note

- Up to 16 programs (maximum of 8 programs in the case of ASEL/PSEL/SSEL controllers) can be run at the same time. To use other programs when the controller is already running 16 programs, switch programs by closing a program or programs that are not required.
- If an ABPG command was executed to end a program while the program was executing a movement command, the actuator immediately decelerates to a stop.

3.5.26 Aborting a Program

[1] Description

Abort a program currently running.
Execute an ABPG command (command to abort other program) from other program in the multi-tasking mode.

[2] Example of Use

Main program (Prg. 1)

```
EXPG    n     The abort control program starts.
WTON   10
MOVP    1
BTON  303
       :
       :
```

Abort control program (Prg. n)

```
WTON   20    Wait for an abort input.
ABPG    1    Prg. 1 is aborted.
EXIT         The program ends.
```

[3] Note

If the running program was executing any movement command, the applicable axis immediately decelerates to a stop and then the program ends.

3.5.27 Way to Prevent Duplicated Startup by Program

[1] Description

How to prevent other programs from starting redundantly using virtual I/O port N710☐ (Program No.☐ running) is explained. If a given program is not running as determined by the checking of corresponding virtual I/O port N710☐ (Program No.☐ running), that program is started.

[2] Example of Use

```
   PRG1 (for task management)
   TAG1
N7102   EXPG      2   PRG2 stopped
                              PRG2 running
N7103   EXPG      3   PRG3 stopped
                              PRG3 running
N7104   EXPG      4   PRG4 stopped
                              PRG4 running
        TIMW   0.02   Task open
        GOTO      1
```

```
PRG2 (auto)
ABPG   3   4
   .
   .
```

```
PRG3 (step)
ABPG   2
ABPG   4
   .
   .
```

```
PRG4 (JOG)
ABPG   2   3
   .
   .
```

3.5.28 How to Cause Rotational Axis [Multi-rotation Specification] to Rotate Multiple Times

Regarding the axis operation types and rotation axis modes
(1) Axis-specific parameter No. 1, “Axis operation type”

No.	Parameter name	Default value	Input range	Unit
1	Axis operation type	Varies depending on the actuator.	0 to 1	None

● Explanation
This parameter defines the type of the actuator used. (Set this to 1.)
● Setting values
0: Linear movement axis   Actuator other than rotational axis
1: Rotational movement axis   Rotational axis (RS-30/60, RCS2-RT*/RTC*)

(2) Axis-specific parameter No. 66, “Rotational axis mode selection”

No.	Parameter name	Default value	Input range	Unit
66	Rotational axis mode selection	0	0 to 5	None

● Explanation
This parameter selects a desired rotational axis mode. (Set this to 1.)
Related parameter: Axis-specific No. 7, “Soft limit+”
● Setting values
0: Normal
1: Index mode
* When the index mode is selected, the soft limit is fixed to 359.999mm internally. Short-cut control is enabled while the index mode is selected.
2 to 5: Reserved by the system

⚠Caution: Absolute-specification actuators do not support the following settings:
- If this parameter is set to 0 (Linear movement axis), the infinite stroke mode cannot be set with parameter No. 68.
- If this parameter is set to 1 (Rotational movement axis), short-cut control cannot be selected in parameter No. 67.

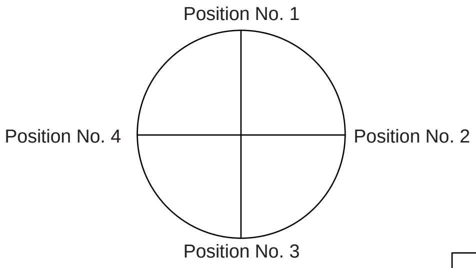
(3) Axis-specific parameter No. 67, “Short-cut control selection for rotational movement axis”

No.	Parameter name	Default value	Input range	Unit
67	Short-cut control selection for rotational movement axis	0	0 to 1	None

- Set this parameter to 1 when the rotation of the rotary axis is required to be in one way.
  Multi-rotation operation can be performed by setting this parameter to 1 (Short-cut control selected) and repeating a movement command in the same rotating direction.
  What is short-cut control?
  A type of operation in which the actuator moves to the closest point in the next move.
  It may shortcut if a far point is indicated. Pay attention to the indication value if the multi-rotation operation is preferred.

- Setting values
  0: Not selected
  1: Selected

Example: Move the actuator through position No. 2 → 3 → 4 → 1 successively based on position No. 1 being the reference point.



Program

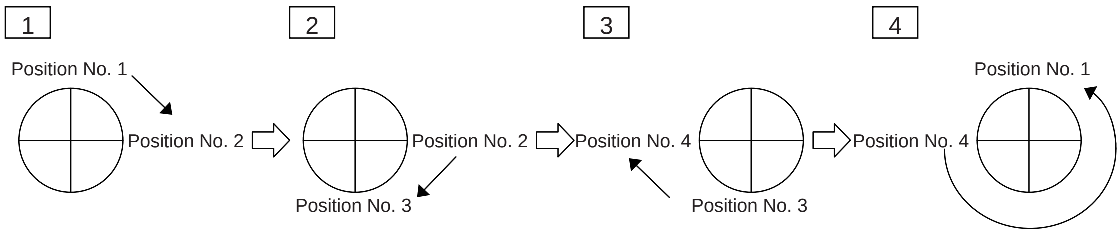
Step No.	Command	Operation
1	MOVP	2
2	MOVP	3
3	MOVP	4
4	MOVP	1

Position

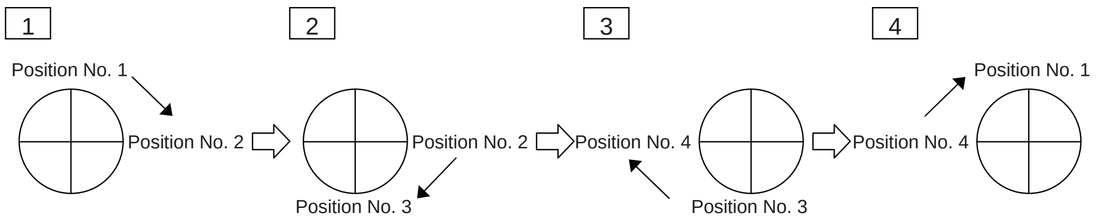
Position No.	Position data
1	0
2	90
3	100
4	270

1° of position data corresponds to 1mm.

Short-cut control not selected



Short-cut control selected



* By selecting short-cut control, you can cause the actuator to rotate in a specific direction.

3.6 For Advanced Operations (program edit)

3.6.1 Handling of Axis Numbers and Patterns

You can specify each axis using an axis number or multiple axes using an axis pattern.

[1] Axis number and indication of axis
Axes are indicated as follows so that multiple axes can be expressed.
To specify only one of multiple axes, specify it by the applicable axis number.

Single axis/rectangular axis

Axis	Axis number
Axis 1	1
Axis 2	2
Axis 3	3
Axis 4	4
Axis 5	5
Axis 6	6
Axis 7	7
Axis 8	8

SSEL, ASEL and PSEL controllers support only up to two axes. TT robots support only up to three axes.

In addition to following the above rule, you can also express axis numbers using symbols.

SCARA robot

Axis	Axis number
X-axis	1
Y-axis	2
Z-axis	3
R-axis	4

(Note) The movements of arms 1 and 2 of a SCARA robot are interlocked. It is not that arm 1 always represents the X-axis and arm 2, Y-axis.
Consider that the X-axis (axis No. 1) moves in the direction of X coordinates, while the Y-axis (axis No. 2) moves in the direction of Y coordinates.
Note that only when an AXST command is issued, the X-axis represents the arm 1 axis, while the Y-axis represents the arm 2 axis.

In addition to following the above rule, you can also express axis numbers using symbols.

[2] Axis pattern

[Single axis/rectangular axis]
Use "1" or "0" to indicate which axis(es) you want to use.

(Higher) (Lower)

Axis	Axis 8	Axis 7	Axis 6	Axis 5	Axis 4	Axis 3	Axis 2	Axis 1
Use	1	1	1	1	1	1	1	1
Do not use	0	0	0	0	0	0	0	0

[Example] Use axes 1 and 2.
Axis 2
↓
You should state 000011… (The 0's at the beginning are not necessary. Take them away and use 11.)
↑
Axis 1

[Example] Use axes 1 and 4.
Axis 4
↓
You should state 1001… (In this case, the 0's are needed to indicate the position of axis 4.)
↑
Axis 1

Specifying an axis pattern indirectly using a variable
Consider an axis pattern as a binary expression and assign a decimal equivalent of it to a variable.

[Example] Axis pattern where home return is performed only for axis 3

```
HOME    100
```

This pattern is specified indirectly as follows:
100 (binary) → 4 (decimal)
Accordingly:

```
LET     6       4
HOME    *6
```

If multiple axes must be specified at the same time, use an axis pattern.

- Commands where an axis pattern is used to specify axes
  OFST, GRP, SVON, SVOF, HOME, JFWN, JFWF, JBWN, JBWF, STOP, PTST, PRED CHVL, PBND, WZNA, WZNO, WZFA, WZFO, PAXS, NBND, PTRQ, MOVD, MVDI, NTCH, RAXS, XAXS, ECMD(250)

[SCARA robot]
Use “1” or “0” to indicate which axis(es) you want to use.

	(Higher)			(Lower)
Axis	R-axis	Z-axis	Y-axis	X-axis
Use	1	1	1	1
Do not use	0	0	0	0

(Note) The movements of arms 1 and 2 of a SCARA robot are interlocked. It is not that arm 1 always represents the X-axis and arm 2, Y-axis.
Consider that the X-axis (axis No. 1) moves in the direction of X coordinates, while the Y-axis (axis No. 2) moves in the direction of Y coordinates.

[Example] Use the X-axis and Y-axis.

Y-axis
↓
You should state 0011… (The 0’s at the beginning are not necessary. Take them away and use 11.)
↑
Axis 1

[Example] Use the X-axis and R-axis.

R-axis
↓
You should state 1001… (In this case, the 0’s are needed to indicate the position of axis R.)
↑
X-axis

Specifying an axis pattern indirectly using a variable
Consider an axis pattern as a binary expression and assign a decimal equivalent of it to a variable.

If multiple axes must be specified at the same time, use an axis pattern.

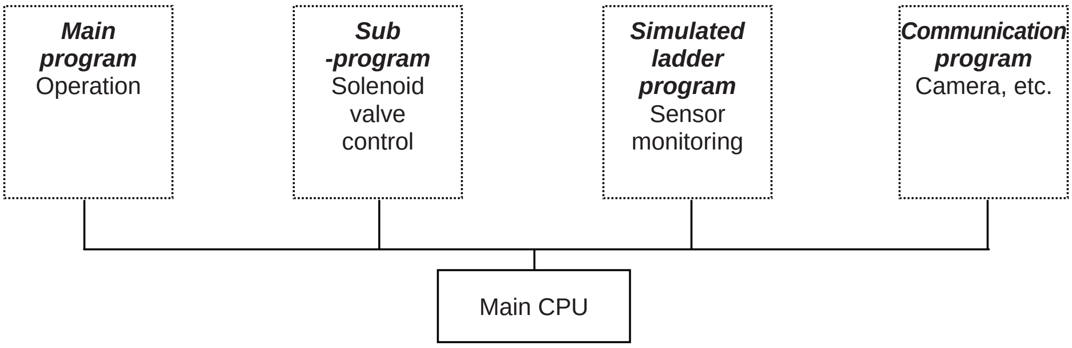
- Commands where an axis pattern is used to specify axes
  OFST, GRP, PTST, PRED, PBND
  (Note) In the case of SVON, SVOF and STOP, all axes are specified regardless of the axis pattern.

3.6.2 Setting of Multi-tasking and Task Level

[1] Multi-tasking

Controllers come standard with the multi-tasking function that allows multiple programs to be run simultaneously, such as moving actuators in one program and turning ON/OFF solenoid valves in another programs.
"Multi-tasking" literally means performing multiple tasks. The main CPU processes each program step by step. If multiple programs are run that contain actuator commands, timer commands, input waiting commands, etc., however, the main CPU uses an idle time while waiting for completion of each commanded task to process different programs.
(If a given program has no idle time, the system forcibly switches to the next program after 1mS based on "task slicing" action.)
XSEL controllers adopt high-speed CPUs, so multi-tasking is also performed at high speed.
Note that this function also supports simulated ladder circuits, which means that as long as your equipment is small enough you can build it as a sequencer.

[Example of multi-tasking (running multiple programs)]



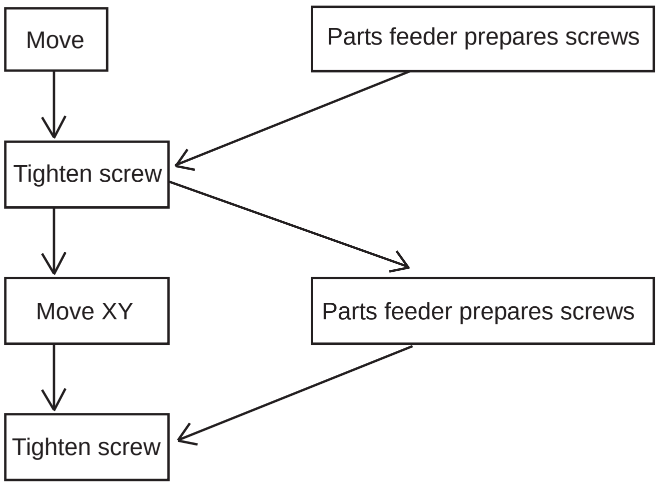
[2] Task level

If you want to perform a given task (program) preferentially over other tasks (programs), you can do so with a CHPR command by setting the parameter to "1: HIGH". If the parameter is set to "0: NORMAL", no priority is set.
You can also set task levels for simulated ladder programs. [Refer to Section 3.6.3]

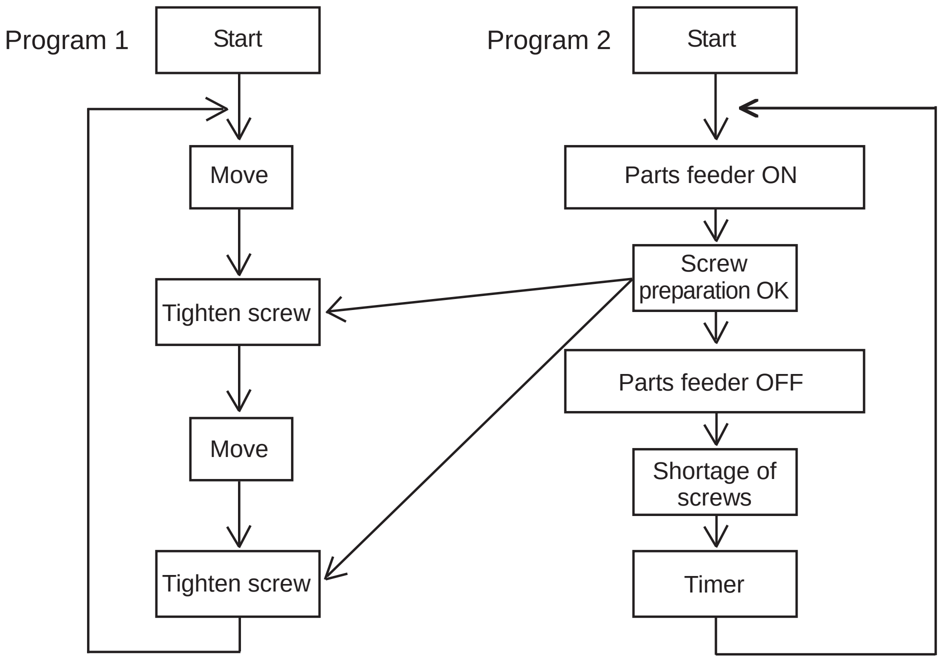
[3] Multi-Tasking

Take a screw-tightening robot, for example.
In general, a screw-tightening robot consists of axis 1 and axis 2 actuators and a screw-tightening machine (up/down air cylinder, etc.).

Operation Flow



Although the flow chart is simple, the movement of axis 1 and axis 2 actuators and the operation of the parts feeder must take place simultaneously. This requires “multi-tasking” operation.



[4] Program Switching

Various methods are available to switch between programs, depending on the purpose of programs. The representative methods are explained below.

```
                        ┌─ External start
Program switching ──────┤
                        └─ Program ──────┬─ Single-tasking -----------EXIT command
                                         └─ Multi-tasking -------------EXPG command
```

There are mainly two ways. One is to conduct with external startup and the other with application program.

(1) External start method ······· Refer to the Instruction Manual for each controller.

(2) Program method

○ Single-tasking

By executing EXIT Command (program finish) at the end of each program, finish the program and put back to the condition when the power is turned off. The home position, however, is remained, thus the next program can be executed with external start input by specifying another program number.

○ Multi-tasking

By creating a program for control and executing EXPG Command (startup of another program) in the program, multiple programs run in parallel one after another.

3.6.3 Pseudo-Ladder Task

A pseudo-ladder task function can be used depending on the command and extension condition.
The input format is shown below. Note that this function must be used by expert engineers with a full knowledge of PLC software design.

[1] Basic Frame

Extension condition	N	Input condition	Command	Operand 1	Operand 2	Output	
E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	
LD		7001	CHPR	1			
			TPCD	1			
			TAG	1			
⋮		⋮	⋮	⋮			Ladder statement field
LD		7001	TSLP	1 to 100			
⋮		⋮	⋮	⋮			Ladder statement field
LD		7001	TSLP	1 to 100			
LD		7001	GOTO	1			
LD		7001	EXIT				

* * Virtual input 7001: "Normally ON" contact

[2] Ladder Statement Field

1) Extension conditions

LD ·······LOAD
A ·······AND
O ·······OR
AB ·······AND BLOCK
OB ·······OR BLOCK
All of the above extension conditions can be used in non-ladder tasks.

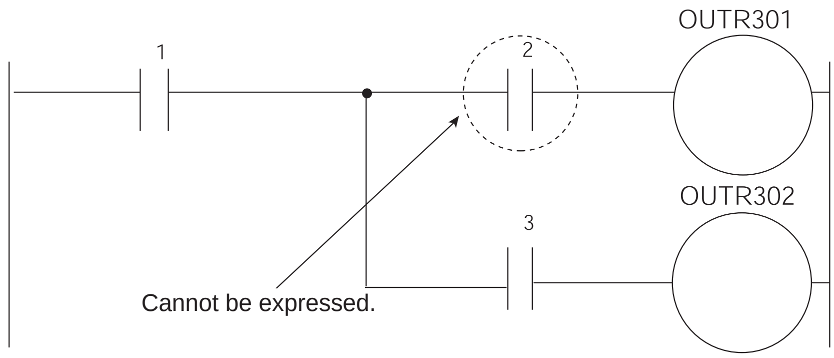
2) Ladder commands

OUTR ·········· Ladder output relay (Operand 1 = Output, flag number)
TIMR ·········· Ladder timer relay
(Operand 1 = Local flag number, Operand 2 = Timer setting (sec))

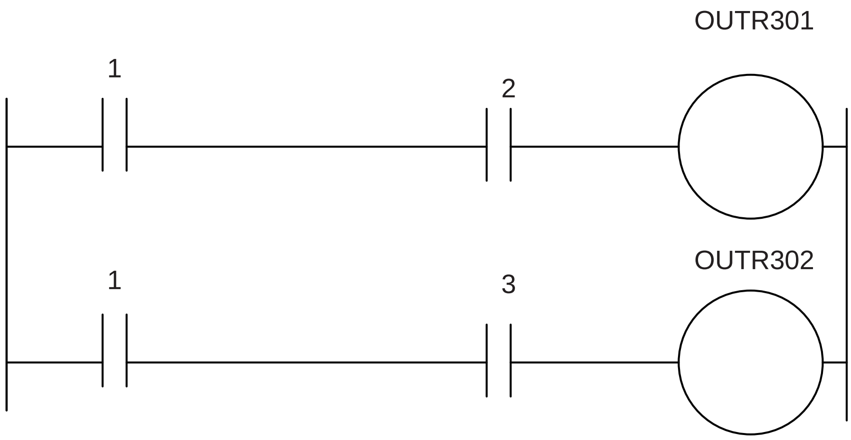
[3] Points to Note

- This system only processes software ladders using an interpreter. Therefore, the processing time is much longer than that of a dedicated commercial sequencer.
  (This system is not suitable for large-scale ladder processing.)
- If an extension condition is not specified for steps in which an input condition is specified, the steps will be treated as LD (LOAD).
- Always specify a “normally ON” contact for those steps that must be processed without fail, such as CHPR, TSLP and GOTO. (LD 7001)
  Virtual input 7001: “Normally ON” contact

- Ladder processing is based on software ladders using an interpreter, you cannot branch an output “1” to produce an input “2” or “3” as shown in the input circuit below.

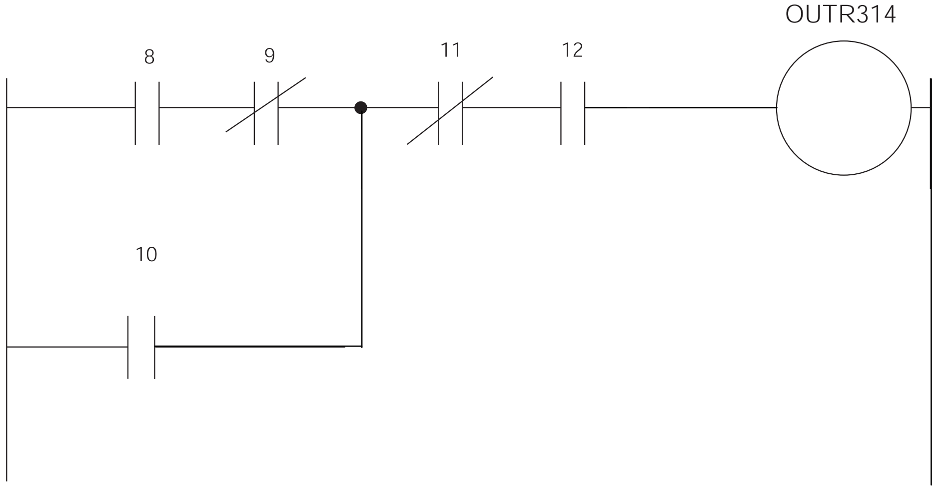


To perform this operation, you can write a ladder as follows, for example.
However, this is conditional upon the output “1” not changing during the output processing at OUTR301 in line 1. Make sure the output “1” does not change due to other programs.



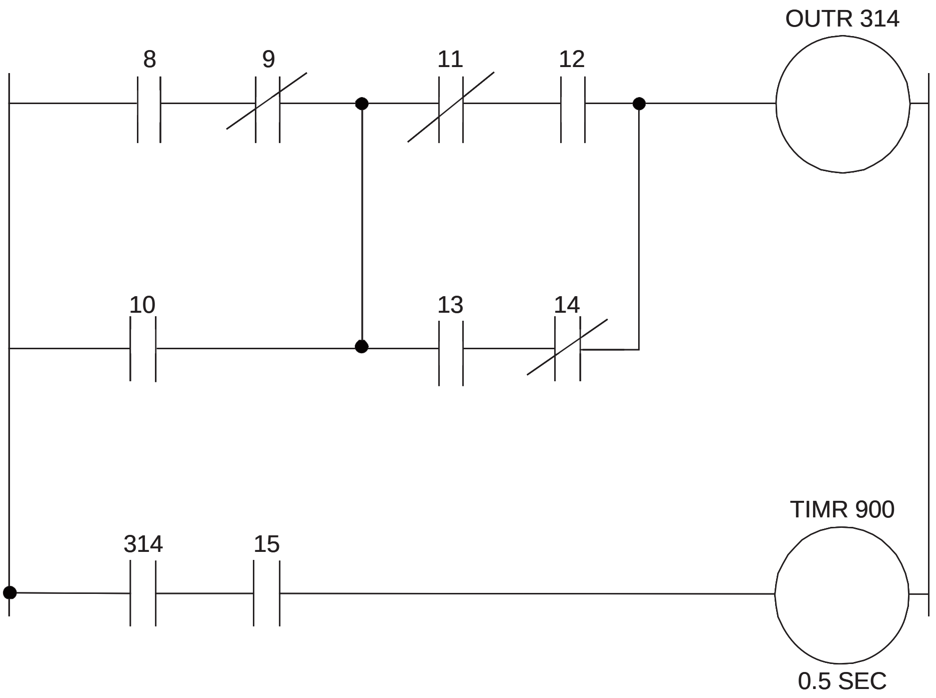
Extension condition	N	Input condition	Command	Operand 1	Operand 2	Operand 3
E	N	Cnd	Cmnd	Operand 1	Operand 2	Operand 3
LD		1				
A		2	OUTR	301		
LD		1				
A		3	OUTR	302		

[4] Program Example



Extension condition	N	Input condition	Command	Operand 1	Operand 2	Output
E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
LD		7001	CHPR	1		
			TPCD	1		
			TAG	1		
LD		8				
A	N	9				
O		10				
LD	N	11				
A		12				
AB			OUTR	314		
LD		7001	TSLP	3		
LD		7001	GOTO	1		
LD		7001	EXIT			

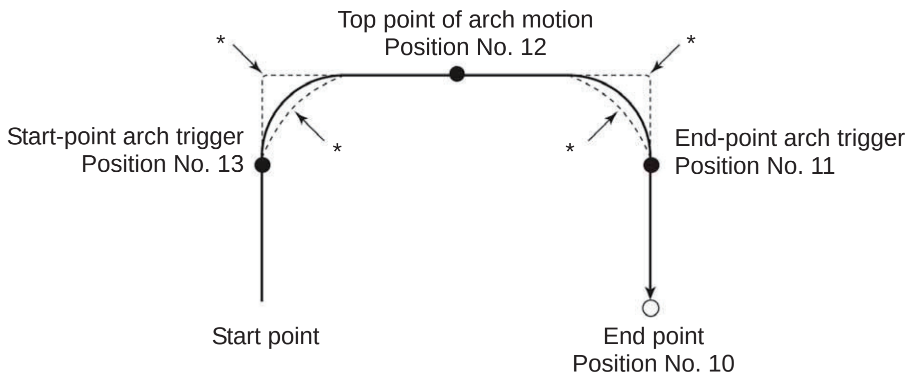
An example where 13, 14, 15 and timer TIMER900 are added further is given below.



Extension condition	N	Input condition	Command	Operand 1	Operand 2	Output
E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst
LD		7001	CHPR	1		
			TPCD	1		
			TAG	1		
LD		8				
A	N	9				
O		10				
LD	N	11				
A		12				
LD		13				
A	N	14				
OB						
AB			OUTR	314		
LD		314				
A		15	OUTR	900	0.5	
LD		7001	TSLP	3		
LD		7001	GOTO	1		
LD		7001	EXIT			

3.6.4 How to Use Arch Motion

Move from the current position to end point via arch motion.



Example of program

```
ACHZ   3                                                        -------- Declare arch motion Z-axis
ATRG  13 (start-point arch trigger position)  11 (end-point arch trigger position)  -------- Set arch trigger
                :
ARCH  10 (position of end point)          12 (position of top point of arch motion) ------  Arch motion
```

- Use an ACHZ command to specify the arch motion Z-axis. In the case of a SCARA robot, you only need to specify the Z-axis (axis 3) to perform arch motion. (ACHZ 3)
- Use an ATRG command to specify the arch motion trigger.
  After rising up to the start arch trigger from the current position by ARCH Command, a movement in the direction other than Z-axis starts with the arch motion.The actuator passes the top point being the Z point specified in operand 2 and completes the movement in a direction other than that of arch motion Z-axis, after which it passes near the end-point arch trigger and reaches the positions of the specified point.

Note

When operation is resumed after a pause, the transition from rising movement to horizontal movement and transition from horizontal movement to rising movement may follow the paths indicated by * (dotted lines) in the figure. Exercise caution to prevent contact.

- The arch motion Z-axis coordinate at the end point corresponds to the sum of the arch-motion Z-axis component of position data specified in operand 1, if any, and the arch-motion Z-axis offset. If the position data has no arch-motion Z component, the arch motion Z-axis coordinate corresponds to the sum of the arch motion Z-axis coordinate at the start point and the arch motion Z-axis offset. (Normally an offset is added to all positions such as the arch trigger and Z point.)
- If the start-point arch trigger is set below the start point or end-point arch trigger is set below the end point, an error occurs. (Note: The upward and downward directions have nothing to do with + and - of coordinates.)
- The rising direction of the arch motion Z-axis is the direction of moving from the end point to Z point (while the downward direction is the opposite of that direction), and has nothing to do with the magnitude correlation of coordinate values. Accordingly, be sure to check the actual operating direction when using this command.
- As for the data of end-position arch trigger, also start/end the operation at a point above the applicable arch trigger for any effective axis data other than data of the arch motion Z-axis, if available.
- If a composite arch trigger is set and any effective axis data is available other than data of the effective axis at the end point or arch motion Z-axis, the applicable axis also operates. In this case, also start/end the operation at a point above the applicable arch trigger.

3.6.5 How to Use Palletizing Function

The SEL language provides palletizing commands that support palletizing operation. These commands allow simple specification of various palletizing settings and enable arch motion ideal for palletizing. You can also call a subroutine at the palletizing destination to perform palletizing operation.

[1] How to Use

Use palletizing commands in the following steps:

(1) Palletizing setting
Set palletizing positions, arch motion, etc., using palletizing setting commands.

(2) Palletizing calculation
Specify palletizing positions using palletizing calculation commands.

(3) Palletizing movement
Execute motion using palletizing movement commands.

[2] Palletizing Setting

Use the palletizing setting commands to set items necessary for palletizing operation. The setting items include the following:

[Palletizing number setting .... Command: BGPA]

At the beginning of a palletizing setting, determine a palletizing number using a BGPA command to declare the start of palletizing setting.
At the end, declare the end of palletizing setting using an EDPA command.

BGPA	1	Declare the start of setting for palletizing No. 1.
		Set palletizing in these steps.
EDPA		Declare the end of palletizing setting at the end.

A maximum of 10 sets (palletizing No. 1 to 10) of palletizing setting can be specified for each program.

[Palletizing pattern .... Command: PAPN]
Select a pattern indicating the palletizing order.
The two patterns illustrated below are available.
The encircled numbers indicate the order of palletizing and are called “palletizing position numbers”.

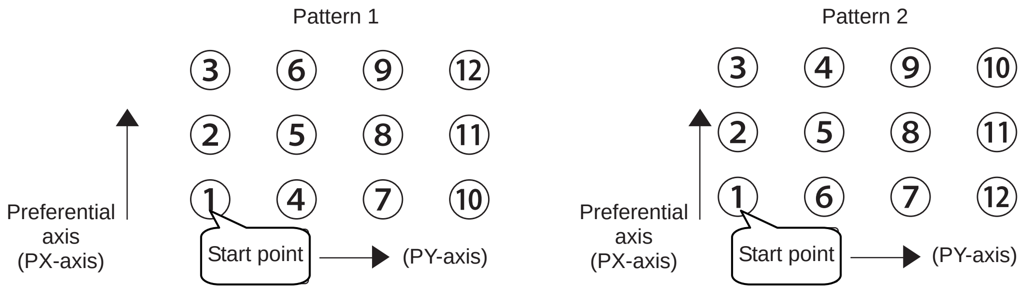


Fig. 1

PAPN	2	When pattern 2 is selected (Setting is not necessary if pattern 1 is selected.)

The row from 1 to 3 to be placed first is called the “preferential axis (PX-axis)”, while the other direction comprising the palletizing plane is called the “PY-axis”.

[Palletizing counts .... Command: PAPI]
Set the palletizing counts.

PAPI	3	4	Count for preferential axis (PX-axis): 3, Count for PY-axis: 4

[Palletizing position setting]
Palletizing position setting is performed mainly by method A or B, as explained below. Set the palletizing positions for each palletizing setting based on method A or B.

	Setting method	Commands
A	[3-point or 4-points teaching method] Set three position-data points or four position-data points specifying the palletizing positions.	PAPS
B	[Method to set palletizing positions in parallel with the actuators (in parallel with an axis on the work coordinate system in the case of a SCARA robot)] Set from the palletizing axes, palletizing reference point and palletizing pitches.	PASE, PAST PAPT

A. 3-point teaching method

To set the palletizing positions by 3-point teaching, store desired positions in position data fields as three continuous position data and then specify the first position number using a PAPS command.
This method allows you to set the PX-axis and PY-axis as three-dimensional axes not parallel with the actuators and not crossing with each other.
In the example shown below, position data [1], [3] and [10] are stored in three continuous position data fields.

When three points are taught from position No. 11
- Position No. 11 [1] : Start point (First palletizing position)
- Position No. 12 [3] : Palletizing position corresponding to the end point in the PX-axis direction
- Position No. 13 [10] : Palletizing position corresponding to the end point in the PY-axis direction

The encircled numbers indicate palletizing position numbers (palletizing order).

Use a PAPS command to specify the position number corresponding to the start point.

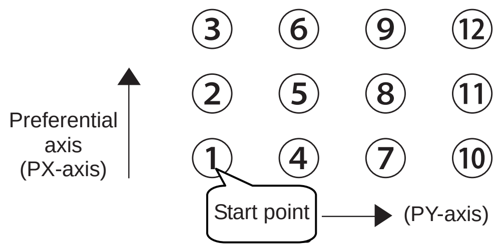


Fig. 1

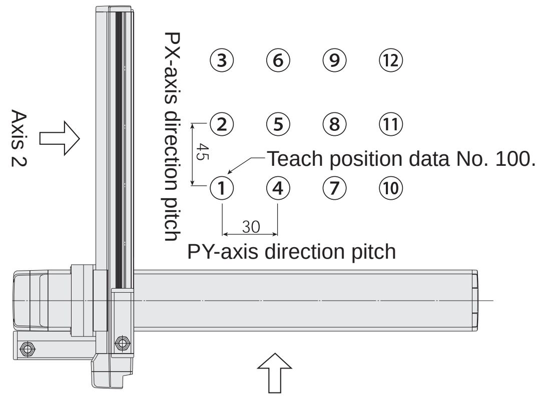
```
PAPS        11
```

The pitches are calculated automatically from the count set for each axis.
In 3-point teaching, you can specify position data for two axes or three axes. If position data is specified for three axes, the palletizing plane becomes a three-dimensional plane.

B. Method to set palletizing positions in parallel with the actuators

Palletizing reference point:	Store the position data of the start point (palletizing position No. 1) in a position data field and specify the applicable position number using a PAST command, as shown below.
Palletizing pitches:	Use a PAPT command to specify the pitches in the PX-axis and PY-axis directions.
Palletizing axes:	Use a PASE command to specify the two axes, one representing the PX-axis direction and the other representing the PY-axis direction, to be used in palletizing.

(An actuator axis number parallel with the preferential axis (PX-axis) and another perpendicular to the preferential axis)



PAST	100		Teach position data No. 100 as the start point.
PAPT	45	30	The PX-axis direction pitch is 45mm and the PY-axis direction pitch is 30mm.
PASE	2	1	Set axis 2 as the priority axis (PX-axis) and axis 1 as the rectangular axis and the priority axis.

(Note) When the above palletizing axes, palletizing pitches and palletizing reference point are used, the PX-axis and PY-axis must be parallel with the actuators and crossing with each other.

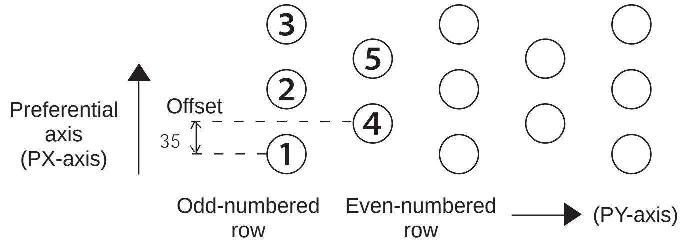
Select either method A or B for each palletizing setting.

[Zigzag setting .... Command: PSLI]

Use a PSLI command to set a zigzag layout as shown below.

Zigzag offset: Offset amount in the preferential-axis direction, which will be applied when even-numbered rows are placed.
"Even-numbered rows" refer to the rows occurring at the even numbers based on the row placed first representing the first row.

Zigzag count: Number in the even-numbered rows. Two in the diagram below.



PSLI 35 2

[Arch motion setting]

(a) Arch motion Z-axis No. .... Applicable command: ACHZ

- In the case of a SCARA robot, you only need to specify the Z-axis (axis 3) to perform arch motion.

  ACHZ 3

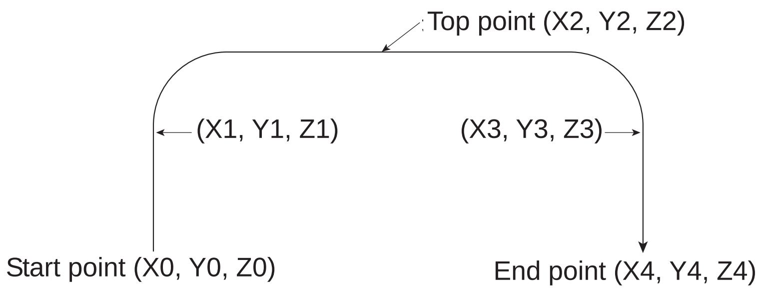
(b) Arch motion Z-axis offset .... Applicable command: OFAZ

(c) Composite arch motion .... Applicable command: AEXT
Composite arch motion data refers to position data used when you want to cause any axis other than the effective axis at the end point or arch motion Z-axis to perform an additional operation (such as when setting a rotational angle).
Note, however, that any composite axis operation starts and ends at a position above the applicable arch trigger.
Set this composite arch motion setting command by specifying a position number under which composite arch motion data is set.

(d) Arch trigger .... Applicable command: ATRG
The following arch trigger settings are available for arch motion.
Set this arch trigger setting command by specifying a position number under which arch trigger coordinate data is stored.

(d-1) Start-point arch trigger
Specify the position to be reached after the arch motion is started from the start point and the actuator moves in the arch motion Z-axis coordinate direction, after which the actuator will start moving in the direction of other axis. Start-point arch trigger = Z1

(d-2) End-point arch trigger
Specify the position to be reached in the arch-motion Z-axis coordinate direction during the arch motion down movement, after which the actuator will end moving in the direction of other axis. End-point arch trigger = Z3



[Palletizing arch motion setting]

(a) Axis number corresponding to palletizing Z direction...... Applicable command: PCHZ

(b) Palletizing Z-axis offset .... Applicable command: OFPZ

(c) Composite palletizing .... Applicable command: PEXT

Composite palletizing data refers to position data used when you want to cause any axis other than the PX- or PY- (or PZ-) axis to perform an additional operation in a palletizing movement command (such as when setting a rotational angle).

Note, however, that any composite axis operation starts and ends at a position above the applicable palletizing arch trigger.

Set this composite palletizing setting command by specifying a position number under which composite palletizing data is stored.

(d) Palletizing arch trigger .... Applicable command: PTRG

If the end point is the palletizing point, you need palletizing arch triggers just like arch triggers.

Set this palletizing arch trigger setting command by setting a position number under which palletizing arch trigger coordinate data is stored.

(d-1) Palletizing start-point arch trigger

(d-2) Palletizing end-point arch trigger

[3] Palletizing Calculation

The items that can be operated or obtained using palletizing calculation commands are shown below:

[Palletizing position number Commands ...... PSET, PINC, PDEC, PTNG]

Number showing the ordinal number of a palletizing point. (In Fig. 1 for [2] given in the explanation of palletizing pattern, the encircled numbers are palletizing position numbers.)

Always set this command before executing a palletizing movement command --- PSET

For example, executing a palletizing movement command by setting 1 as the palletizing position number will move the axes to the start point. Executing a palletizing movement command by setting 2 as the palletizing position number will move the axes to the point immediately next to the start point in the PX-axis direction.

[Palletizing angle Command .... PARG]

This is the angle formed by the physical axis and the preferential palletizing axis (PX-axis) (θ in the figure below).

θ represents an angle calculated by ignoring the coordinate in palletizing Z-axis direction.

In the figure below, θ will become a negative value if axis 1 is used as the reference for angle calculation.

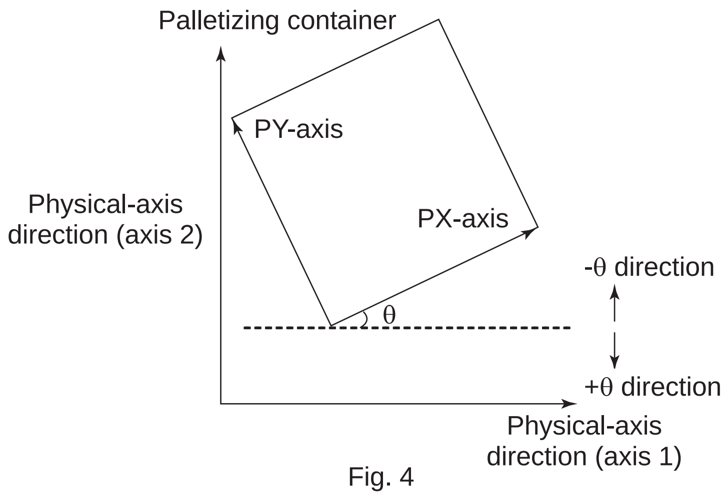


Fig. 4

If the composite axis is a rotating axis, you can obtain the palletizing angle and add it as an offset to the operation of the composite axis in order to correct the composite axis against any position error of the palletizing container.

With XSEL commands, executing a “get palletizing angle” command following a palletizing setting via 3-point teaching will automatically obtain the palletizing angle.

If 3-point teaching is set three-dimensionally, you must specify the palletizing Z-axis.

[Palletizing calculation data Command ..... PAPG]

When a palletizing position number is set, this data refers to the position coordinate data of the palletizing point corresponding to that palletizing position number.

Note, however, that this position coordinate data does not reflect any normal offset or palletizing Z-axis offset.

[4] Palletizing Movement

Palletizing movement commands include commands used to move the actuator to palletizing points and other that uses position data to specify the end point.

[Movement commands to palletizing point .... PMVP, PMVL, PACH]

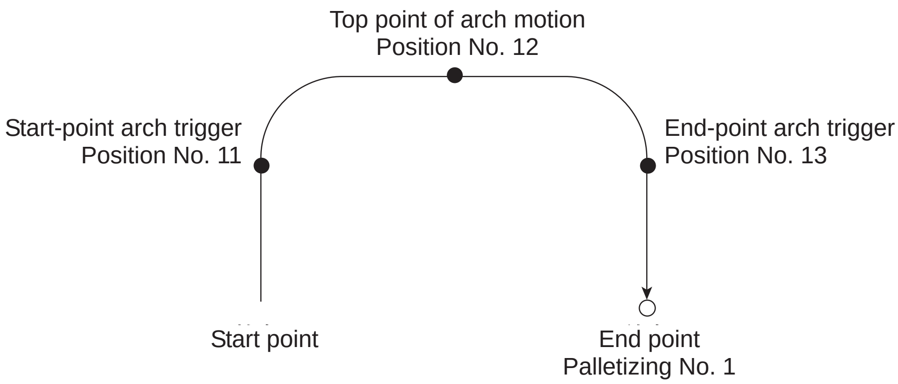
Calculate the position coordinate of a two-dimensionally or three-dimensionally positioned palletizing point and use this coordinate as the end point to move the actuator. (The actuator moves to the palletizing point corresponding to the palletizing position number specified in the command when executed)

You need two actuator axes to constitute a two-dimensional plane. If you need a vertical axis (PZ-axis), you must specify one more axis.

PMVP: Move from the current position to a palletizing point via PTP.

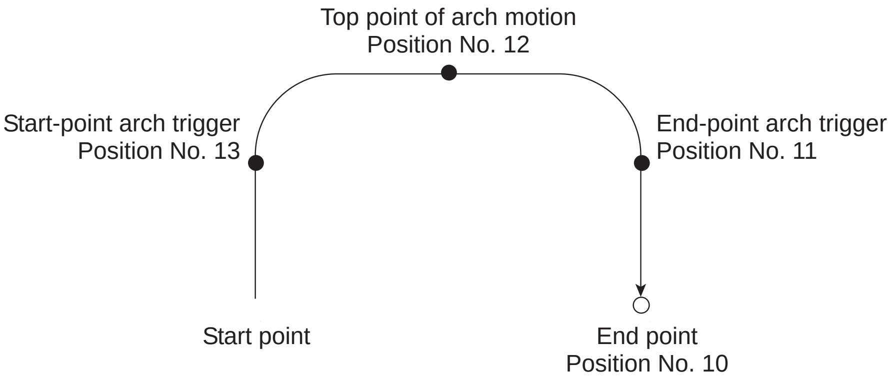
PMVL: Move from the current position to a palletizing point via interpolation.

PACH: Move from the current position to palletizing position via arch motion.
You must set palletizing arch motion based on palletizing setting.




```
PCHZ    3
PTRG    11    13
  :
  :
  :
PACH    1     12
```

[Movement command that uses position data as end point...ARCH]
Arch motion is performed to the end point specified by position data.
If the movement is linear in parallel with the actuator, arch motion operation can be possible by specifying only two axes including the applicable axis and PZ-axis.
Arch motion must be set.



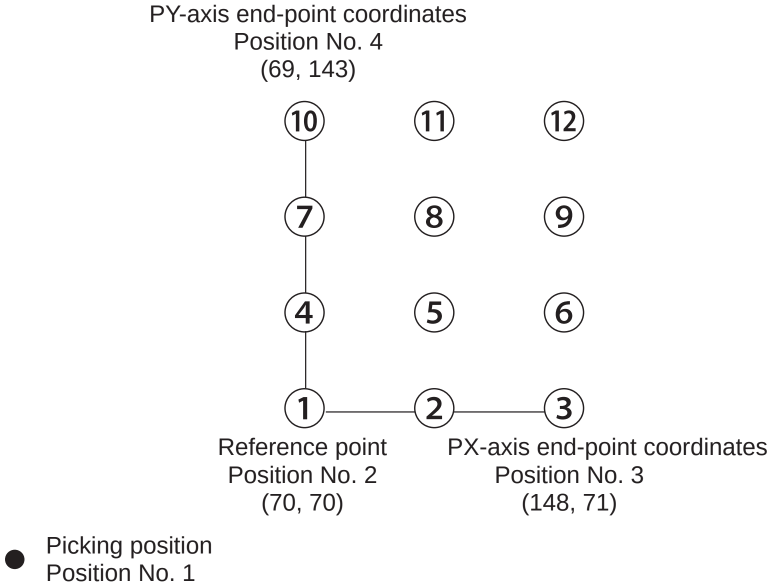

```
ACHZ    3
ATRG    13    11
  ·
  ·
  ·
ARCH    10    12
```

[5] Program Examples

[Simple program example (two-axis specification) using PAPS (set by 3-point teaching)]
The example below specifies movement only and does not cover picking operation.

Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
1				BGPA	1			Start setting palletizing No. 1
2				PAPI	3	4		Number of palletizing points 3 × 4
3				PAPS	2			Set 3-point teaching
4				EDPA				End setting palletizing No. 1
5								
6				VEL	200			Speed 200mm/sec
7				MOVL	1			Move to pick position
8				PSET	1	1		Set palletizing position number to 1
9				TAG	1			
10				PMVL	1			Move to palletizing position via interpolation
11				MOVL	1			Move to pick position via interpolation
12				PINC	1		600	Increment palletizing position number by 1
13			600	GOTO	1			Move to beginning of loop if PINC was successful
14				EXIT				End

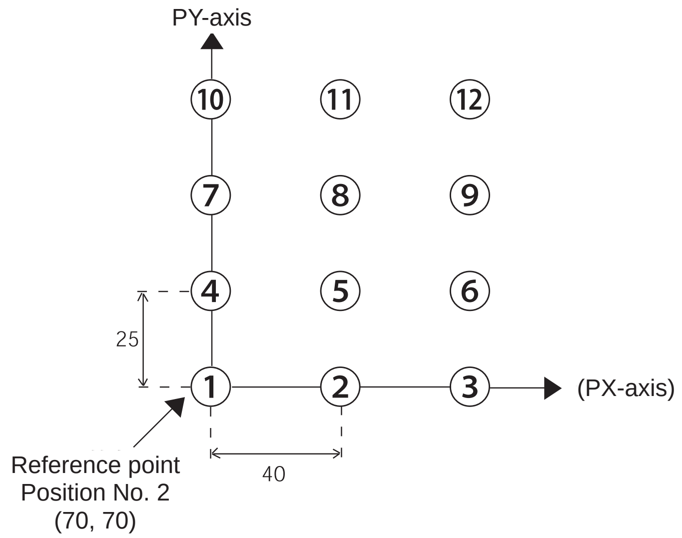
No.	Axis 1	Axis 2	Vel	Acc	Dcl	Remarks
1	10.000	10.000				Pick position
2	70.000	70.000				Position data of reference point
3	148.000	71.000				Position data PX-axis end point
4	69.000	143.000				Position data PY-axis end point



[Simple program example (two-axis specification) using PAPS, PAPT and PAST]
The example below specifies movement only and does not cover picking operation.

Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
1				BGPA	1			Start setting palletizing No. 1
2				PAPI	3	4		Number of palletizing points 3 × 4
3				PASE	1	2		PX-axis = Axis 1, PY-axis = Axis 2
4				PAPS	40	25		Pitch X = 40, Y = 25
5				PAST	2			Position No. 2 = Reference point
6				EDPA				End setting palletizing No. 1
7								
8				VEL	200			Speed 200mm/sec
9				MOVL	1			Move to pick position
10				PSET	1	1		Set palletizing position number to 1
11				TAG	1			
12				PMVL	1			Move to palletizing position via interpolation
13				MOVL	1			Move to pick position via interpolation
14				PINC	1		600	Increment palletizing position number by 1
15			600	GOTO	1			Move to beginning of loop if PINC was successful
16				EXIT				End

No.	Axis 1	Axis 2	Vel	Acc	Dcl	Remarks
1	10.000	10.000				Pick position
2	70.000	70.000				Position data of reference point



Pitch in PX-axis direction: 40
Pitch in PY-axis direction: 25
The PX-axis is parallel with axis 1,
while the PY-axis is parallel with axis 2.

● Picking position
Position No. 1

[Program example using PAPS (set by 3-point teaching)]
The example below specifies movement only and does not cover picking operation.

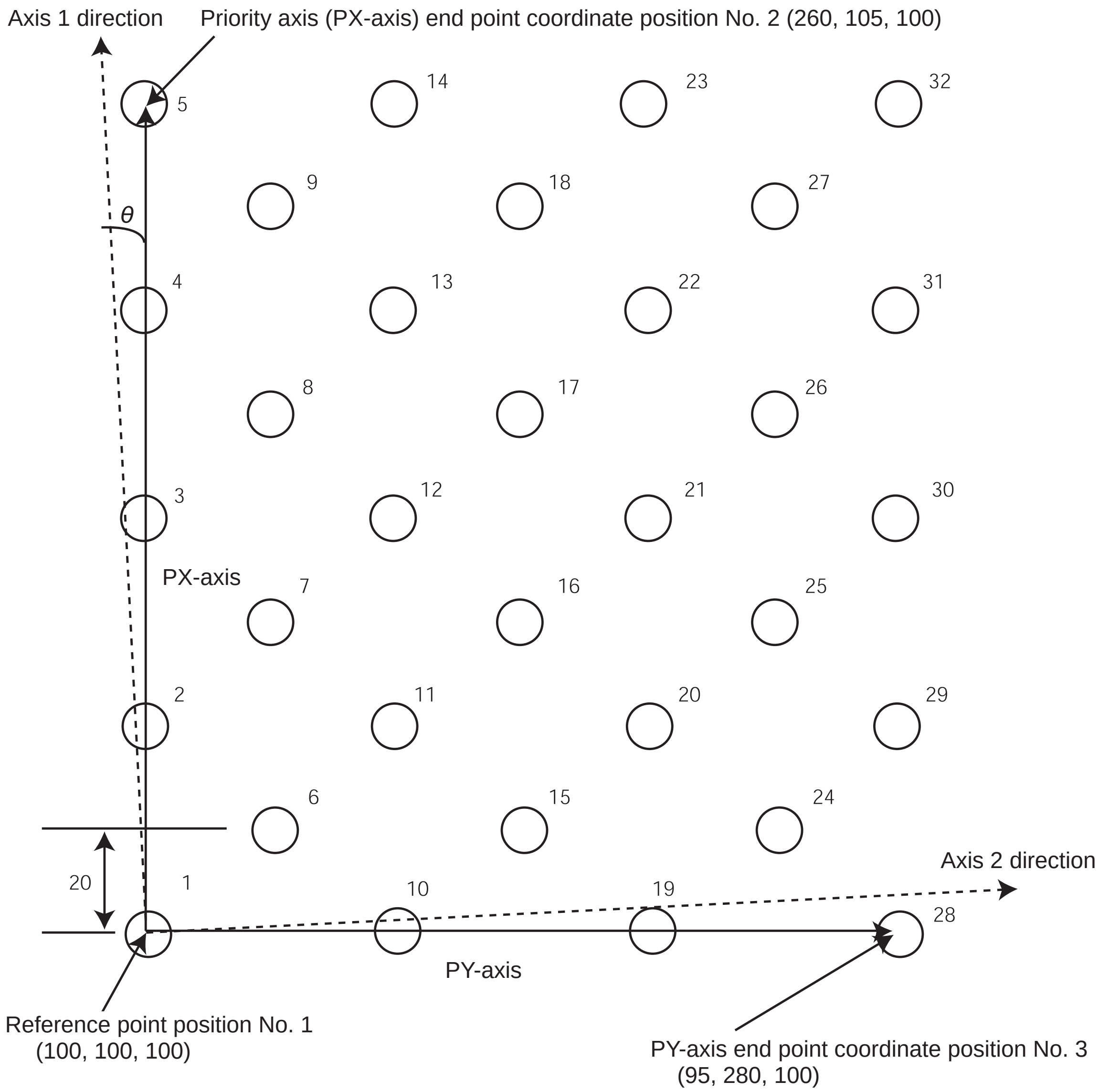
Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
1				BGPA	1			Start setting palletizing No. 1
2								
3				PAPI	5	7		Number of palletizing points 5 × 7
4				PAPN	1			Palletizing pattern 1
5				PAPS	1			Set by 3-point teaching
6								Use data of position No. 1
7				PSLI	20	4		Zigzag offset = 20mm
8				PCHZ	3			Palletizing Z-axis = Axis 3
9				PTRG	4	4		Set palletizing arch trigger
10								Use data of position No. 4
11				OFPZ	100			PZ-axis offset = 100mm
12				PEXT	6			Set composite palletizing
13								Use data of position No. 6
14				EDPA				
15								
16				PARG	1	1		Get palletizing angle
17								The data is stored in variable 199.
18				PPUT	4	6		Store angle data in variable 199 under axis 4 in position No. 6
19								
20	*///							
21								
22				ATRG	4	4		Set arch trigger
23								Use data of position No. 4
24				ACHZ	3			Set arch motion Z-axis
25								
26				ACC	0.3			Acceleration
27				DCL	0.3			Deceleration
28				VLMX				
29								
30				PSET	1	1		Set palletizing position number to 1

Continues to the next page

Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
31				MOVP	8			Move to pick position
32								
33				TAG	1			Beginning of loop process
34				PACH	1	9		Palletizing arch motion
35								Z point specified under position No. 9
36				ARCH	8	9		Arch motion
37								Z point specified under position No. 9
38				PINC	1		600	Increment palletizing position number by 1
39			600	GOTO	1			Move to beginning of loop if PINC was successful
40								
41				EXIT				End task
42								
43								
44								
45								

No.	Axis 1	Axis 2	Axis 3	Axis 4	Remarks
1	100.000	100.000	100.000	*.***	Position data of reference point
2	260.000	105.000	100.000	*.***	Position data PX-axis end point
3	95.000	280.000	100.000	*.***	Position data PY-axis end point
4	*.***	*.***	50.000	*.***	Position data for arch trigger
5	*.***	*.***	*.***	*.***	(Not used)
6	*.***	*.***	*.***	–1.79	Position data for composite palletizing
7	*.***	*.***	*.***	*.***	(Not used)
8	0.000	0.000	100.000	0.000	Position data of pick position
9	*.***	*.***	0.000	*.***	Z position data
10					

Schematic diagram of placement point positions according to the program defined earlier



- The number at the top right of each circle indicates the palletizing position number.
- Number of points in PX-axis direction = 5, Number of points PY-axis direction = 7
- Zigzag offsets: 20
- Number of zigzags: 4
- Pallet angle error θ: -1.79°

[Example of program using PASE, PAPT and PAST]
The following program consists of movements only and does not support pick operation.

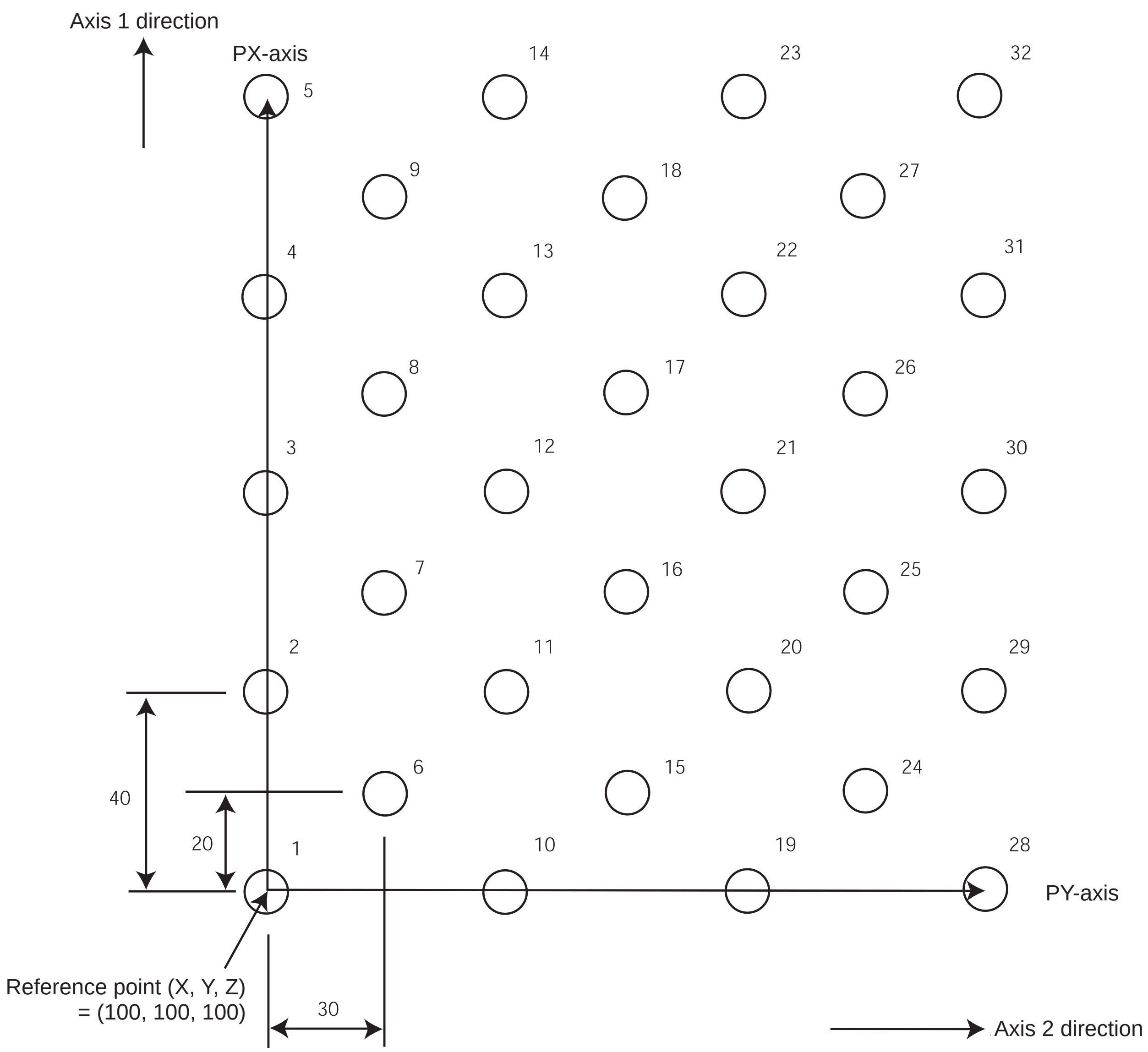
Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
1				BGPA	1			Start setting palletizing No. 1
2								
3				PAPI	5	7		Number of palletizing points 5 × 7
4				PAPN	1			Palletizing pattern 1
5				PASE	1	2		PX-axis = Axis 1, PY-axis = Axis 2
6				PAPT	40	30		Pitch (X = 40, Y = 30mm)
7				PAST	1			Set reference position data
8								Use data of position No. 1
9				PSLI	20	4		Zigzag offset = 20mm
10								Number of zigzags = 4
11				PCHZ	3			Palletizing Z-axis = Axis 3
12				PTRG	4	4		Set palletizing arch trigger
13								Use data of position No. 4
14				OFPZ	100			PZ-axis offset = 100mm
15								
16				EDPA				
17								
18	*///							
19				ATRG	4	4		Set arch trigger
20								Use data of position No. 4
21				ACHZ	3			Set arch motion Z-axis
22								
23				ACC	0.3			Acceleration
24				DCL	0.3			Deceleration
25				VLMX				
26								
27				PSET	1	1		Set palletizing position number
28				MOVP	8			Move to pick position
29	*///							
30								

Continues to the next page

Step	E	N	Cnd	Cmnd	Operand1	Operand2	Pst	Comment
31				TAG	1			Beginning of loop process
32				PACH	1	9		Palletizing arch motion
33								Z point specified under position No. 9
34				ARCH	8	9		Arch motion
35								Z point specified under position No. 9
36				PINC	1		600	Increment palletizing position number by 1
37			600	GOTO	1			Move to beginning of loop if PINC was successful
38								
39				EXIT				End task
40								

No.	Axis 1	Axis 2	Axis 3	Axis 4	Remarks
1	100.000	100.000	100.000	*.***	Position data of reference point
2	*.***	*.***	*.***	*.***	(Not used)
3	*.***	*.***	*.***	*.***	(Not used)
4	*.***	*.***	50.000	*.***	Position data for arch trigger
5	*.***	*.***	*.***	*.***	(Not used)
6	*.***	*.***	*.***	*.***	(Not used)
7	*.***	*.***	*.***	*.***	(Not used)
8	0.000	0.000	100.000	0.000	Position data of pick position
9	*.***	*.***	0.000	*.***	Z position data
10					

Schematic diagram of placement point positions according to the program defined earlier



- The number at the top right of each circle indicates the palletizing position number.
- Number of points in PX-axis direction = 5, Number of points PY-axis direction = 7
- Pitch in PX-axis direction: 40
- Pitch in PY-axis direction: 30
- Zigzag offsets: 20
- Number of zigzags: 4

3.6.6 Handling of WAIT Timers

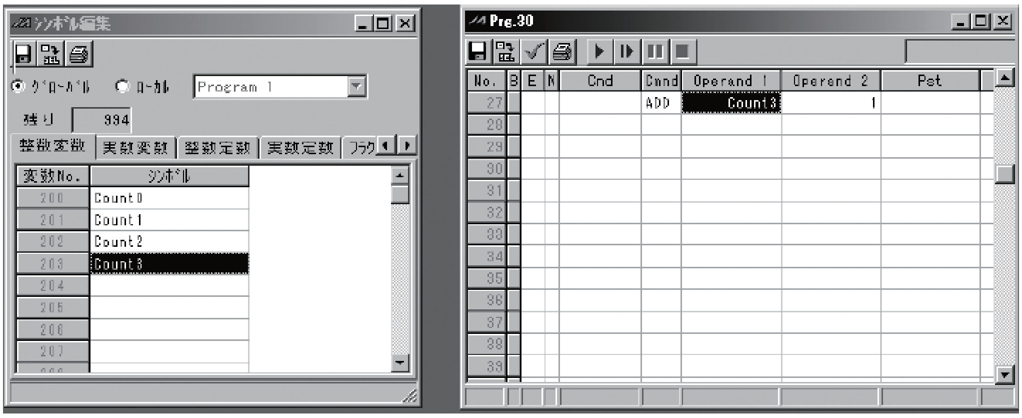
WAIT timers are provided to wait for certain events to occur.
Use a TIMW command to specify waiting.
WAIT timers can be actuated in each program.

3.6.7 Handling of Shot Pulse Timers

Shot pulse timers provide a function to turn ON/OFF an I/O flag for a specified time.
You can use a BTPN command to turn ON an I/O or flag for a specified time.
Similarly, you can use a BTPF command to turn OFF an I/O or flag for a specified time.
The maximum number of shot pulse timers that can be actuated simultaneously in one program is 16 as a total of BTPN and BTPF commands.
Note, however, that there are no limitations to how many times these timers can be used in one program.

3.6.8 Handling of Number of Symbol Definitions

With XSEL controllers, you can create a program with ease by using symbols representing variable numbers, flag numbers, etc.
In the example below, variable No. 203 is defined the symbol “Count3” in the symbol edit screen.
The defined symbol can be used in programs, and each statement of “Count3” in programs indicates variable No. 203.



Symbol edit screen   Program edit screen

For information on how to edit symbols, refer to “Editing Symbols” in the XSEL Teaching Pendant Instruction Manual or “Symbol Edit Window” in the XSEL PC Software Instruction Manual.

[1] Scope of support
The following items support use of symbols:
Variable number, flag number, tag number, subroutine number, program number, position number, input port number, output port number, axis number, constant

[2] Convention of symbol statement
1) Up to nine single-byte alphanumeric or underscore characters, starting from an alphabet. (Note: Up to eight single-byte characters in the case of character string literals)
   * Symbols can also start with an underscore if you are using PC Software Ver.1.1.0.5 or later and Teaching Pendant Ver.1.04 or later.
   * Among the ASCII codes 21h to 7Eh, those single-byte characters that can be entered from the keyboard can be entered as the second and subsequent characters in a symbol, if you are using PC Software Ver.1.1.0.5 or later.
   * Note that same ASCII codes may be expressed differently if the font used on the PC is different from that used on the teaching pendant (the same also applies to character string literals).
     5Ch ····· PC software: Backslash ＼ (overseas specification, etc.)
     Teaching pendant: Yen symbol ¥
     7Eh ····· PC software: ~
     Teaching pendant: Right arrow →
2) Defining symbols of the same name within the same function is prohibited. (Defining local symbols of the same name in different programs is permitted.)
3) Defining symbols of the same name within the flag number group, input port number group or output port number group is prohibited.
   (Defining local symbols of the same name in different programs is permitted.)
4) Defining symbols of the same name within the integer variable number group or real variable number group is prohibited.
   (Defining local symbols of the same name in different programs is permitted.)
5) Defining symbols of the same name within the integer constant group or real constant group is prohibited.

3.6.9 RS232C Communication

[1] String processing commands

Strings are character strings. Strings used by the controllers covered by this manual include global strings and local strings.
Global strings can be read or written commonly from any program.
Local strings are valid only within each program and cannot be used in other programs.
Global strings and local strings are differentiated by the range to which their number belongs.

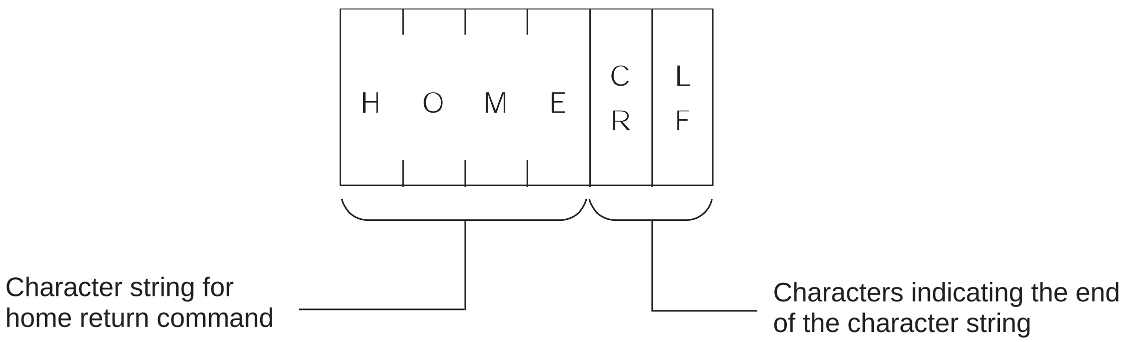
Global areas	300 to 999 (700)
Local areas	1 to 299 (299)

The communication with the external devices requires to be conducted with the serial communication using character lines, thus a use of the string is required.

[2] Explanation of transmission format

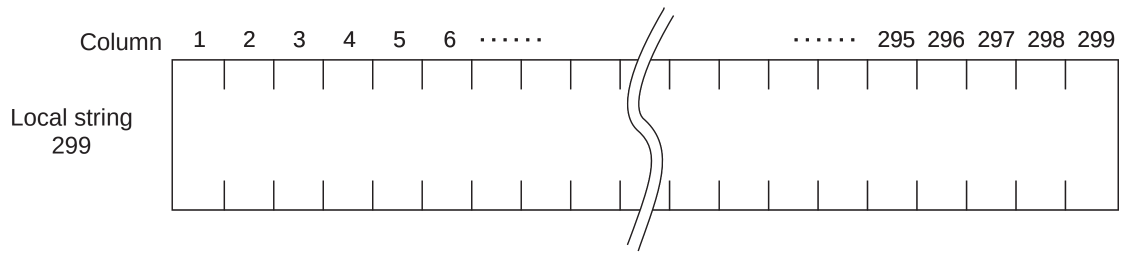
Communication performed by the systems covered by this manual is basically exchange of character strings.
Which character strings should be used for which operations is determined beforehand, so that the receiving side can recognize each character string and perform the corresponding operation.
A combination of these strings and characters indicating the end of one character string is called “transmission format”, and the user can determine a desired transmission format freely.
For example, assume a character string consisting of four characters “HOME” which is used as a home return command.
It is determined the character to finish the character line should be either “CR” or “LF” on PC. Therefore, it is necessary to follow this rule.

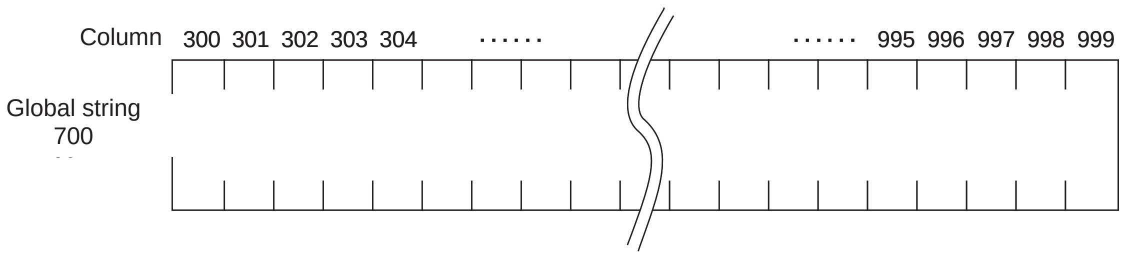
Example of transmission format



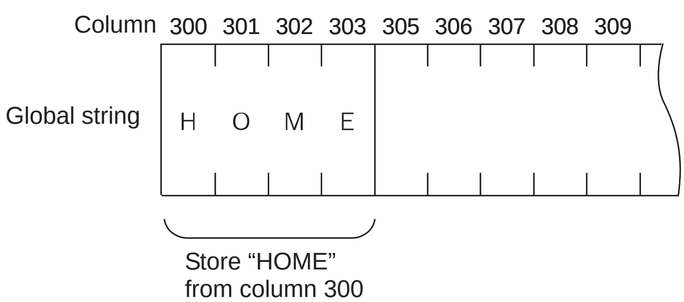
[3] Explanation of string

Strings sent according to the format explained above are stored in boxes designed to contain character strings, so that they can be used freely in the program.
Two types of strings are available: global strings that can be read or written in all programs, and local strings that can be red or written only in each program. Both strings are differentiated by their column numbers.





One character is stored in each of the fields of these strings.
The position of a given field comprising a string is indicated by “column XX” and which column to store can be set freely for each command.
For instance, if a character line “HOME”, which indicates the home-return command, is received, and the character line is desired to be used in several programs, you should save the data to Column 300 in the global string.



[4] Determination of transmission format

In this example of application program, three types of transmission formats are required, or namely transmission formats for home return command, movement command and movement completion. These formats are determined as follows. Note that these are only examples and the user can freely determine each format.

A. Home return command format

This format is used to issue a home return command from the PC to the controller.

H	O	M	E	CR	LF

B. Movement command format

This format is used to issue an axis movement command from the PC to the controller.

				Speed			Axis 1 position							Axis 2 position								
M	O	V	E	9	9	9	9	9	9	.	9	9	9	9	9	9	.	9	9	9	CR	LF

C. Movement completion format

This format is sent from the controller to the PC when the home return or movement is completed.

O	K	CR	LF

[5] Processing procedure

The processing procedure you should follow to program this application example is explained.

A. Set “LF” as characters (terminator characters) indicating the end of a string.
B. Open channel 1 of the RS232 unit to use this channel 1.
C. Program so that any data, if sent through channel 1, is received into columns starting from column 1 for local strings.
D. Program so that if the received data is “MOVE”, the applicable speed data is converted to a binary value and the converted binary value is set in variable 10, while the applicable position data is converted to a binary value and the obtained binary value is set in position No. 1, after which the actuator moves and when the movement is completed, “OK” is sent.

[6] Application program

STEP	No.	N	OP-CODE	OPRND1	OPRND2	POST	Comment
1			SCHA	10			Set LF as terminator characters
2			OPEN	1			Open SIO channel 1
3			TAG	1			
4			READ	1	1		Read into columns starting from SIO 1 column 1
5							
6			ISEQ	1	‘HOME’		If Home return command
7			HOME	11			Home return
8			EXSR	1			Send OK
9			EDIF				
10							
11			ISEQ	1	‘MOVE’		If movement command:
12			SLEN	3			Reading period with three digits
13			VAL	10	5		Set speed in variable 10
14			VEL	*10			Set speed
15							
16			PCLR	1	1		Clear position 1
17			SLEN	3.3			
18			VAL	199	8		Set axis 1 position in variable 199
19			PPUT	1	1		Set axis 1 data
20							
21			VAL	199	15		Set axis 2 position in variable 199
22			PPUT	2	1		Set axis 2 data
23			MOVL	1			Move
24			EXSR	1			Send OK
25			EDIF				
26							
27			GOTO	1			
28							-----------
29			BGSR	1			OK send subroutine
30			SCPY	1	‘OK’		Set OK
31			SPUT	3	13		Set CR
32			SPUT	4	10		Set LF
33			WRIT	1	1		Send
34			EDSR				

[7] Number of SIO Channels for each Controller

The channel numbers of SIO channels used in RS232C serial communication are as shown below.
Use OPEN and CLOS commands to specify SIO channel numbers that are used to open and close the RS232C serial communication line.
How many SIO channel numbers are available varies depending on the controller.

Controller	SIO channel number
XSEL-P/Q/PCT/QCT/PX/QX/R/ S/RX/SX/RXD/SXD	1 to 2
XSEL-J/JX TT, TTA, MSEL	1 *1
XSEL-K/KE/KT/KET, KX/KETX	1 *1 *2
SSEL, ASEL, PSEL	0 *1

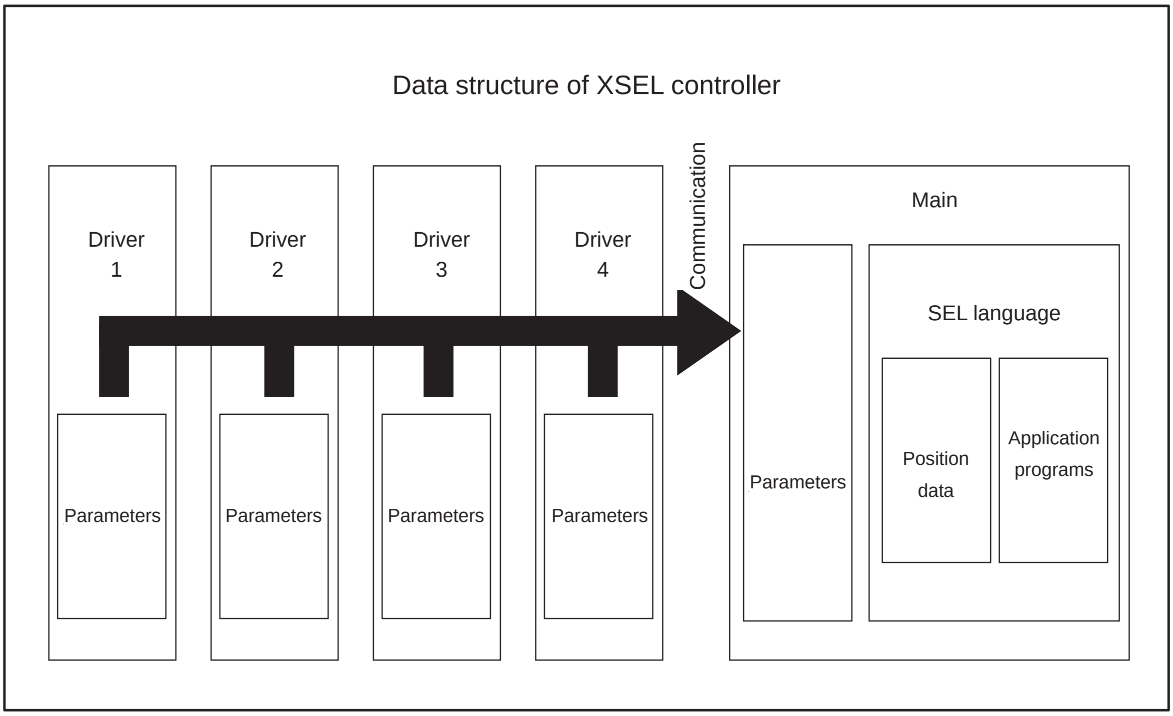
*1 This channel is used as the teaching-pendant connector port.

*2 If an expansion SIO board is installed, No. 2 and subsequent channels can be used.

3.7 Controller Data Structure and Saving of Data

3.7.1 XSEL-J/K/KE/KT/KET, JX/KX/KETX

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.

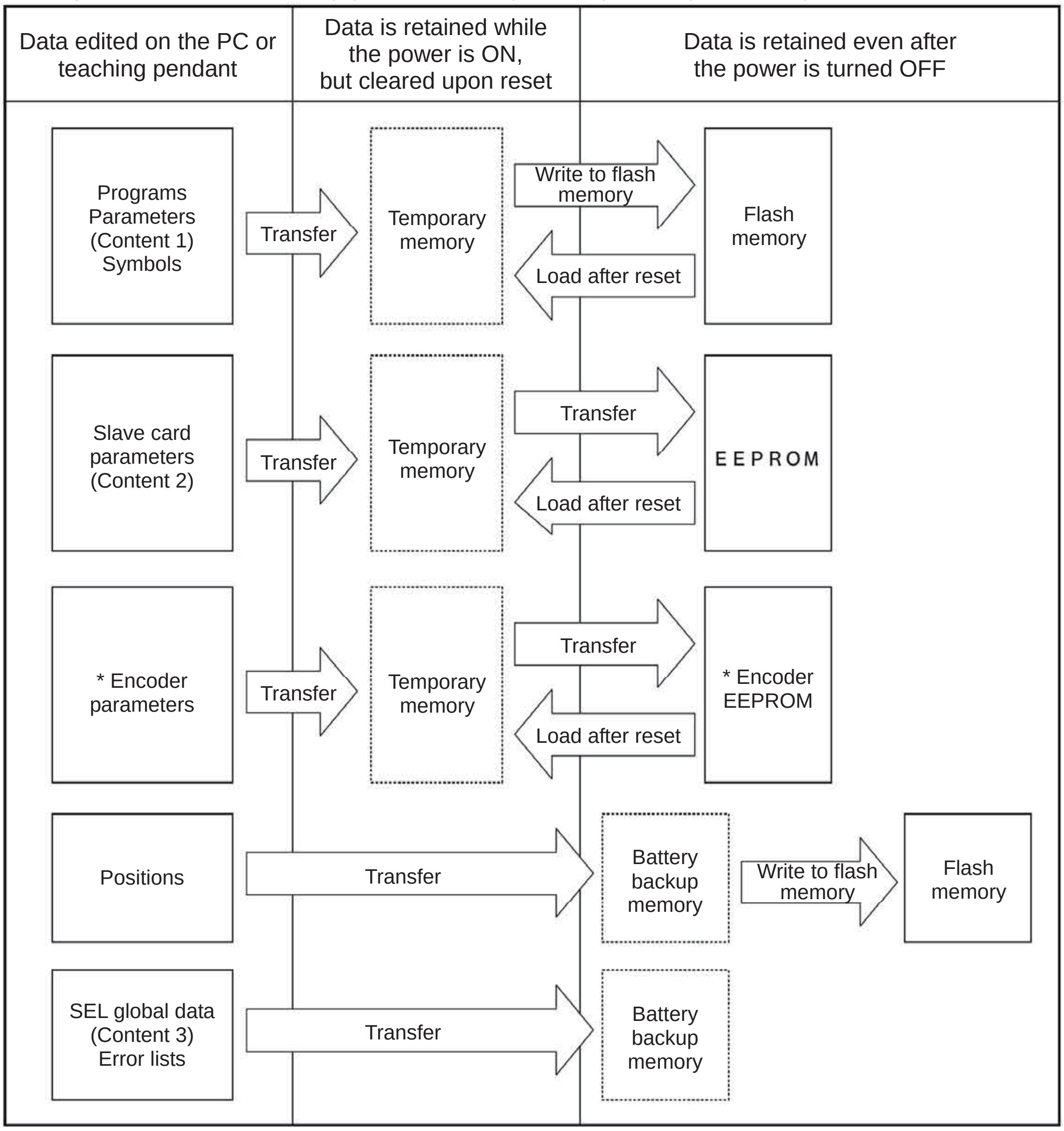


The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.

[2] Saving of data

XSEL controllers have areas saved by the backup battery and areas saved by the flash memory.
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
So that your important data is saved without fail, write the data to the flash memory.

[System-memory backup battery is used]
Other parameter No. 20 = 2 (System-memory backup battery installed)



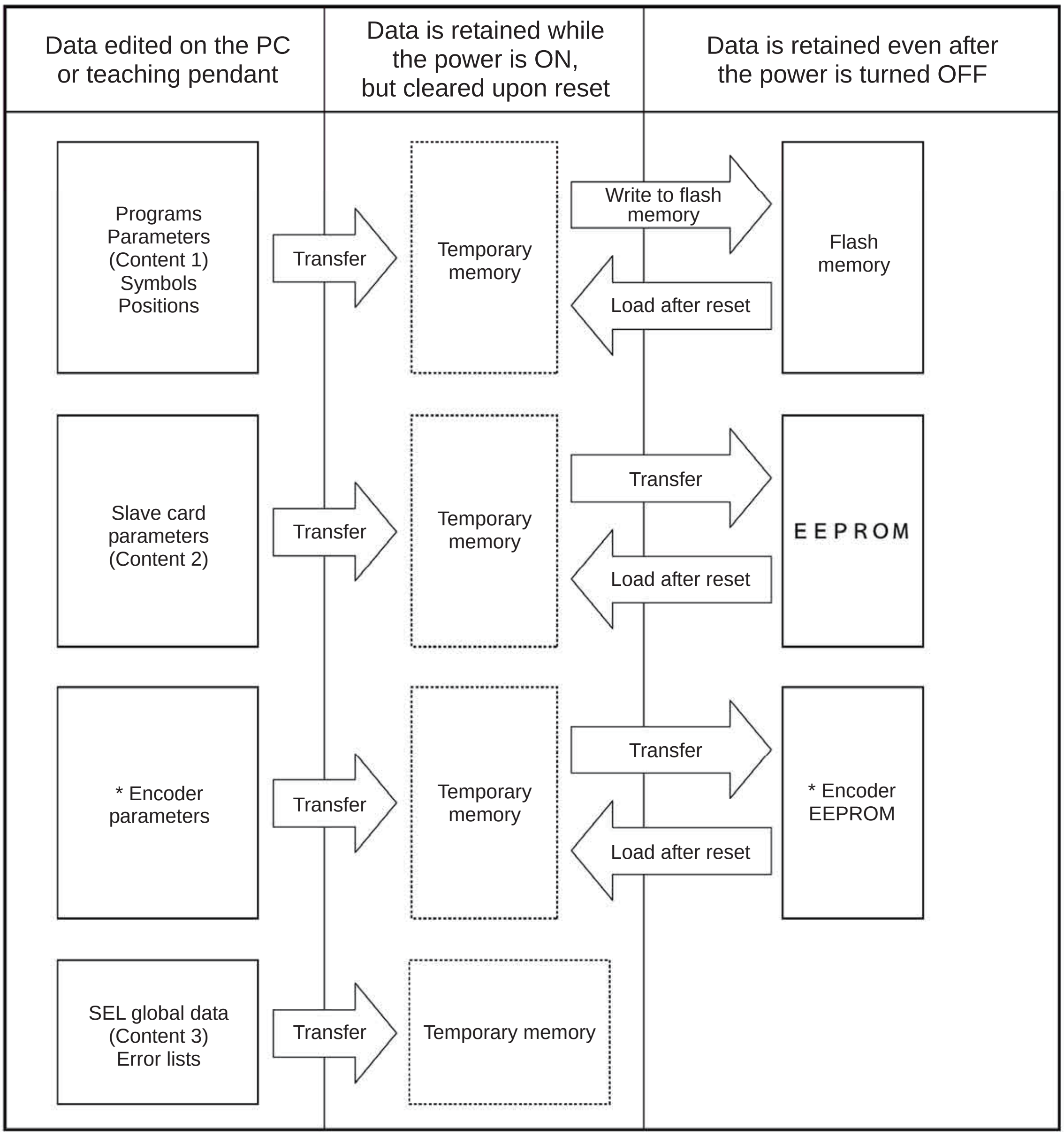
* Encoder parameters are stored not in the controller, but in the EEPROM of the actuator's encoder. Accordingly, they are loaded to the controller when the power is turned on or software is reset.

Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

Content 1: Parameters other than those included in Content 2 below and encoder parameters
Content 2: Driver card and I/O slot card (power card) parameters
Content 3: Flags, variables and strings

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)



Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.

[3] Notes

⚠ **Caution**

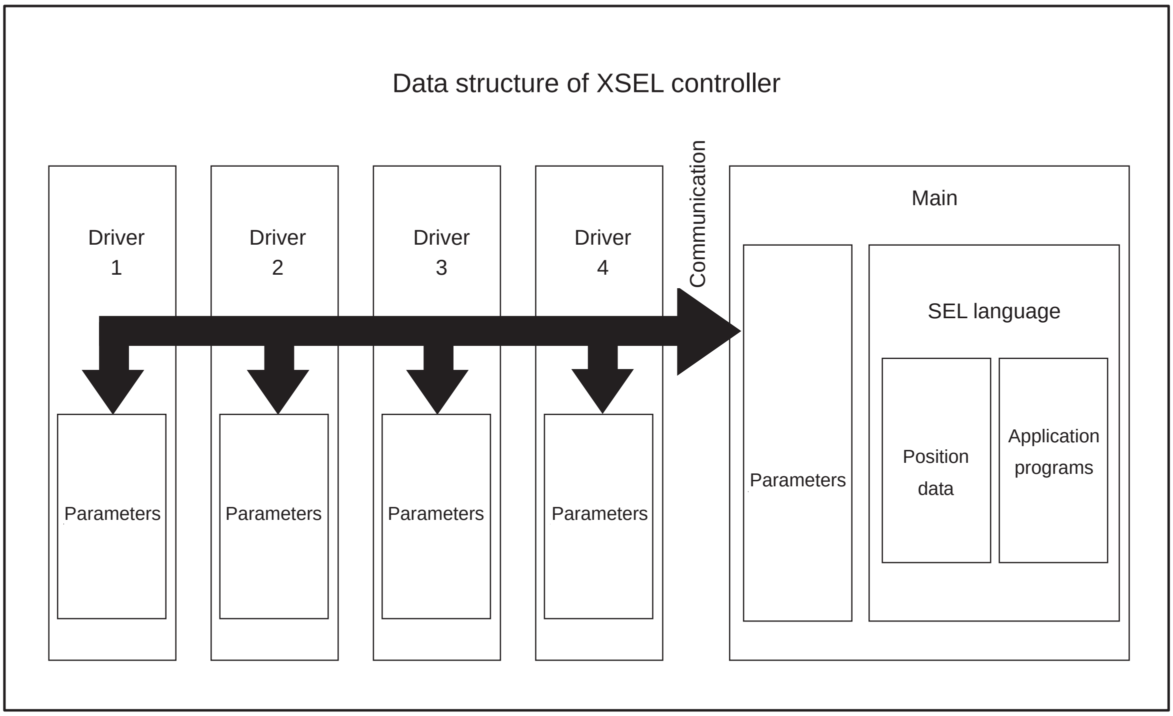
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned on or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

3.7.2 XSEL-P/Q/PCT/QCT, PX/QX

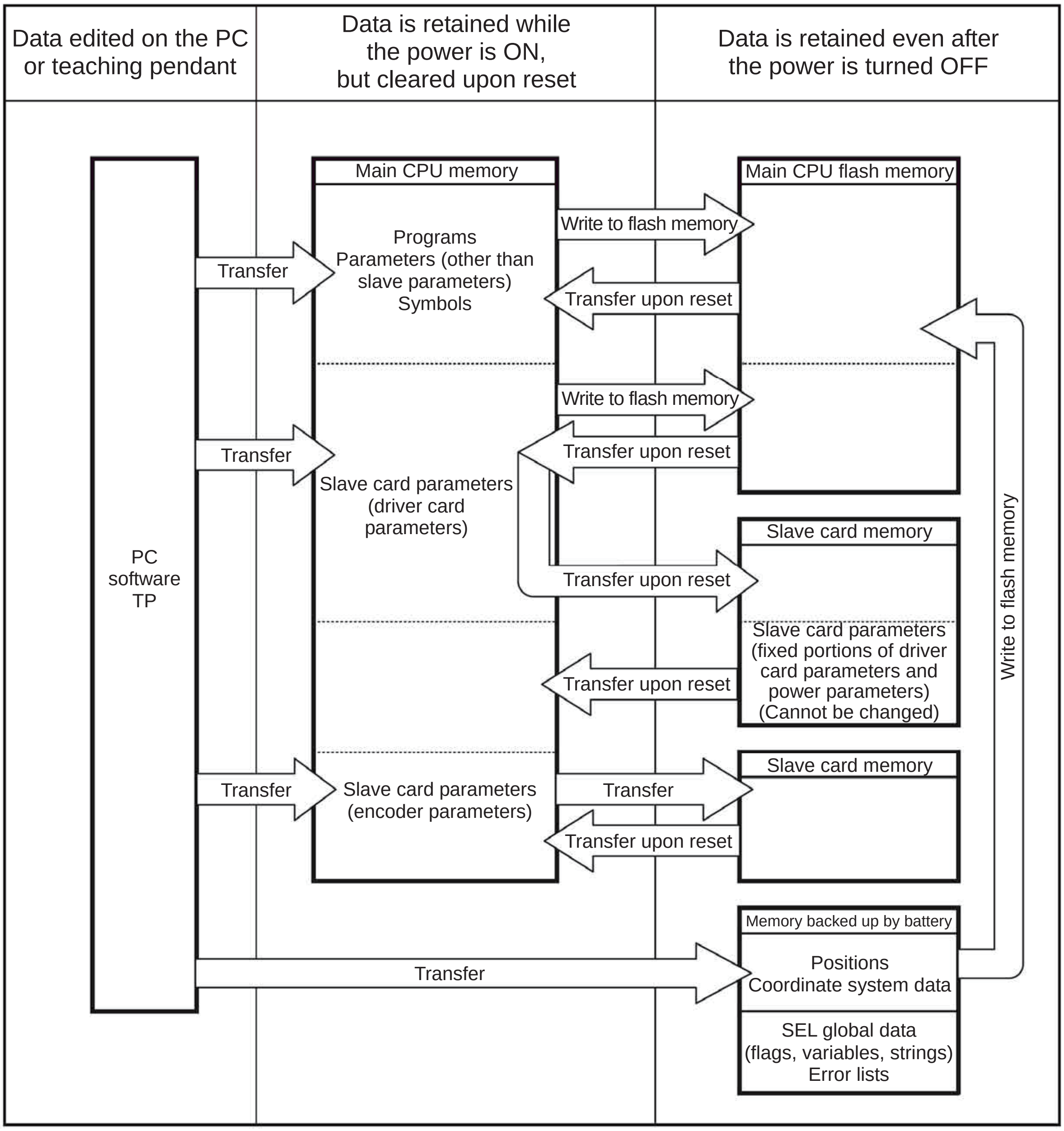
[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.



The customer must create position data and application programs. Certain parameters can be changed according to the customer's system.

[2] Saving of data
XSEL controllers have areas saved by the backup battery and areas saved by the flash memory.
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
So that your important data is saved without fail, write the data to the flash memory.

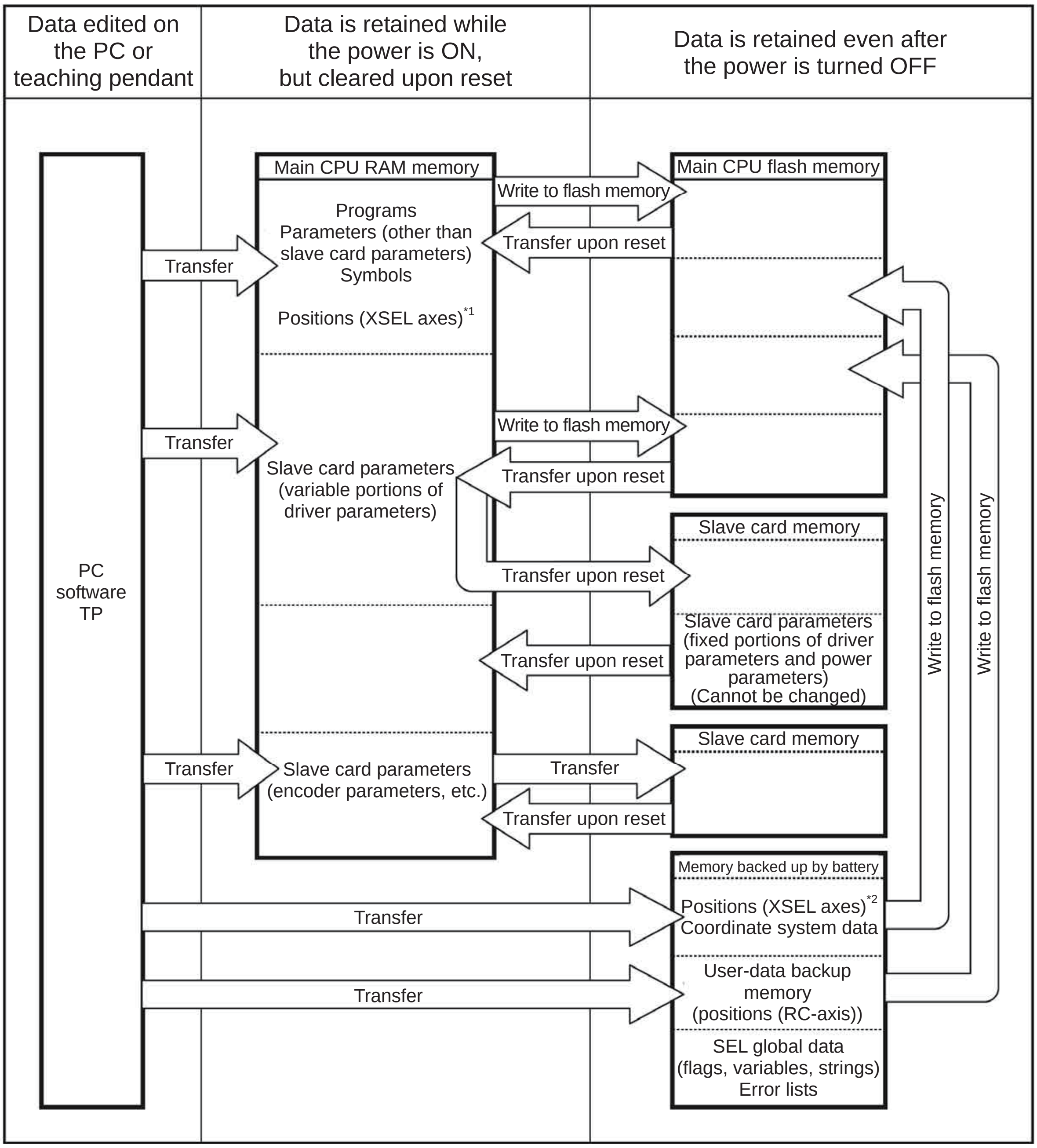
[System-memory backup battery is used]
1) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch not available, memory capacity 16M)
Other parameter No. 20 = 2 (System-memory backup battery installed)



Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

2) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch available, memory capacity 32M)

Other parameter No. 20 = 2 (System-memory backup battery installed)



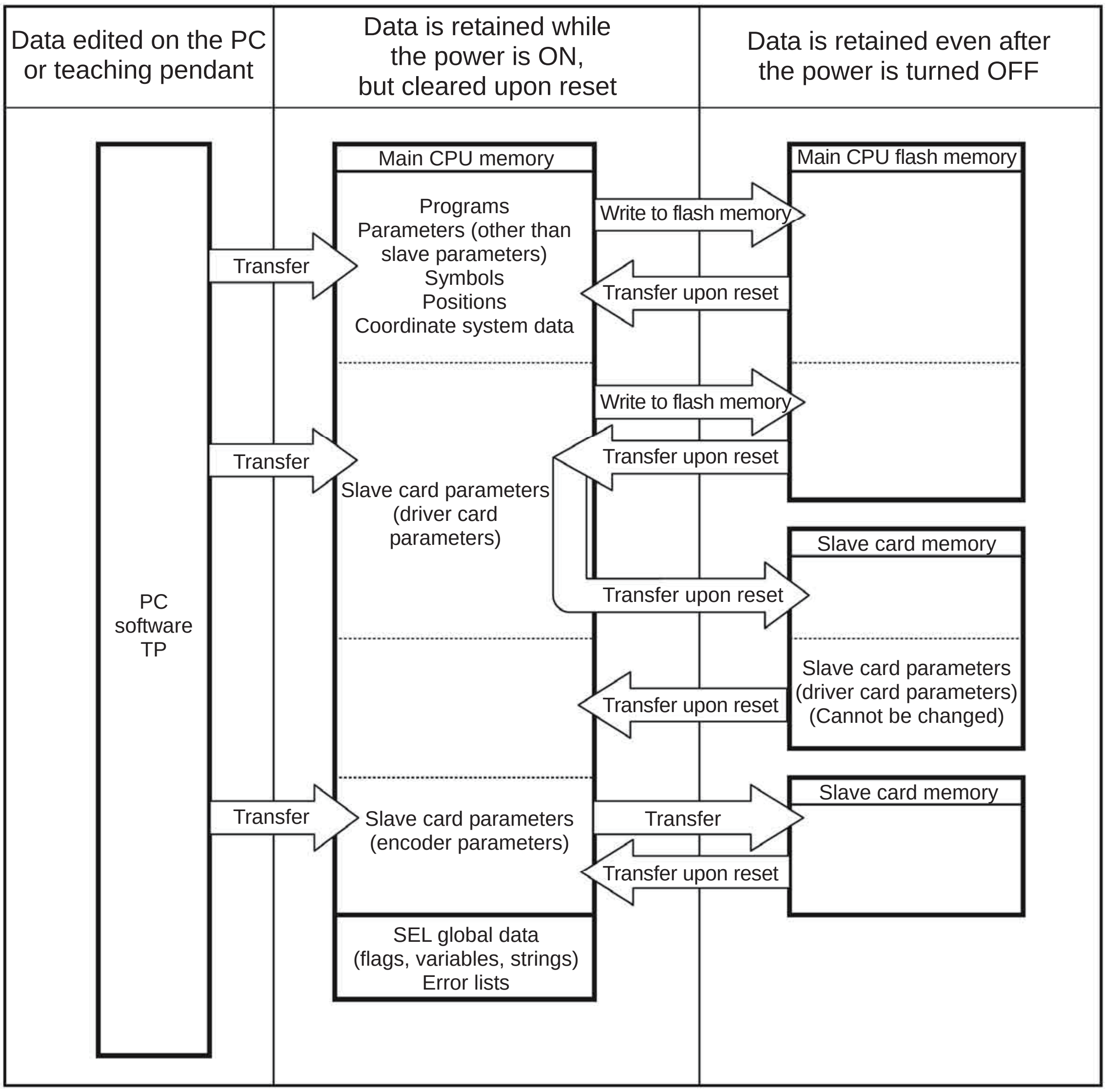
Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

*1 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 10001 to 20000.
*2 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 1 to 10000.

[System-memory backup battery is not used]

1) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch not available, memory capacity 16M)

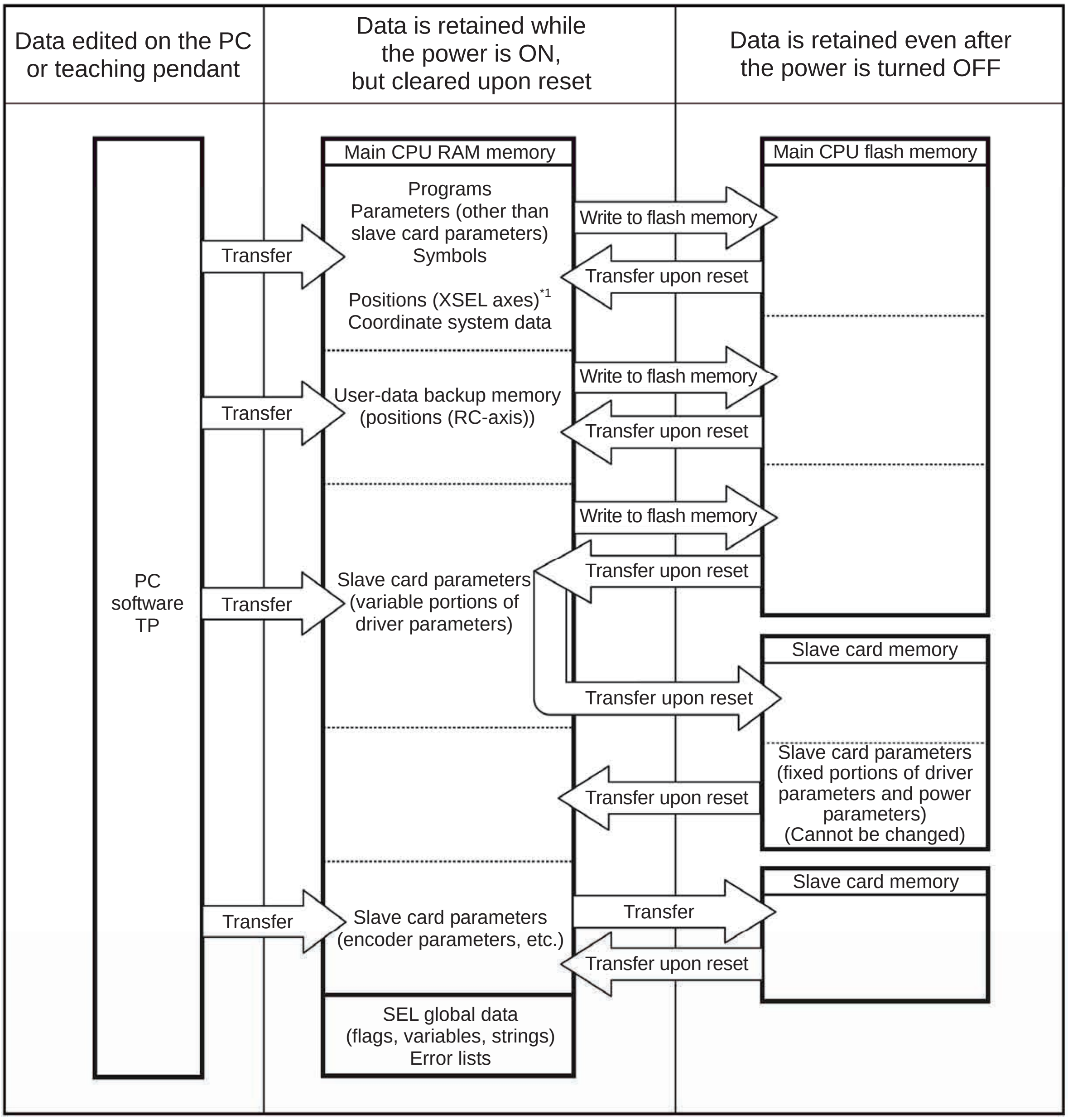
Other parameter No. 20 = 0 (System-memory backup battery not installed)



Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).
Note: SEL global data cannot be retained unless the backup battery is installed.

2) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch available, memory capacity 32M)

Other parameter No. 20 = 0 (System-memory backup battery installed)



Since programs, parameters, symbols, positions and user-data backup memory are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).
Note: SEL global data cannot be retained unless the backup battery is installed.

*1 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 1 to 20000.

[3] Notes

⚠ **Caution**

- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

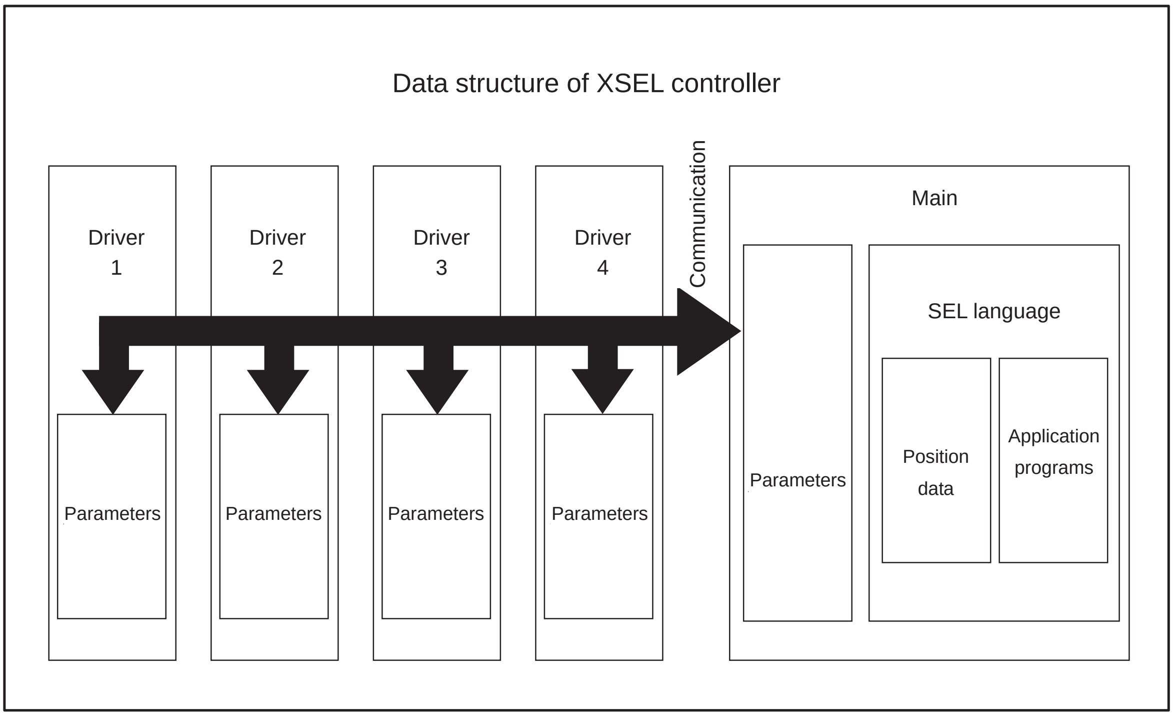
- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

- **Notes on increased number of positions**
  On controllers with increased memory capacity (with gateway function), the number of position data points has increased to 20000.
  Accordingly, take note of the following points:
  * If the memory backup battery is used (other parameter No. 20 = 2), position data is saved in the memory backup battery for position No. 1 to 10000, and in the main CPU flash ROM for position No. 10001 to 20000. Accordingly, <u>turning OFF the power or resetting the software without writing the position data to the flash ROM will cause the data of position No. 10001 to 20000 to be cleared and the data previously written to the flash ROM will be loaded the next time the controller is started.</u> To retain your data, therefore, make sure you write it to the flash ROM. If the memory backup battery is not used (other parameter No. 20 = 2), all position data of No. 1 to 20000 is saved in the main CPU flash ROM. In this case, again, write your data to the flash ROM to make sure the data is retained.

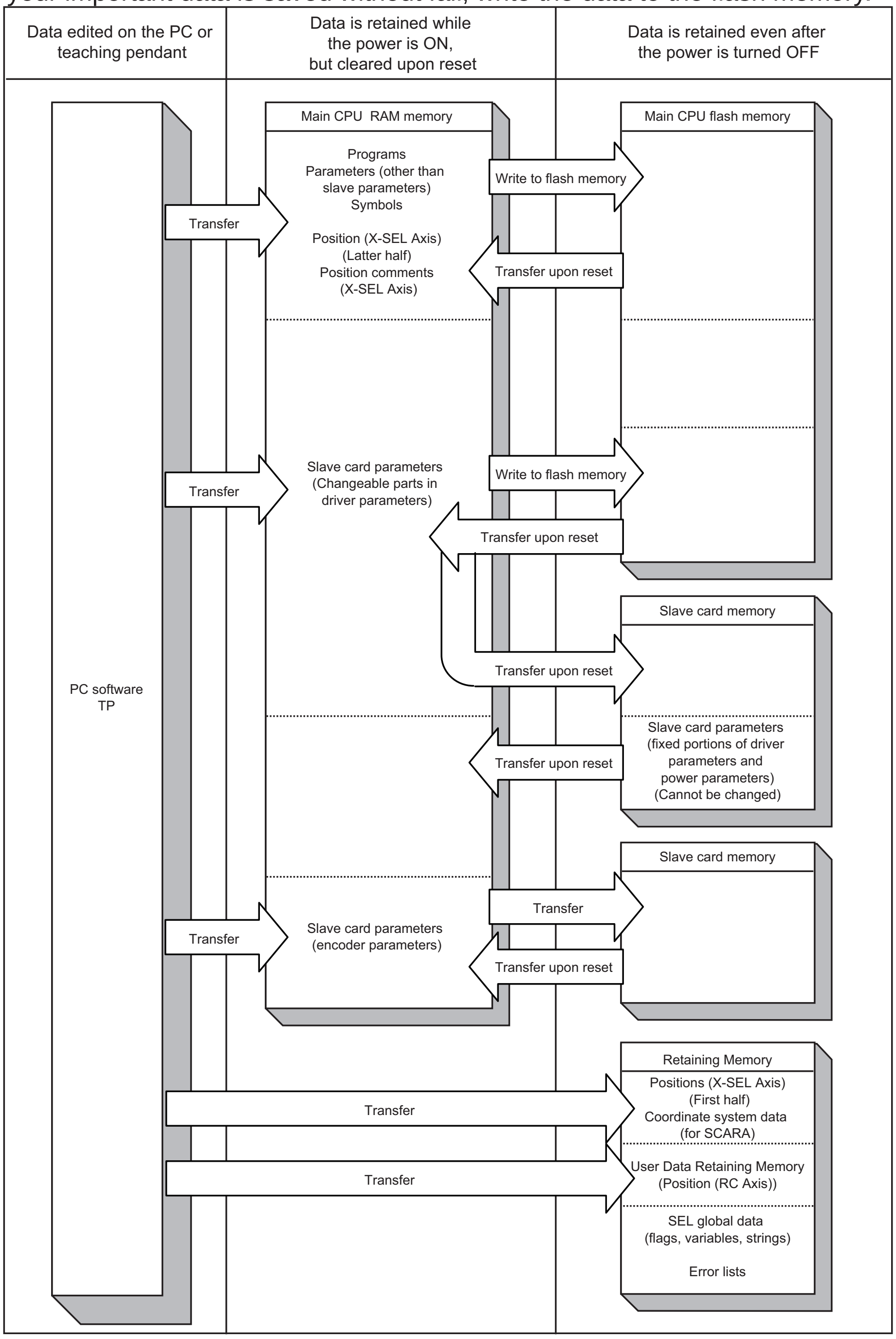
3.7.3 XSEL-R/S/RX/SX/RXD/SXD

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.



The customer must create position data and application programs. Certain parameters can be changed according to the customer's system.

[2] Saving of data
In XSEL controller, there is a storage domain with saving memory and a storage domain with flash memory.
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
So that your important data is saved without fail, write the data to the flash memory.



Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).
Note: The first half of the position data is stored in the saving memory while the second half in flash memory. The comment for each position data can be used in Positions No. 1 to 10000, and it is saved in the flash memory.

[3] Notes

⚠ **Caution**

- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

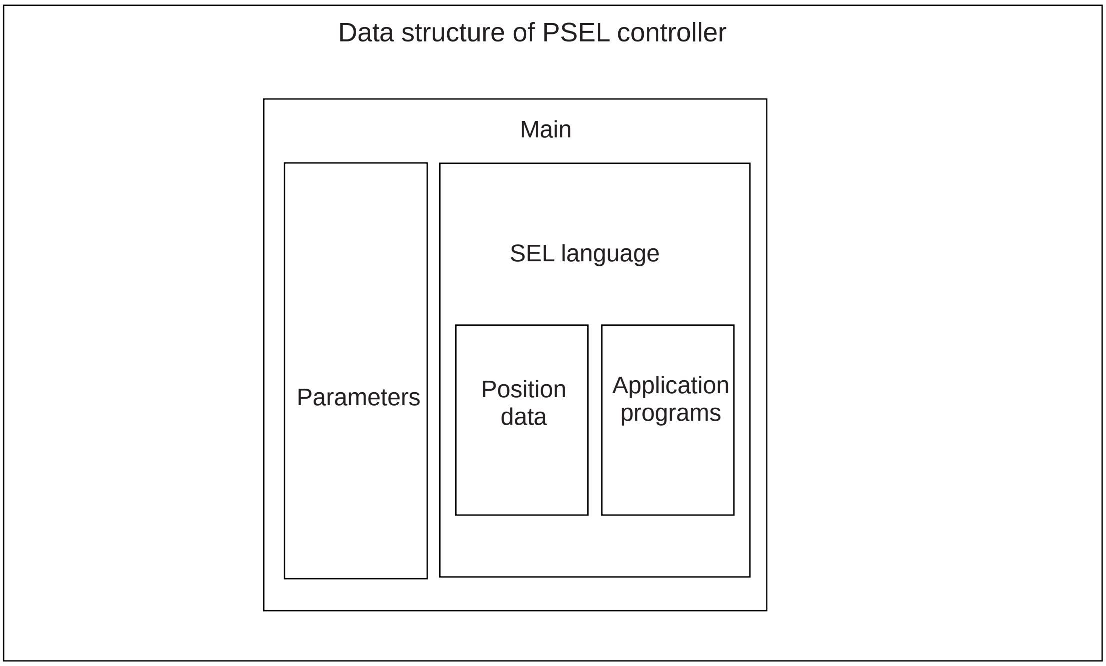
- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

- **Regarding Position Data Save**
  The storage domain for the position data is saving memory for the position (first half) and flash ROM of the main CPU for the position (second half). All the position data comment is to be stored in the flash ROM of the main CPU. Therefore, if the power is turned OFF or the software reset is conducted before writing to the flash ROM, the position (second half) and the position comment data are deleted, and the data that was previously written to the flash ROM is read out the next time the system is turned on. Do not fail to conduct the flash ROM writing when data saving is required.

- **About Initializing of Memory**
  Because the position data, maintenance information data and SEL global data will not be initialized (error data remains) even after an error is detected, make sure not to use the data without canceling it. To cancel an error, initialize the memory of the data which an error has been detected.
  For the position data (No. 10001 and after), do not fail to conduct the flash ROM writing at the same time after initializing.
  (Reference) How to Initialize Memory
  - Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
  - Coordinate System Data: Select [Memory Initialization] → [Coordinate System Definition Data] Menu in the PC software
  - User Retaining Memory: Select [Memory Initialization] → [User Retaining Memory] Menu in the PC software
  - SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
  - Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
    * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

3.7.4 ASEL, PSEL

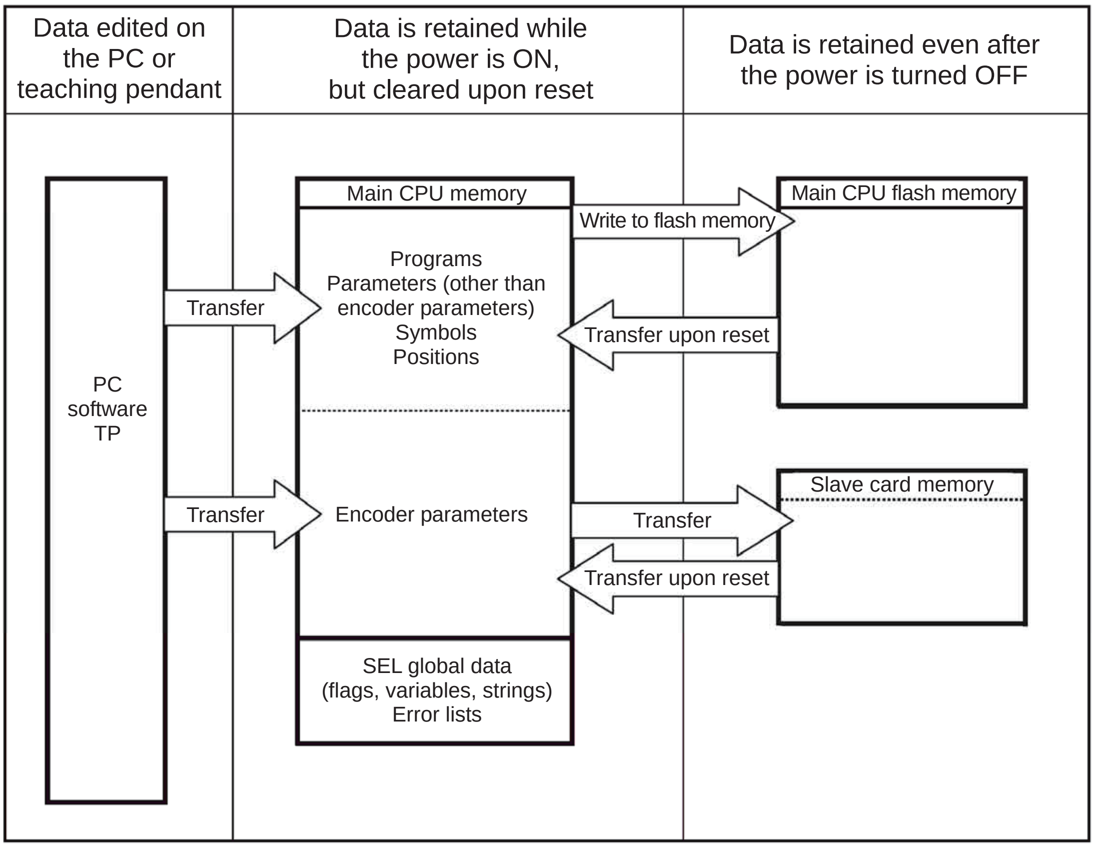
[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.



[2] Saving of data
On ASEL and PSEL controllers, data is saved as shown below.
Even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
To save the data without fail, be sure to write the data you want to save to the flash ROM.

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)

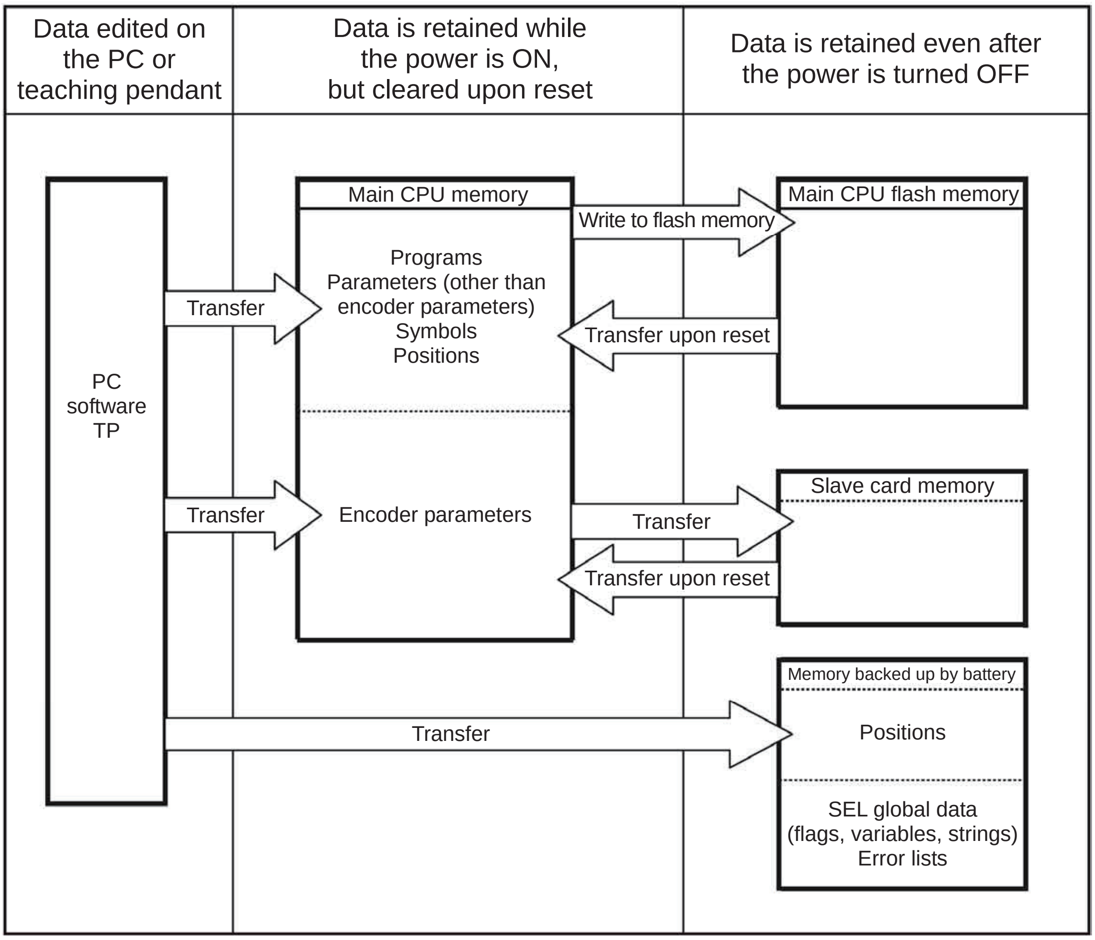


Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.
SEL global data is cleared once the control power is turned OFF or software is reset.
Error lists are cleared once the control power is turned OFF.

[System-memory backup battery (optional) is used]

The setting of other parameter No. 20 = 2 (System-memory backup battery installed) must be changed.



Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

[3] Notes

Caution

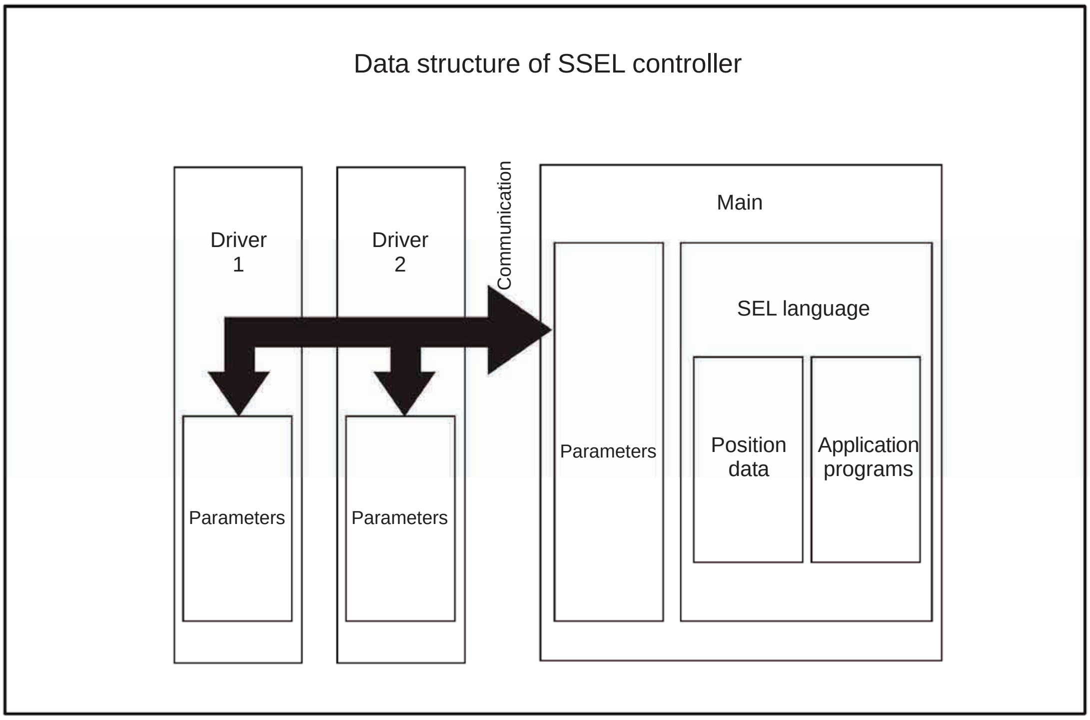
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

3.7.5 SSEL

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.

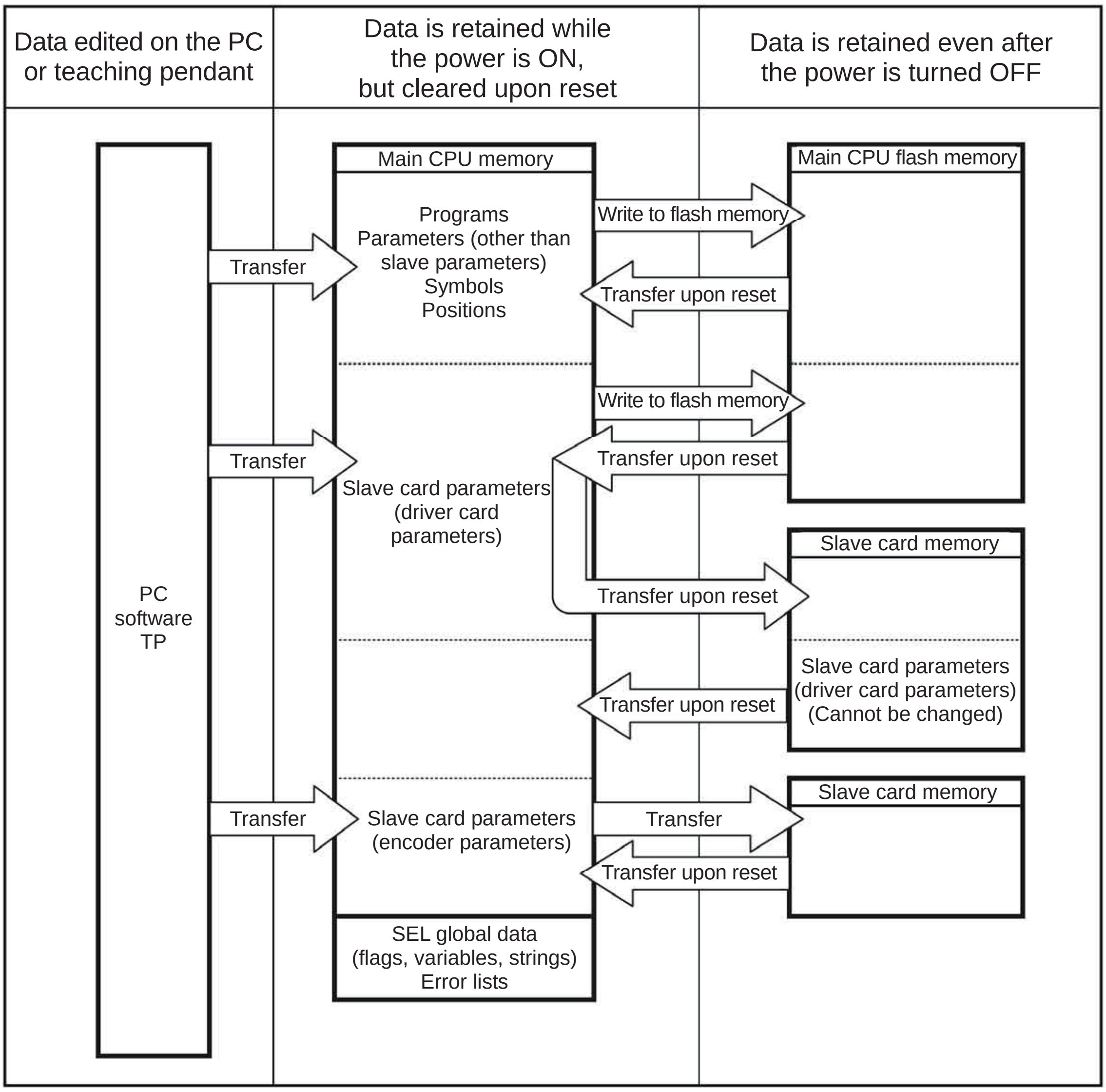


The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.

[2] Saving of data
On SSEL controllers, data is saved as shown below.
Even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
To save the data without fail, be sure to write the data you want to save to the flash ROM.

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)

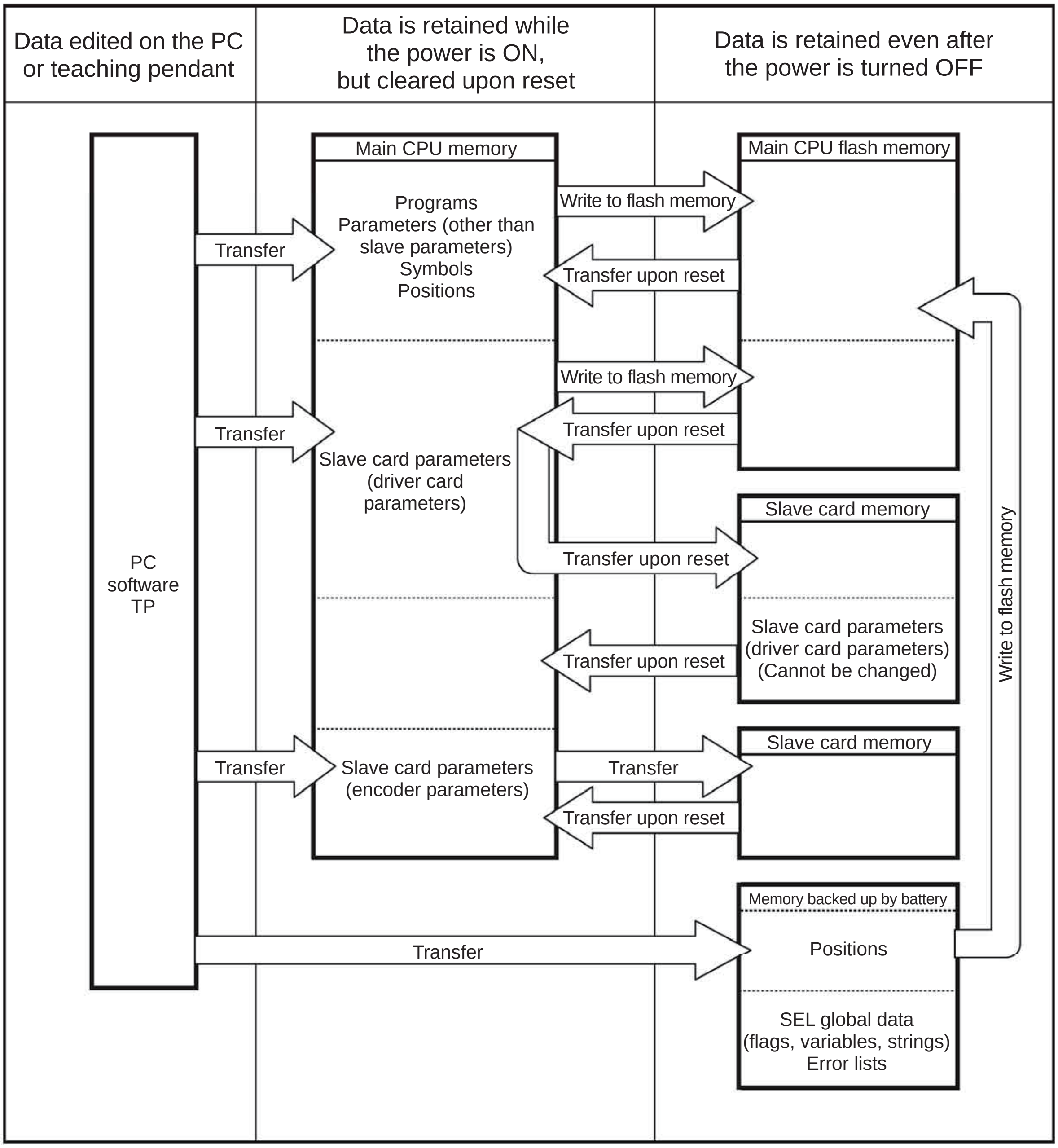


Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.
SEL global data is cleared once the control power is turned OFF or software is reset.
Error lists are cleared once the control power is turned OFF.

[System-memory backup battery (optional) is used]

The setting of other parameter No. 20 = 2 (System-memory backup battery installed) must be changed.



Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

[3] Notes

⚠ **Caution**

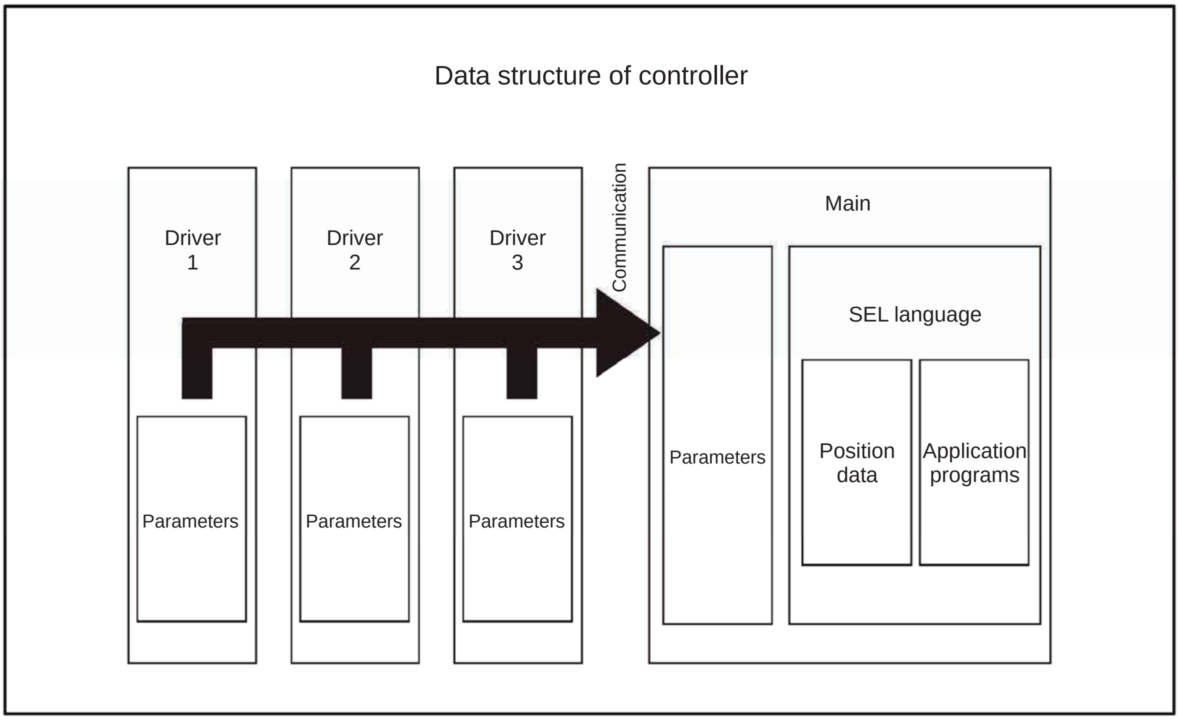
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

3.7.6 TT/TTA

[1] Data structure
The controller module of a tabletop robot contains parameters as well as position data and application programs used to drive the SEL language.



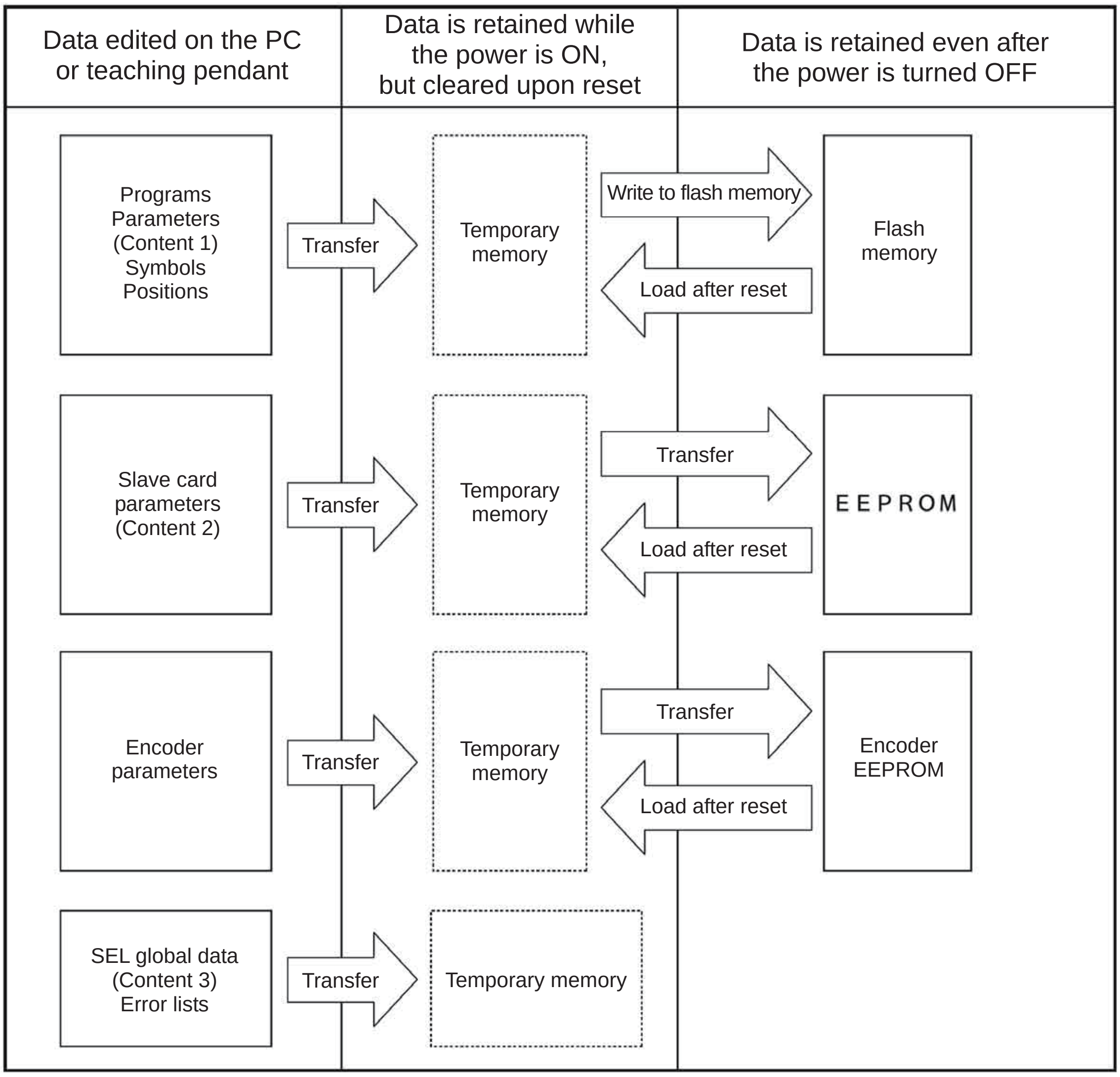
The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.
[Refer to tabletop robot TT Instruction Manual provided separately.]

[2] Data Saving of TT

When data created/edited using the PC software or teaching pendant is transferred to the controller (by pressing the WRT key if you are using the teaching pendant), the data is temporarily stored in the controller's memories. Accordingly, such data will be cleared once the power is turned off or software is reset (restarted).

If you want your data to be retained, be sure to write it to the flash memory.

Note: Global data (variables, flags, strings) is cleared once the power is turned OFF or software is reset (restarted) (global data cannot be retained after the power is turned OFF). Error lists are retained after the software is reset, but cleared if the power is turned OFF.



Content 1: Parameters other than those included in Content 2 below and encoder parameters
Content 2: Driver card and I/O slot card (power card) parameters
Content 3: Flags, variables and strings

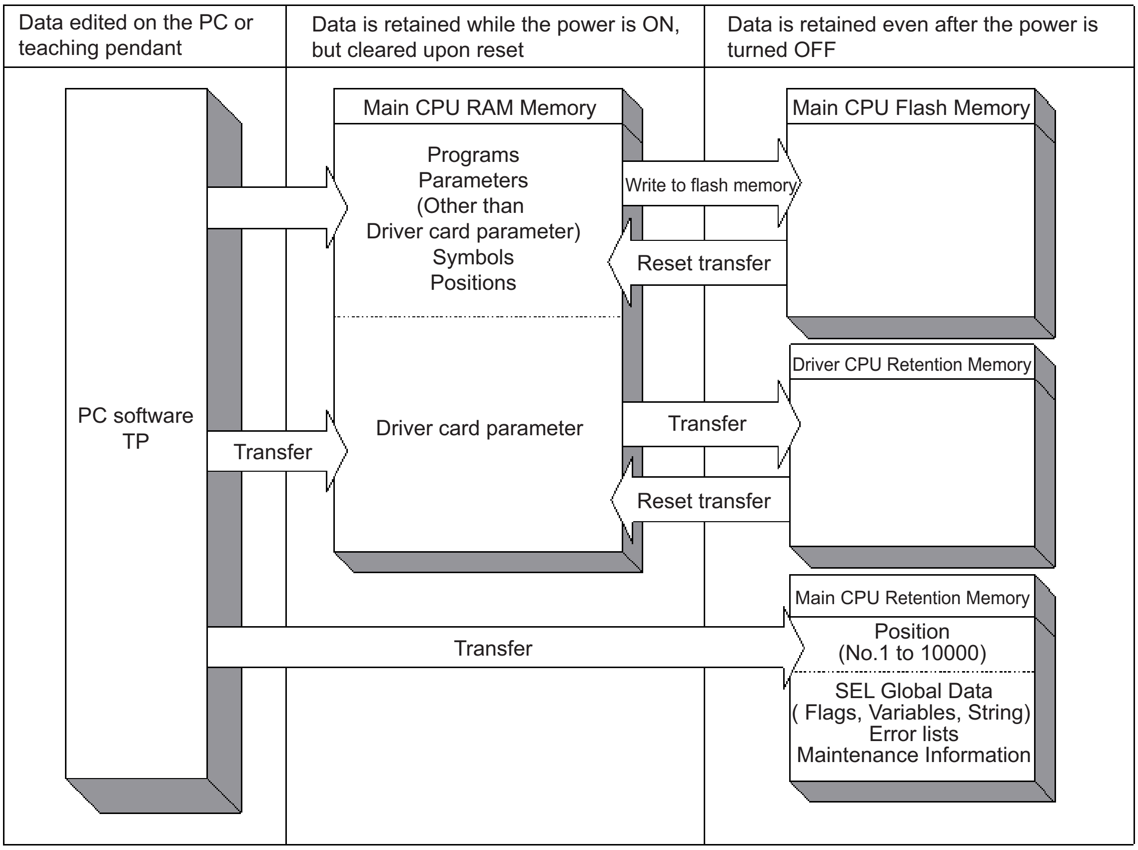
Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

[3] Notes

> ⚠ **Caution**
> **Notes on transferring data and writing it to the flash memory**
> Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

[4] Data Saving of TTA

In the retaining memory (FeRAM), Position data (No. 1 to 10000), SEL global data, error list and maintenance information are stored for backup in standard with no battery.
Position (No. 1 to 10000) is stored only in the retaining memory. (It is not necessary to have the flash ROM writing.)



(Note 1) Do not attempt to turn the power OFF during the memory initialization (position, global variables and flags) or the maintenance information initialization. As the initializing process terminates incomplete, errors described below* may be generated in the next startup. Redo initializing in case the power is turned OFF accidently, and an error is generated. (* Error No. 4A4, 69E, 6C7 or 826)

(Note 2) As the position data, maintenance information data and SEL global data will not be initialized even if an error gets detected (error data can be seen as it is), do not attempt to use the data as it is. To cancel the error, initialize the memory in the data the error was detected.
For position data (No. 10001 to 30000), conduct also the flash ROM writing after initializing.

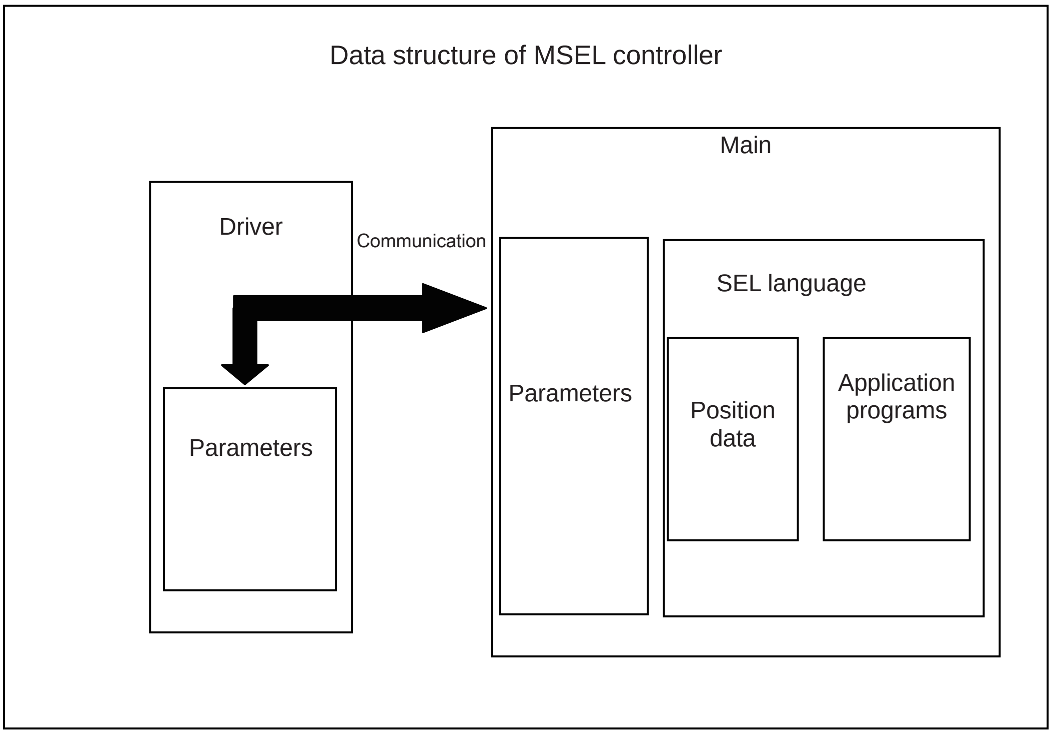
(Reference) How to Initialize Memory

- Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
- SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
- Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
  * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

3.7.7 MSEL

[1] Data structure

The controller module of a MSEL contains parameters as well as position data and application programs used to drive the SEL language.

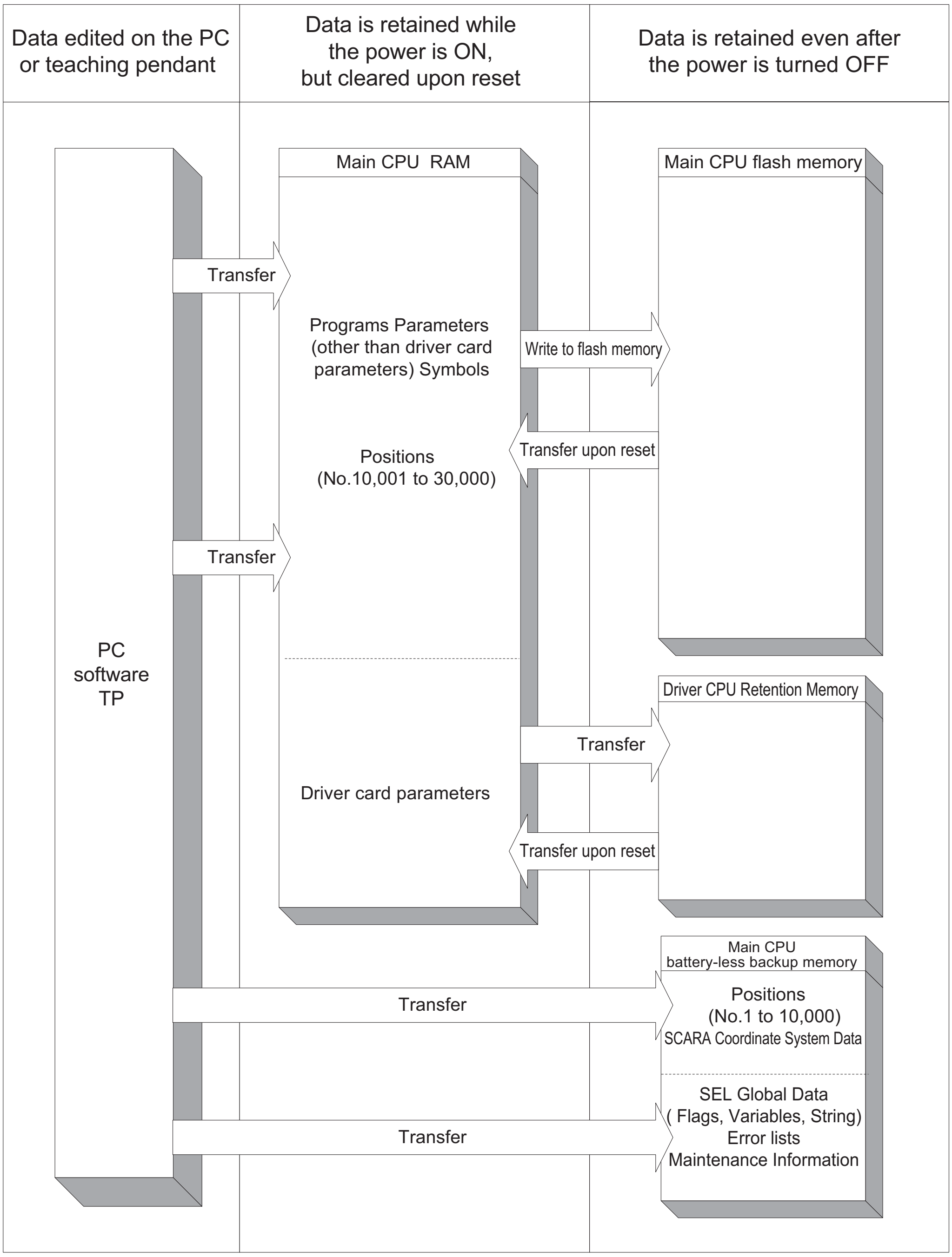


The customer must create position data and application programs.
Certain parameters can be changed according to the customer’s system.

[2] Saving of data

On MSEL controllers, data is saved as shown below.
Even if you transfer data via the PC software or teaching pendant, the data, except for some (Note), is only written to the memories temporarily and will be cleared once the power is turned off or controller is reset.
To save the data without fail, be sure to write the data you want to save to the flash ROM.

(Note) The position data (No. 1 to 10000), SEL global data, error list, maintenance information and SCARA coordinate system data are stored in the battery-less backup memory (FRAM). There is no need of flash ROM writing.



(Note 1) Do not attempt to turn the power off while initializing the memories (position, global variables and flags) or maintenance information. It may cause to generate such as an error* in the next startup due to incomplete of initializing process. Have an initializing process again in case the power is turned off accidently. (* Error No. 4A4, 69E, 6C7, 826)

(Note 2) Because the position data, maintenance information data and SEL global data will not be initialized (error data remains) even after an error is detected, make sure not to use the data without canceling it. To cancel an error, initialize the memory of the data which an error has been detected.
For the position data (No. 10001 to 30000), do not fail to conduct the flash ROM writing at the same time after initializing.

4. Program Edit

4.1 Each Type of Data Available to Handle on the Program and its Range

In SEL language, separate areas are provided for each task such as I/O port, variables, flags, etc.
Some areas are separated to the global area and local area. Data set to the global area can be read and written from multiple programs.
The global domain is backed up in the controller battery for the models except for XSEL-R*/S*.
Data in local area gets cleared each time the program is booted.
In the following, explains about the area and range.

Function	Global area		Local area		Remarks
	Range	Total number	Range	Total number	
Input port	000 to 299	300			
Output port	300 to 599	300			
Extended Input Ports	1000 to 3999	3000			Applied for XSEL-P/Q/PCT/QCT and XSEL-R/S/RX/SX/RXD/SXD
Extended Output Ports	4000 to 6999	3000			
Flag	600 to 899	300	900 to 999	100	
Variable (integer)	200 to 299	100	1 to 99	99	99 is a special variable used in IN, INB, OUT and OUTB Variable (integer) commands, etc.
	1200 to 1299	100	1001 to 1099	99	
Variable (real number)	300 to 399	100	100 to 199	100	199 is a special variable used in PPUT, PGET and PAPG commands, etc.
	1300 to 1399	100	1100 to 1199	100	
String	300 to 999	700	1 to 299	299	
Tag number			1 to 256	256	
Sub routine number			1 to 99	99	
Work coordinate system number	0 to 31	32			For SCARA robots
Tool coordinate system number	0 to 127	128			For SCARA robots
Simple contact check zone number	1 to 10	10			For SCARA robots
Zone number	1 to 4	4			For single-axis/Cartesian robots
Palletizing number			1 to 10	10	
Axis number	1 to 8	8			Varies depending on the controller.
Axis pattern	0 to 11111111				Varies depending on the controller.
Program number (XSEL-P/Q/PX/QX/PCT/QCT, XSEL-R/S/RX/SX/RXD/SXD, SSEL)	1 to 128	128			
Program number (XSEL-J/K/KE/KTKET/JX/KX/KETX, TT, ASEL/PSEL)	1 to 64	64			
Program number (TTA, MSEL)	1 to 256	256			

Function		Global area Range	Global area Total number	Local area Range	Local area Total number	Remarks
Position number	XSEL-R/S/RX/SX/RXD/SXD	1 to 53332 (MAX)	53332 (MAX)			Depend on how many axes are to be used
	XSEL-P/Q/PX/QX/PCT/QCT, SSEL	1 to 20000	20000			
	XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT	1 to 3000	3000			
	ASEL/PSEL	1 to 1500	1500			
	TTA, MSEL	1 to 30000	30000			
Position comments (Half-sized 32 characters)	XSEL-R/S/RX/SX/RXD/SXD			1 to 10000	10000	
Task level		0: NORMAL/ 1: HIGH	2			Comment can be added only in Positions No. 1 to 10000
SIO channel number	XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD	1 to 2	2			
	XSEL-J/JX, TT	1	1			To be communized with teaching and PC software
	XSEL-K/KE/KT/KET/KX/KETX	1	1			
	SSEL/ASEL/PSEL	0	1			
WAIT timer				1		TIMW command
1-shot pulse timer				16 (Can be operated simultaneously.)		BTPN, BTPF command
Ladder timer				Use local area flags. 900 to 999	100	TIMR command
Virtual input port (SEL system → SEL user program)		7000 to 7299	300			
Virtual output port (SEL user program → SEL system)		7300 to 7599	300			
Number of symbol definitions	XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD	1000				
	XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT, TTA, MSEL	1000				
	SSEL/ASEL/PSEL	500				
Number of symbol used in commands	XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD	5000 (including string literals)				
	XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT, TTA, MSEL	5000 (including string literals)				
	SSEL/ASEL/PSEL	2500 (including string literals)				
Number of recorded history	XSEL-R/S/RX/SX/RXD/SXD, TTA, MSEL	400				
	XSEL-P/Q/PCT/QCT/PX/QX	200				
	XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT	200				
	SSEL/ASEL/PSEL	100				

4.2 Setting of Function and Values

Explanation below shows how you should handle the I/O port and how you should take the variables in your mind when you create a program with SEL language.

4.2.1 Handling of I/O Port

Refer to “2.1 I/O Signal” for I/O ports.

[1] Input ports
These ports are used as input ports for limit switches, sensor switches, etc.

Input number assignment
000 to 031 (standard)

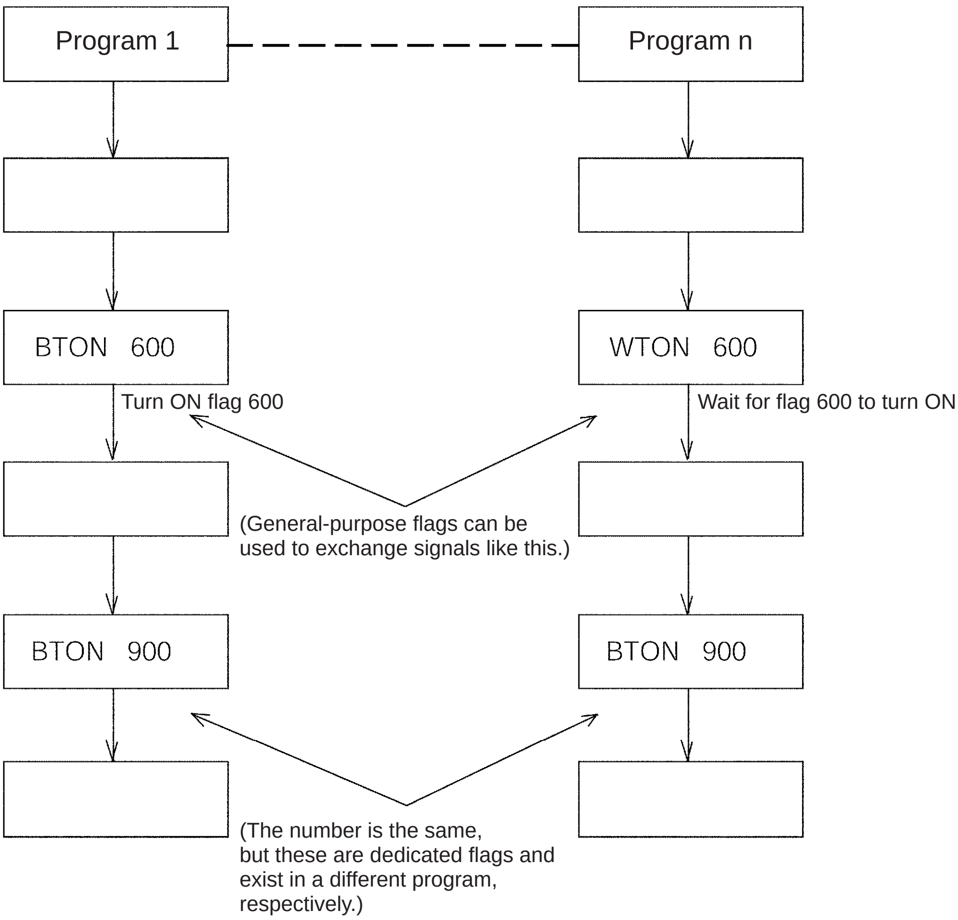
[2] Output ports
These ports are used as various output ports.

Output number assignment
300 to 315 (standard)

4.2.2 Handling (Setting and Resetting) of Flags

Unlike their literal meaning, flags are actually “memories” where data is set and reset. Flags correspond to “auxiliary relays” in sequencers.
Flags are classified into two types: general-purpose flags (global flags) that are assigned numbers from 600 to 899 and usable in all programs, and dedicated flags (local flags) that are assigned numbers from 900 to 999 and usable only in each program.
The general-purposed flags (global flags) can be saved (in the battery backup or saving memory, depends on the models) even after the power is turned OFF.
Dedicated flags (local flags) will be cleared once the power is turned OFF.

Flag No.	600 to 899	Usable in all programs.	"General-purpose flags (global flags)"
Flag No.	900 to 999	Usable only in each program.	"Dedicated flags (local flags)"



4.2.3 How to Deal with Values and Variables

(1) How to Deal with Values

If the last digit of the set value is H, set with hexadecimal number.
Refer to the following.
Input the value of hexadecimal number transformed from the binary number.

●Binary number

Binary number expresses a numeral figure with using 2 numbers, 0 and 1.
The number increases in the order of 0, 1, and then the number of digit increases, and goes 10, 11 …

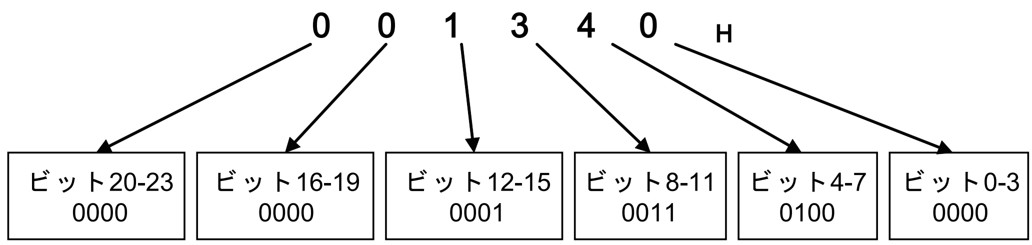
Decimal number	0	1	2	3	4	5	6	7	8	9	10
Binary number	0	1	10	11	100	101	110	111	1000	1001	1010

●Hexadecimal number

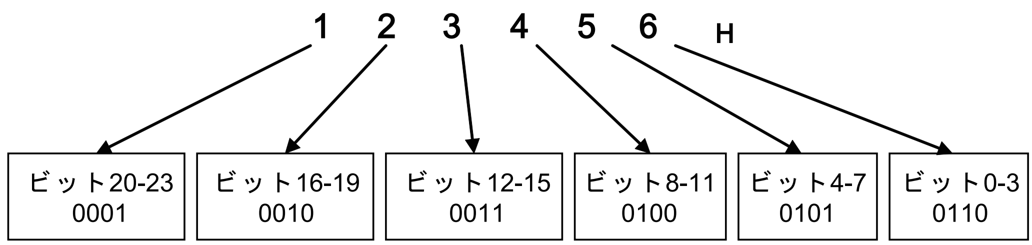
Hexadecimal number expresses a numeral figure with using numbers from 0 to 9 and alphabets from A to F. The number increases in the order of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B,C, D, E, F, and then the number of digit increases, and goes 10, 11, …

Decimal number	0 to 9 (Same for decimal and hexadecimal numbers)	10	11	12	13	14	15	16
Hexadecimal number		A	B	C	D	E	F	10

Example 1 : 001340_H

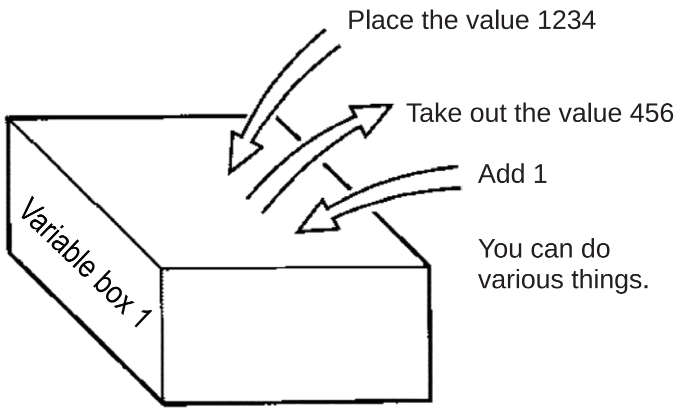


Example 2 : 123456_H



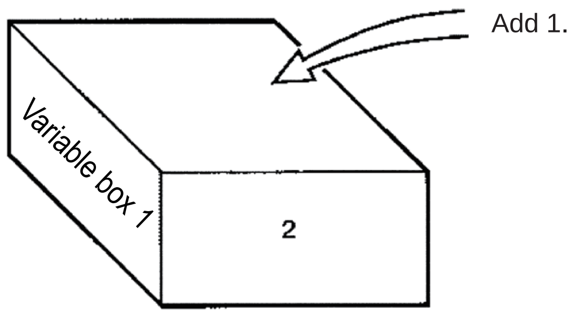
(2) Types and Handling of Variables

1) Meaning of variables
"Variable" is a technical software term. Simply put, a variable is a "container in which a value is placed". You can use variables in many different ways such as placing a value in a variable, taking a value out of a variable, and adding or subtracting a value to/from a variable, to name a few.



Command	Operand 1	Operand 2
ADD	1	1

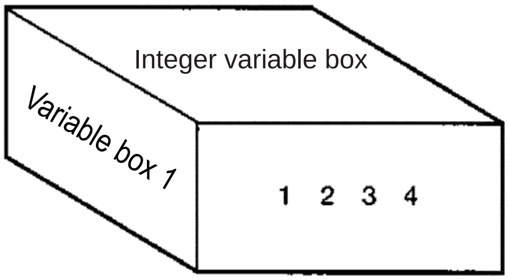
With this command, if 2 is already placed in the box of variable 1 as shown, then 1 is added and the content of variable 1 becomes 3.



(2 is already placed inside.)

2) Types of variables
Variables are classified into two types as explained below.

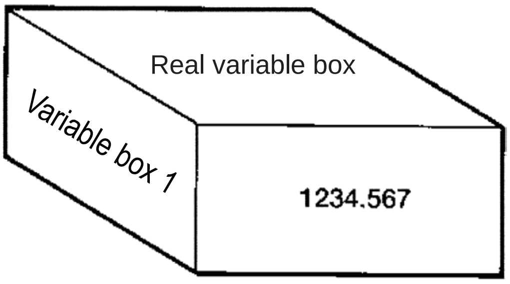
[Integer variables]
These variables cannot handle decimal points.
[Example] 1234



Integer variable No.	200 to 299 1200 to 1299	Usable in all programs.	“Global integer variables”
Integer variable No.	1 to 99 1001 to 1099	Usable only in each program.	“Local integer variables”

Caution
- Values from -9,999,999 to 99,999,999 can be entered in programs.
- Variable 99 is a special register used for integer calculations by the system.

[Real variables]
These variables are actual numbers and can also handle decimal points.
[Example] 1234.567
↑
(decimal point)

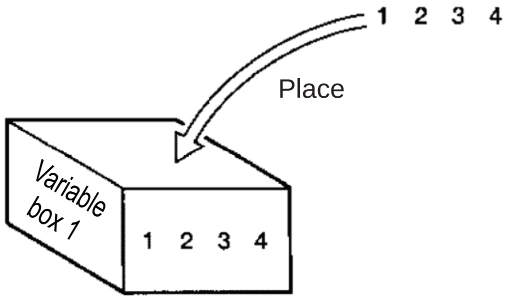


Real variable No.	300 to 399 1300 to 1399	Usable in all programs.	“Global real variables”
Real variable No.	100 to 199 1100 to 1199	Usable only in each program.	“Local real variables”

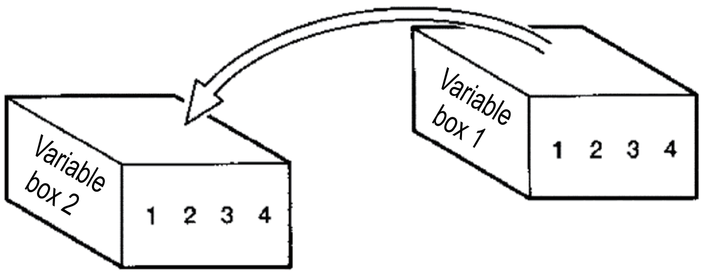
Caution
- Values from -99,999.9 to 999,999.9 (up to eight digits including the sign and decimal point) can be entered in programs.
- Variable 199 is a special register used for integer calculations by the system.

[Indirect specification of variables]
Variables are specified with a "*" (asterisk) appended to them.
In the example below, the content of variable box 1 is placed in variable box 2. If "1234" is in variable box 1, "1234" is placed in variable box 2.

Command	Operand 1	Operand 2
LET	1	1234



Command	Operand 1	Operand 2
LET	2	*1



This usage is called "indirect specification".

"*" is to be applied also when making an indirect specification of symbolized variables.

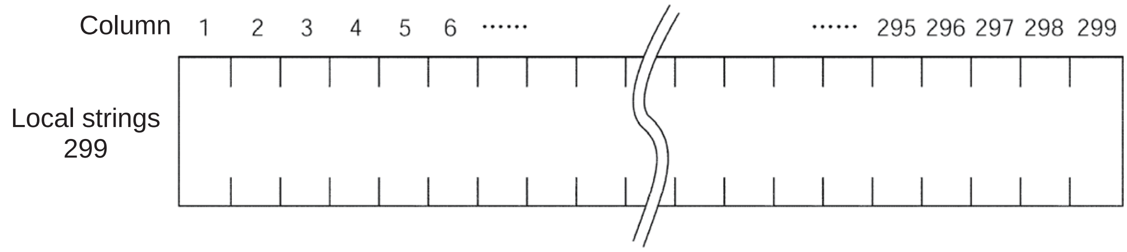
Command	Operand 1	Operand 2
LET	ABC	1
LET	BCD	2
AD	ABC	*BCD

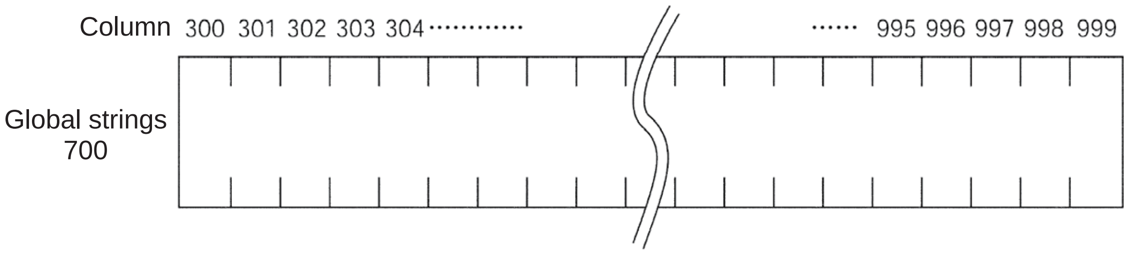
Place 1 in variable ABC.
Place 2 in variable BCD.
Add the content of variable BCD, or 2, to variable ABC.
(The content of variable ABC becomes 3.)

4.2.4 Specification Method for Local String and Global String

RS232C serial communication is implemented basically by means of exchange of character strings.
These character strings are called “string”.
Strings sent in the communication transmission format can be used freely in programs, or specifically they are stored in boxes (columns) in which strings are placed.
These string are classified into global string that can be read or written in all programs, and local string that can be read or written only in each program.
String are differentiated by the range of their number.

	Column number
Global string	300 to 999 (700)
Local string	1 to 299 (299)

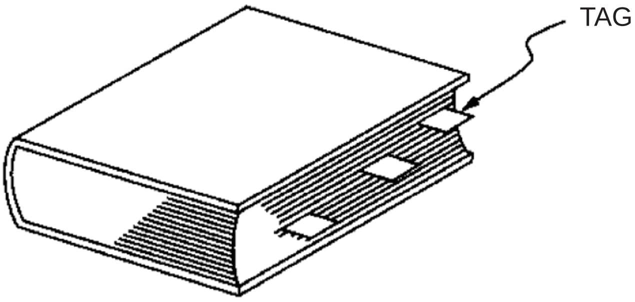




The characters constituting a string are stored one by one in each of these fields.
The position of a given field in a string is expressed by column X, and the column to store each character in can be set freely using a command.

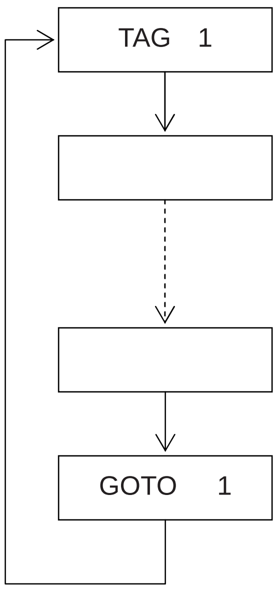
4.2.5 Handling of Tag Numbers

A “TAG” is a “heading”.
You may stick labels on pages you want to read frequently. Tags are used for the same purpose.
The destination to jump to where you specify in the jump command “GOTO” is a “TAG”.



Command	Operand 1
TAG	Tag number (integer of 1 to 256)

Usable only in each program.



5. SEL Commands

5.1 How to Read Explanation of Command

How a command is explained is described using an example of LET command.

● LET (Assign)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	LET	Variable number	Data	ZR

[1] SEL language structure

Applicable models							
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT
○	○	○	○	○	○	○	○

[2] Applicable models

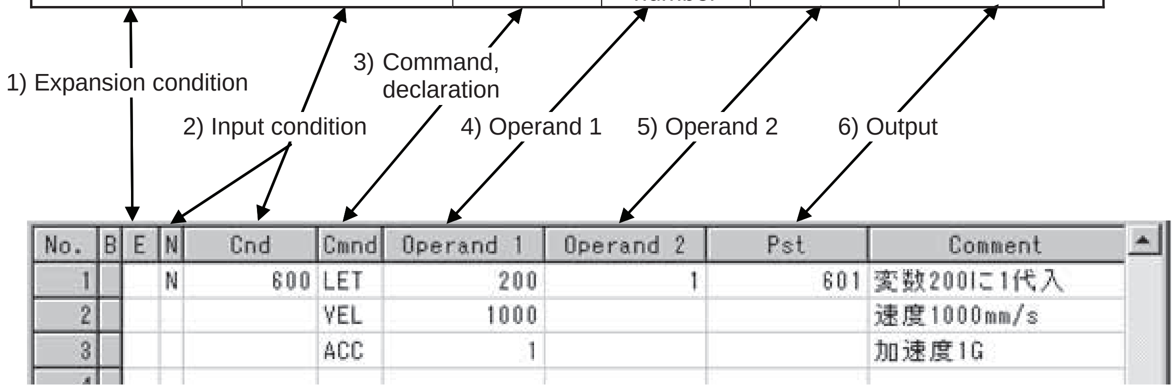
[Function] Assign the value specified in operand 2 to the variable specified in operand 1.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[3] Description of Functions

[Example 1]	LET	1	10	Assign 10 to variable 1.
[Example 2]	LET	1	2	Assign 2 to variable 1.
		3	10	Assign 10 to variable 3.
	LET	*1	*3	Assign the content 10 of variable 3 to the variable that corresponds to the content 2 of variable 1.

[1] SEL language structure

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	LET	Variable number	Data	ZR



Program screen

The details of SEL language structure components are explained.

1) Expansion condition (LD, A, O, AB, OB)

Free ······ You can freely set a desired expansion condition for simulated ladder tasks by selecting LD, A, O, AB and OB. This condition can also be set as an expansion condition for tasks other than simulated ladder tasks.

LD : LOAD
A : AND
O : OR
AB : ANDBLOCK
OB : ORBLOCK

2) Input condition (I/O, flag)
Free ⋯⋯ You can freely set a desired input condition by selecting an input port, output port or flag (global area or local area).

3) Command/declaration
State a command/declaration command*1. The command explained in the applicable section is described.
*1 Once executed in the program, "Actuator Control Declaration" Command (VEL command, VELS command, etc.) will remain effective while the program is running, until the command is changed. If you want to change a value (operand 1, operand 2, etc.) previously set by an "Actuator Control Declaration" Command, you must reset (change) the value at the necessary location in the program.

4) Operand 1,
5) Operand 2
What is set in these items varies depending on the command. Set an appropriate item according to each command.

6) Output (output port, flag)
This is where the result of command execution is shown, and the output 6) turns ON and OFF*2. You can freely set an output port or flag (global area or local area) in which to store the condition of this output. What is turned ON/OFF in output varies depending on the command.
In the 6) Output (output port, flag), the following types are to be shown depending on the operational conditions.
(Output operation types)
CC ⋯⋯ Command successful
ZR ⋯⋯ Calculation result zero
PE ⋯⋯ Operation complete
CP ⋯⋯ Command passing
TU ⋯⋯ Timeout
(CP☐☐ comparison command)
EQ ⋯⋯ Operand 1 = Operand 2
NE ⋯⋯ Operand 1 ≅ Operand 2
GT ⋯⋯ Operand 1 > Operand 2
GE ⋯⋯ Operand 1 ≥ Operand 2
LT ⋯⋯ Operand 1 < Operand 2
LE ⋯⋯ Operand 1 ≤ Operand 2

*2 The output turns OFF when the command is executed. After the command has been executed, the output turns ON depending on the condition specified as the output operation type. (If the condition is not met, the output remains OFF.)
Take note that the output of a CP☐☐ comparison command does not turn OFF when the command is executed.

[2] Applicable models

Controllers that support the command are denoted by a "○".
Controllers that do not support the command are denoted by a "×".
The following controllers are applicable when described as "Applicable for all models".
- XSEL-J/K/JX/KX
- XSEL-P/Q/PX/QX/PCT/QCT
- XSEL-R/S/RX/SX/RXD/SXD
- TT/TTA
- ASEL/PSEL/SSEL
- MSEL-PC/PG/PCX/PGX

[3] Description of functions

Explanation of the function is provided for the corresponding command.

5.2 SEL Language Code Table for each Function

For Operand 1, Operand 2 and the output, the variable indirect specification is available.
For the condition, Operand 1, Operand 2 and the output, an input with symbols is available.
Input into ( ) for Operand 1 and Operand 2 is not compulsory.
“Actuator control declaration” command is kept effective though the program run once it is executed during the program. A reconstruction of the settings is required for the appropriate areas in the program if a change to the values (Operand 1, Operand 2, etc.) already set by “actuator control declaration” command is needed. It means that the values set by the executed command in the last operation are effective.
The output section is turned OFF when the command is executed. After the command execution, it may get turned ON depending on the condition of the output section operation type. (It is turned OFF if the condition does not meet the requirement.)

> ⚠ Caution: Comparative command CP☐☐ (CPEQ, CPNE, CPGT, CPGE, CPLT, CPLE) output section does not get turned OFF during the command execution.

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Variable Assignment	Optional	LET	Variable Assignment	Assignment number	ZR	Assign	231
	Optional	TRAN	Variable to copy data to	Variable to copy data from	ZR	Copy	232
	Optional	CLR	Clear start variable	Clear finish variable	ZR	Clear variable	233
Arithmetic Operation	Optional	ADD	Augend variable	Addend	ZR	Add	234
	Optional	SUB	Minuend variable	Subtrahend	ZR	Subtract	235
	Optional	MULT	Multiplicand variable	Multiplier	ZR	Multiply	236
	Optional	DIV	Dividend variable	Divisor	ZR	Divide	237
	Optional	MOD	Modulus assignment variable	Divisor	ZR	Modulus calculation	238
Function Operation	Optional	SIN	Sine assignment variable	Operant [Radian]	ZR	Sine	239
	Optional	COS	Cosine assignment variable	Operant [Radian]	ZR	Cosine	240
	Optional	TAN	Tangent assignment variable	Operant [Radian]	ZR	Tangent	241
	Optional	ATN	Arc tangent assignment variable	Operant	ZR	Inverse-tangent	242
	Optional	SQR	Square root assignment variable	Operant	ZR	Root	243
Logical Operation	Optional	AND	Logical conjunction variable	Operant	ZR	Logical AND	244
	Optional	OR	Logical disjunction variable	Operant	ZR	Logical OR	245
	Optional	EOR	Logical operation exclusive disjunction	Operant	ZR	Logical exclusive-OR	246
Comparison	Optional	CP☐☐	Comparative variable	Compared number	EQ NE GT GE LT LE	Compare [EQ/NE/GT/GE/LT/LE]	247
Timer	Optional	TIMW	Waiting time [sec]	Prohibited	TU	Wait for certain time	248
	Optional	TIMC	Program No.	Prohibited	CP	Cancel waiting	249
	Optional	GTTM	Time assignment variable	Prohibited	CP	Get time	250
I/O, Flag Operation	Optional	BT☐☐	Start output, Flag	(Complete output, Flag)	CP	Output, flag [ON/OF/NT]	251
	Optional	BTPN	Output port, Flag	Timer setting	CP	Output ON pulse	252
	Optional	BTPF	Output port, Flag	Timer setting	CP	Output OFF pulse	253
	Optional	WT☐☐	I/O, Flag	(Waiting time)	TU	Wait for input and output, flag [ON/OF]	254
	Optional	IN	Head I/O, Flag	Complete input and output, Flag	CC	Input binary number (Max 32 bit)	255
	Optional	INB	Head I/O, Flag	Convertible digits	CC	Input BCD (Max eight digits)	256
	Optional	OUT	Head output, Flag	Complete input and output, Flag	CC	Output binary number (Max 32 bit)	257
	Optional	OTPS	Output port No.	Axis No.	CC	Output current position data	263
	Optional	OUTB	Head output, Flag	Convertible digits	CC	Output BCD (Max eight digits)	258
	Optional	FMIO	Format type	Prohibited	CP	IN (B) OUT (B) command format	259

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Program Control	Optional	GOTO	Tag No. to jump to	Prohibited	CP	Jump	264
	Prohibited	TAG	Declaration tag No.	Prohibited	CP	Declaration of destination to jump to	265
	Optional	EXSR	Execution sub routine No.	Prohibited	CP	Execute subroutine	266
	Prohibited	BGSR	Declaration sub routine No.	Prohibited	CP	Start subroutine	267
	Prohibited	EDSR	Prohibited	Prohibited	CP	End subroutine	268
Task Management	Optional	EXIT	Prohibited	Prohibited	CP	End program	269
	Optional	EXPG	Execution program No.	(Execution program No.)	CC	Start other program	270
	Optional	ABPG	Termination program No.	(Termination program No.)	CC	Abort other program	271
	Optional	SSPG	Pause program No.	(Pause program No.)	CC	Pause program	272
	Optional	RSPG	Resume program No.	(Resume program No.)	CC	Resume program	273
Position Operation	Optional	PGET	Axis No.	Position No.	CC	Assign position to Variable 199	274
	Optional	PPUT	Axis No.	Position No.	CP	Assign Variable 199 value	275
	Optional	PCLR	Start position No.	Termination position No.	CP	Clear position data	276
	Optional	PCPY	Position No. to copy data to	Position No. to copy data from	CP	Copy position data	277
	Optional	PRED	Axis pattern read	Position No. to save data to	CP	Read current axis position	278
	Optional	PRDQ	Axis No.	Variable No.	CP	Read current axis position (single-axis direct)	279
	Optional	PTST	Axis pattern confirmation	Confirmation position No.	CC	Check position data	281
	Optional	PVEL	Speed [mm/sec]	Position No. to assign to	CP	Assign position speed	282
	Optional	PACC	Acceleration [G]	Position No. to assign to	CP	Assign position acceleration	283
	Optional	PDCL	Deceleration [G]	Position No. to assign to	CP	Assign position deceleration	284
	Optional	PAXS	Axis pattern assignment variable No.	Position No.	CP	Read axis pattern	285
	Optional	PSIZ	Size assignment variable No.	Prohibited	CP	Check position data size	286
	Optional	PTAM	Variable No.	Position No.	CP	Substitution of target arm system data	280
	Optional	GTAM	Variable No.	Position No.	CP	Acquirement of target arm system data	287
	Optional	GVEL	Variable No.	Position No.	CP	Get speed data	288
	Optional	GACC	Variable No.	Position No.	CP	Get acceleration data	289
	Optional	GDCL	Variable No.	Position No.	CP	Get deceleration data	290
Actuator Control Declaration	Optional	VEL	Speed [mm/sec]	Prohibited	CP	Set speed	291
	Optional	OVRD	Speed ratio [%]	Prohibited	CP	Speed coefficient settings	293
	Optional	ACC	Acceleration [G]	Prohibited	CP	Set acceleration	294
	Optional	DCL	Deceleration [G]	Prohibited	CP	Set deceleration	296
	Optional	SCRV	Ratio [%]	Prohibited	CP	Set sigmoid motion ratio	298
	Optional	OFST	Setting axis pattern	Offset value [mm]	CP	Set offset	302
	Optional	DEG	Division angle [deg]	Prohibited	CP	Division angle settings	303
	Optional	BASE	Datum axis No.	Prohibited	CP	Datum axis setting	304
	Optional	GRP	Effective axis pattern	Prohibited	CP	Set group axes	305
	Optional	HOLD	(Pause input port)	(HOLD type)	CP	Declare port to pause	306
	Optional	CANC	(Cancel complete input port)	(CANC type)	CP	Declare port to abort	307
	Optional	VLMX	Prohibited	Prohibited	CP	Specify VLMX speed	311
	Optional	ACMX	ACMX Acceleration No.	Prohibited	CP	Indicate ACMX acceleration	308
	Optional	DIS	Distance	Prohibited	CP	Set division distance at spline movement	312
	Optional	POTP	0 or 1	Prohibited	CP	Set PATH output type	313
	Optional	PAPR	Distance	Speed	CP	PUSH Command distance and speed settings	314
	Optional	QRTN	0 or 1	Prohibited	CP	Set quick-return mode	315
	Optional	ACCS	Ratio	Prohibited	CP	Set acceleration ratio	295
	Optional	DCLS	Ratio	Prohibited	CP	Set deceleration ratio	297
	Optional	DFIF	Contact check zone No.	Position No.	CP	Define simple contact check zone coordinate	336
	Optional	DFTL	Tool coordinate system No.	Position No.	CP	Define tool coordinate system	320

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Actuator Control Declaration	Optional	DFWK	Work coordinate system No.	Position No.	CP	Define work coordinate system	325
	Optional	GTIF	Contact check zone No.	Position No.	CP	Get simple contact check zone definition coordinate	340
	Optional	GTTL	Tool coordinate system No.	Position No.	CP	Get tool coordinate system definition data	323
	Optional	GTWK	Work coordinate system No.	Position No.	CP	Get work coordinate system definition number	328
	Optional	NBND	Axis pattern	Close distance	CP	Set close distance	345
	Optional	PTPD	Prohibited	Prohibited	CP	Specify PTP target arm system to current arm	334
	Optional	SLTL	Tool coordinate system No.	Prohibited	CP	Select tool coordinate system	322
	Optional	SEIF	Contact check zone No.	0 to 2	CP	Specify type of simple contact check zone	339
	Optional	RIGH	Prohibited	Prohibited	PE	Change current arm system to right arm	330
	Optional	LEFT	Prohibited	Prohibited	PE	Change current arm system to left arm	331
	Optional	PTPR	Prohibited	Prohibited	CP	Specify PTP target arm system to right arm	332
	Optional	PTPE	Prohibited	Prohibited	CP	Specify PTP target arm system to current arm	335
	Optional	WGHT	Mass	(Inertial moment)	CP	Set tip work mass, inertial moment	341
	Optional	WGT2	Mass	(Inertial moment)	CP	Tip load condition setting 2	343
	Optional	VELS	Ratio	Prohibited	CP	Set speed ratio	292
	Optional	SOIF	Contact check zone No.	Output, global flag No.	CP	Specify output for simple contact check zone	338
	Optional	SLWK	Work coordinate system No.	Prohibited	CP	Select work coordinate system	327
	Optional	PTPL	Prohibited	Prohibited	CP	Specify PTP target arm system to left arm	333

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Actuator Control Command	Optional	SV☐☐	Operation axis pattern	Prohibited	PE	Turn ON/OFF servo	346
	Optional	HOME	Home-return axis pattern	Prohibited	PE	Home return	347
	Optional	MOVP	Position No. to move to	Prohibited	PE	Move by specifying position data	348
	Optional	MOVL	Position No. to move to	Prohibited	PE	Position-indicated interpolation movement	350
	Optional	MVPI	Movement amount position No.	Prohibited	PE	Position-relative movement	352
	Optional	MVLI	Movement amount position No.	Prohibited	PE	Position-relative interpolation movement	354
	Optional	PATH	Start position No.	End position No.	PE	Move along path	358
	Optional	J☐W☐	Drive axis pattern	Start input and output, flag	PE	Jog [FN/FF/BN/BF]	359
	Optional	STOP	Stop axis pattern	Prohibited	CP	Deceleration and stop of axis	361
	Optional	PSPL	Start position No.	End position No.	PE	Move along spline	362
	Optional	PUSH	Target position No.	Prohibited	PE	Move by push motion	363
	Optional	PTRQ	Axis pattern	Ratio [%]	CC	Change push torque limit parameter	365
	Optional	CIR2	Passing position 1 No.	Passing position 2 No.	PE	Circle movement 2 (Arc interpolation)	366
	Optional	ARC2	Passing position No.	End position No.	PE	Arc movement 2 (Arc interpolation)	368
	Optional	CIRS	Passing position 1 No.	Passing position 2 No.	PE	Move along circle three-dimensionally	370
	Optional	ARCS	Passing position No.	End position No.	PE	Move along arc three-dimensionally	372
	Optional	CHVL	Axis pattern	Speed	CP	Change speed	374
	Optional	ARCD	End position No.	Center angle [°(degree)]	PE	Termination position center angle indicated arc movement	376
	Optional	ARCC	Center position No.	Center angle [°(degree)]	PE	Center position center angle indicated arc movement	378
	Optional	PBND	Axis pattern	Distance	CP	Set positioning width	380
	Optional	CIR	Passing position 1 No.	Passing position 2 No.	PE	Circle movement (CIR2 is recommended)	383
	Optional	ARC	Passing position No.	End position No.	PE	Arc movement (ARC2 is recommended)	385
	Optional	PEND	Prohibited	Prohibited	PE	Wait for end of operation by axes currently used by program	387
	Optional	MOVD	Target position	(Axis pattern)	PE	Move via direct value specification	356
	Optional	MVDI	Travel distance	(Axis pattern)	PE	Move relatively via direct value specification	357
	Optional	TMLI	Position No.	Prohibited	PE	Move incrementally to position on tool coordinate system via CP operation	382
	Optional	TMPI	Position No.	Prohibited	PE	Move incrementally to position on tool coordinate system via PTP operation	381
IF structure	Optional	IF☐☐	Comparative variable	Compared No.	CP	Compare [EQ/NE/GT/GE/LT/LE]	388
	Optional	IS☐☐	Column No.	Column No., character literal	CP	Compare strings	389
	Prohibited	ELSE	Prohibited	Prohibited	CP	Declaration of IF Command unsuccessful execution destination	390
	Prohibited	EDIF	Prohibited	Prohibited	CP	IF termination declaration	391
Structural DO	Optional	DW☐☐	Comparative variable	Compared No.	CP	Loop [EQ/NE/GT/GE/LT/LE]	392
	Optional	LEAV	Prohibited	Prohibited	CP	Pull out of DO	393
	Optional	ITER	Prohibited	Prohibited	CP	Repeat of DO	394
	Prohibited	EDDO	Prohibited	Prohibited	CP	DO termination declaration	395
Multi-Branching	Optional	SLCT	Prohibited	Prohibited	CP	Start declaration for multi-branching	396
	Prohibited	WH☐☐	Comparative variable	Compared No.	CP	Branch values [EQ/NE/GT/GE/LT/LE]	397
	Prohibited	WS☐☐	Column No.	Column No., character literal	CP	Branch character line [EQ/NE]	398
	Prohibited	OTHE	Prohibited	Prohibited	CP	Declaration of condition unsuccessful branching destination	399
	Prohibited	EDSL	Prohibited	Prohibited	CP	SLCT termination declaration	400

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
System Information Acquisition	Optional	AXST	Variable No.	Axis No.	CP	Get axis status	401
	Optional	PGST	Variable No.	Program No.	CP	Get program status	402
	Optional	SYST	Variable No.	Prohibited	CP	Get system status	403
	Optional	GARM	Variable No.	Prohibited	CP	Get current arm system	404
Zone	Optional	WZNA	Zone No.	Axis pattern	CP	Wait for zone ON based on AND gate	405
	Optional	WZNO	Zone No.	Axis pattern	CP	Wait for zone ON based on OR gate	406
	Optional	WZFA	Zone No.	Axis pattern	CP	Wait for zone OFF based on AND gate	407
	Optional	WZFO	Zone No.	Axis pattern	CP	Wait for zone OFF based on OR gate	408
Communication	Optional	OPEN	Channel No.	Prohibited	CP	Open channel	409
	Optional	CLOS	Channel No.	Prohibited	CP	Close channel	410
	Optional	READ	Channel No.	Column No.	CC	Input from channel	411
	Optional	TMRW	Read timer setting	(Write timer setting)	CP	Set READ timeout value	415
	Optional	WRIT	Channel No.	Column No.	CC	Output to channel	416
	Optional	SCHA	Character code	Prohibited	CP	Character setting for sending and receiving	417
	Optional	TMRD	Timer period	Prohibited	CP	Set READ timeout value	413
String Operation	Optional	SCPY	Column No.	Column No., character literal	CC	Copy character string	418
	Optional	SCMP	Column No.	Column No., character literal	EQ	Compare character strings	419
	Optional	SGET	Variable No.	Column No., character literal	CP	Get character	420
	Optional	SPUT	Column No.	Data	CP	Set character	421
	Optional	STR	Column No.	Data	CC	Convert character string; decimal	422
	Optional	STRH	Column No.	Data	CC	Convert character string; hexadecimal	423
	Optional	VAL	Variable No.	Column No., character literal	CC	Convert character string data; decimal	424
	Optional	VALH	Variable No.	Column No., character literal	CC	Convert character string data; hexadecimal	425
	Optional	SLEN	Character string length	Prohibited	CP	Set length	426

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Palletizing Definition	Optional	BGPA	Palletizing No.	Prohibited	CP	Declare start of palletizing setting	433
	Prohibited	EDPA	Prohibited	Prohibited	CP	Declare end of palletizing setting	434
	Optional	PAPI	Count	Count	CP	Set palletizing counts	435
	Optional	PAPN	Pattern No.	Prohibited	CP	Set palletizing pattern	436
	Optional	PASE	Axis No.	Axis No.	CP	Declare palletizing axes	437
	Optional	PAPT	Pitch	Pitch	CP	Set palletizing pitches	438
	Optional	PAST	(Position No.)	Prohibited	CP	Set palletizing reference point	439
	Optional	PAPS	Position No.	Palletizing position setting type	CP	Set palletizing points for 3-point or 4-point teaching	440
	Optional	PSLI	Offset amount	(Count)	CP	Set zigzag	443
	Optional	PCHZ	(Axis No.)	Prohibited	CP	Declare palletizing Z-axis	444
	Optional	PTRG	Position No.	Position No.	CP	Set palletizing arch triggers	445
	Optional	PEXT	(Position No.)	Prohibited	CP	Set composite palletizing	446
	Optional	OFPZ	Offset amount	Prohibited	CP	Set palletizing Z-axis offset	447
	Optional	ACHZ	Axis No.	Prohibited	CP	Declare arch-motion Z-axis	429
	Optional	ATRG	Position No.	Position No.	CP	Set arch triggers	430
	Optional	AEXT	(Position No.)	Prohibited	CP	Set composite arch motion	431
	Optional	OFAZ	Offset amount	Prohibited	CP	Set arch-motion Z-axis offset	432
	Optional	PTNG	Palletizing No.	Variable No.	CP	Get palletizing position number	448
	Optional	PINC	Palletizing No.	Prohibited	CC	Increment palletizing position number by 1	449
	Optional	PDEC	Palletizing No.	Prohibited	CC	Decrement palletizing position number by 1	450
	Optional	PSET	Palletizing No.	Data	CC	Set palletizing position number directly	451
	Optional	PARG	Palletizing No.	Axis No.	CP	Get palletizing angle	452
	Optional	PAPG	Palletizing No.	Position No.	CP	Get palletizing calculation data	453
	Optional	PMVP	Palletizing No.	(Position No.)	PE	Move to palletizing points via PTP	454
	Optional	PMVL	Palletizing No.	(Position No.)	PE	Move to palletizing points via interpolation	455
	Optional	PACH	Palletizing No.	Position No.	PE	Arch motion to palletizing point	456
	Optional	ARCH	Position No.	Position No.	PE	Arch motion	427
Building of Pseudo-Ladder Task	Optional	CHPR	0 or 1	Prohibited	CP	Change task level	458
	Prohibited	TPCD	0 or 1	Prohibited	CP	Specify processing to be performed when input condition is not specified	459
	Prohibited	TSLP	Time	Prohibited	CP	Task sleep	460
	Optional	OUTR	Output, flag No.	Prohibited	CP	Ladder output relay	152
	Optional	TIMR	Local flag No.	Timer setting	CP	Ladder timer relay	152
Extended Command	Optional	ECMD	1	Axis No.	CC	Get motor current value	461
	Optional	ECMD	2	Axis No.	CC	Get home sensor status	462
	Optional	ECMD	3	Axis No.	CC	Get overrun sensor status	463
	Optional	ECMD	4	Axis No.	CC	Get creep sensor status	464
	Optional	ECMD	5	Axis No.	CC	Get axis operation status	465
	Optional	ECMD	6	Axis No.	CC	Current position acquirement on each axis system	466
	Optional	ECMD	20	Axis No.	CC	Get parameter value	467
	Optional	ECMD	250	Axis pattern	CC	Set torque limit/torque limit over detection time	469
Vision System I/F Related	Optional	SLVS	Select using Vision System I/F	(Timeout time)	CC	Select Vision System I/F	545
	Optional	GTVD	Capturing Trigger Classification	Variable No.	CC	Vision System I/F Image-Capture Data Acquirement	547
Conveyor Tracking Related	Optional	TRMD	Select using Tracking Mode	TRAC Command timeout time	CC	Tracking Mode Setting	541
	Optional	TRAC	0 or 1	Position No. to save the work position information	CC	Tracking Operation Setting & Datum Point Position Information Obtainment in Work	542
Anti-Vibration Control Related	Optional	NTCH	Axis pattern	Parameter set number	CC	Anti-Vibration Control Parameter Set Select	549

RC Gateway Function Commands (Controller with Gateway Function Only)

* Refer to "XSEL Controller P/Q/PX/QX RC Gateway Function Instruction Manual" for the commands related to RC gateway functions.

Output operation types
CC: Command successful, ZR: Calculation result zero
PE: Operation complete, CP: Command passing, TU: Timeout
EQ: Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT: Operand 1 > Operand 2, GE: Operand 1 ≧ Operand 2
LT: Operand 1 < Operand 2, LE: Operand 1 ≦ Operand 2

Category	Condition	Command	RC position-data use mode		Operand 1	Operand 2	Output	Function	Page
			XSEL	RC					
RC axis position operation	Optional	RPGT	○	×	RC-axis No.	Position No.	CC	Assign RC axis position location to Variable 199	472
	Optional	RPPT	○	×	RC-axis No.	Position No.	CP	Assign Variable 199 to RC axis position location	473
	Optional	RPCR	○	×	RC-axis No.	Variable No.	CP	Clear RC-axis position data	474
	Optional	RPCP	○	×	RC-axis No.	Variable No.	CP	Copy RC-axis position data	475
	Optional	RPRD	○	×	Position No.	Prohibited	CP	Read current RC-axis position	476
	Optional	RPRQ	○	○	RC-axis No.	Variable No.	CP	Read current RC-axis position (single-axis direct)	477
	Optional	RPVL	○	×	RC-axis No.	Position No.	CP	Assign Variable 199 to RC axis position speed	478
	Optional	RPAD	○	×	RC-axis No.	Position No.	CP	Assign Variable 199 to RC axis position acceleration/deceleration	479
	Optional	RPIP	○	×	RC-axis No.	Position No.	CP	Assign Variable 199 to RC axis position positioning width	480
	Optional	RPTQ	○	×	RC-axis No.	Position No.	CP	Assign Variable 199 to RC axis position current limitation	481
	Optional	RGVL	○	×	RC-axis No.	Position No.	CP	Assign RC axis position speed to Variable 199	482
	Optional	RGAD	○	×	RC-axis No.	Position No.	CP	Assign RC axis position acceleration/deceleration to Variable 199	483
	Optional	RGIP	○	×	RC-axis No.	Position No.	CP	Assign RC position positioning width to Variable 199	484
	Optional	RGTQ	○	×	RC-axis No.	Position No.	CP	Assign RC position current limitation to Variable 199	485
RC actuator control command	Optional	RAXS	○	○	Axis pattern, upper	Axis pattern, lower	CP	Set axis pattern for RC axis	486
	Optional	RSON	○	○	Prohibited	Prohibited	PE	Turn ON RC-axis servo	487
	Optional	RSOF	○	○	Prohibited	Prohibited	PE	Turn OFF RC-axis servo	488
	Optional	RHOM	○	○	Prohibited	Prohibited	PE	Return RC-axis to home	489
	Optional	RMVP	○	○	Position No.	Prohibited	PE	Move RC-axis by position specification	490
	Optional	RMPI	○	×	Position No.	Prohibited	PE	Move RC-axis incrementally by position specification	491
	Optional	RMVD	○	×	RC-axis No.	Variable No.	PE	Move RC axis with direct specification	492
	Optional	RMDI	○	×	RC-axis No.	Variable No.	PE	Move RC axis to directly specified relative position	493
	Optional	RPUS	○	×	RC-axis No.	Position No.	PE	Move RC-axis via push motion	494
	Optional	RSTP	○	○	Prohibited	Prohibited	PE	Decelerate and stop RC axis	495
RC axis information acquisition	Optional	RCST	○	○	Variable No.	RC-axis No.	PE	Acquire RC axis status	496

Electronic Cam Control System Related Commands (Controller with Electronic Cam Function Only)

* Refer to “XSEL Controller P/Q/PCT/QCT Electronic Cam function Instruction Manual” for the details of the commands related to the electronic cam functions.

Output operation types
CC: Command successful, CP: Command passing
PE: Operation complete

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Extension Motion Control Board Input Operations	Optional	XCRP	Pulse input channel No.	Prohibited	CP	Clear input counter record for extension motion control board	498
	Optional	XGTP	Pulse input channel No.	Prohibited	CP	Acquire current record of extension motion control board input counter	499
Extension Motion Control Board Axis Position Operations	Optional	XPGT	Axis No.	Position No.	CC	Read extension motion control board axis position data	500
	Optional	XPPT	Axis No.	Position No.	CP	Write extension motion control board axis position data	501
	Optional	XPCR	Axis No.	Variable No.	CP	Erase extension motion control board axis position data	502
	Optional	XPCP	Axis No.	Variable No.	CP	Copy extension motion control board axis position data	503
	Optional	XPRD	Position No.	Prohibited	CP	Read extension motion control board axis current command position	504
	Optional	XPRQ	Axis No.	Variable No.	CP	Read extension motion control board axis current command position (single-axis direct)	505
	Optional	XPVL	Axis No.	Position No.	CP	Write extension motion control board axis speed data	506
	Optional	XPAC	Axis No.	Position No.	CP	Write extension motion control board axis acceleration data	507
	Optional	XPDC	Axis No.	Position No.	CP	Write extension motion control board axis deceleration data	508
	Optional	XPIP	Axis No.	Position No.	CP	Write extension motion control board axis positioning complete width data	509
	Optional	XGVL	Axis No.	Position No.	CP	Read extension motion control board axis speed data	510
	Optional	XGAC	Axis No.	Position No.	CP	Read extension motion control board axis acceleration data	511
	Optional	XGDC	Axis No.	Position No.	CP	Read extension motion control board axis deceleration data	512
	Optional	XGIP	Axis No.	Position No.	CP	Read extension motion control board axis positioning complete width data	513
Extension Motion Control Board Axis Actuator Control Declarations	Optional	XAXS	Axis pattern, upper	Axis pattern, lower	CP	Set each pulse I/O board axis pattern (0 to 15 axis)	514
Extension Motion Control Board Axis Actuator Control Commands	Optional	XSON	Prohibited	Prohibited	CP	Extension motion control board axis to servo ON	515
	Optional	XSOF	Prohibited	Prohibited	CP	Extension motion control board axis to servo OFF	516
	Optional	XHOM	Prohibited	Prohibited	PE	Extension motion control board axis to home return	517
	Optional	XMVP	Position No.	Prohibited	PE	Move extension motion control board axis to indicated position	518
	Optional	XMPI	Position No.	Prohibited	PE	Perform extension motion control board axis position relative movement	519
	Optional	XMVL	Position No.	Prohibited	PE	Move extension motion control board axis for position indicated interpolation	520
	Optional	XMLI	Position No.	Prohibited	PE	Move extension motion control board axis for position relative interpolation	521
	Optional	XMVD	Axis No.	Variable No.	PE	Move extension motion control board axis to directly indicated absolute position	522
	Optional	XMDI	Axis No.	Variable No.	PE	Move extension motion control board axis to directly indicated relative position	523
	Optional	XJ☐☐	Input, output, flag No.	Prohibited	PE	Perform extension motion control board axis jog operation [FN/FF/BN/BF]	524
	Optional	XPED	Prohibited	Prohibited	PE	Waiting for extension motion control board axis to finish positioning operation of axis used by self-program	525

Output operation types
CC: Command successful, CP: Command passing
PE: Operation complete

Category	Condition	Command	Operand 1	Operand 2	Output	Function	Page
Extension Motion Control Board Axis Actuator Control Commands	Optional	XSTP	Prohibited	Prohibited	PE	Cancel operation of extension motion control board axis	526
	Optional	XWIP	Prohibited	Prohibited	CP	Waiting for extension motion control board axis positioning complete signal to be turned ON	527
	Optional	XCAS	Slave Shaft No.	Variable No.	PE	Start synchronizing extension motion control board axis electronic cam (indicating main axis)	529
	Optional	XCTM	Slave Shaft No.	Variable No.	PE	Move extension motion control board axis individual electronic cam (indicating time)	534
	Optional	XSFS	Slave Shaft No.	Variable No.	PE	Start synchronizing of extension motion control board axis electronic shaft	536
	Optional	XSYE	Slave Shaft No.	(Complete Type)	PE	Cancel operation of extension motion control board axis	538
Extension Motion Control Board Axis Status Acquirement	Optional	XAST	Variable No.	Axis No.	CP	Acquire extension motion control board axis status	540

5.3 Explanation of Commands

[1] Variable Assignment

● LET (Assign)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	LET	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the value specified in operand 2 to the variable specified in operand 1. The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]

LET	1	10	Assign 10 to variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	3	10	Assign 10 to variable 3.
LET	*1	*3	Assign the content 10 of variable 3 to the variable that corresponds to the content 2 of variable 1.

● TRAN (Copy)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TRAN	Variable number	Variable number	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the content of the variable specified in operand 2 to the variable specified in operand 1.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]	TRAN	1	2	Assign the content of variable 2 to variable 1. The above operation can be performed with a LET command as follows.
	LET	1	*2	
[Example 2]	LET	1	2	Assign 2 to variable 1.
	LET	2	3	Assign 3 to variable 2.
	LET	3	4	Assign 4 to variable 3.
	LET	4	10	Assign 10 to variable 4.
	TRAN	*1	*3	Assign 10 of the content 4 of variable 3 to the variable that corresponds to the content 2 of variable 1.

The variables change as follows.

1	2	3	4
2	3	4	10

→

1	2	3	4
2	10	4	10

● CLR (Clear variable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CLR	Variable number	Variable number	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Clear the variables from the one specified in operand 1 through the other specified in operand 2.
The contents of the variables that have been cleared become 0.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]	CLR	1	5	Clear variables 1 through 5.
[Example 2]	LET	1	10	Assign 10 to variable 1.
	LET	2	20	Assign 20 to variable 2.
	CLR	*1	*2	Clear the variables from the contents 10 in variable 1 through the contents 20 in variable 2.

[2] Arithmetic Operation

● ADD (Add)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ADD	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Add the content of the variable specified in operand 1 and the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	3	Assign 3 to variable 1.
ADD	1	2	Add 2 to the content of variable 1 (3). 5 (3 + 2 = 5) will be stored in variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	2	Assign 3 to variable 2.
LET	3	2	Assign 2 to variable 3.
ADD	*1	*3	Add the content of variable 3, or 2, to the variable that corresponds to the content of variable 1, or 2. 3 + 2, or 5, is stored in variable 2.

● SUB (Subtract)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SUB	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Subtract the value specified in operand 2 from the content of the variable specified in operand 1, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	3	Assign 3 to variable 1.
SUB	1	2	Subtract 2 from the content of variable 1 (3). 1 (3 – 2 = 1) will be stored in variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
LET	3	2	Assign 2 to variable 3.
SUB	*1	*3	Subtract the content of variable 3 (2), from the variable corresponding to the content of variable 1 (2). 1 (3 - 2 = 1) will be send in variable 2.

● MULT (Multiply)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MULT	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Multiply the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	3	Assign 3 to variable 1.
MULT	1	2	Multiply the content of variable 1 (3) by 2. 3 × 2, or 6, is stored in variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
LET	3	2	Assign 2 to variable 3.
MULT	*1	*3	Multiply the variable that corresponds to the content of variable 1, or 2, by the content of variable 3, or 2. 3 × 2, or 6, is stored in variable 2.

● DIV (Divide)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DIV	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Divide the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

(Note) If the variable specified in operand 1 is an integer variable, any decimal places will be rounded off.

[Example 1]

```
LET   1    6     Assign 6 to variable 1.
DIV   1    2     Divide the content of variable 1 (6) by 2.
                 6 / 2, or 3, is stored in variable 1.
```

[Example 2]

```
LET   2    2     Assign 2 to variable 1.
LET   2    6     Assign 6 to variable 2.
LET   3    2     Assign 2 to variable 3.
DIV   *1   *3    Divide the variable that corresponds to the content
                 of variable 1, or 2, by the content of variable 3, or 2.
                 6 / 2, or 3, is stored in variable 2.
```

● MOD (Remainder)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MOD	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign, to the variable specified in 1, the remainder obtained by dividing the content of the variable specified in operand 1 by the value specified in operand 2. The output will turn ON when the operation result becomes 0.

(Note) A MOD command is used with integer variables.

[Example 1]

LET	1	7	Assign 7 to variable 1.
MOD	1	3	Obtain the remainder of dividing the content of variable 1 (7) by 3. The remainder of 7 / 3 = 2, or 1, is assigned to variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	7	Assign 7 to variable 2.
LET	3	3	Assign 3 to variable 3.
MOD	*1	*3	Obtain the remainder of dividing the variable that corresponds to the content of variable 1, or 2, by the content of variable 3, or 3. The remainder of 7 / 3 = 2, or 1, is assigned to variable 2.

[3] Function Operation

● SIN (Sine operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SIN	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the sine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle $\times \pi$ / 180

[Example 1]	SIN	100	0.523599	Assign the sine of 0.523599 (0.5) to variable 100.
[Example 2]	LET	1	100	Assign 100 to variable 1.
	LET	101	30	30 $\times \pi$ / 180 (radian)
	MULT	101	3.141592	(30° is converted to radian and the result is
	DIV	101	180	assigned to variable 101.)
	SIN	*1	*101	Assign the sine of the content of variable 101, or 0.5, to the variable that corresponds to the content of variable 1, or 100.

● COS (Cosine operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	COS	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the cosine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle × π / 180

[Example 1]

COS	100	1.047197	Assign the cosine of 1.047197 (0.5) to variable 100.

[Example 2]

LET	1	100	Assign 100 to variable 1.
LET	101	60	60 × π / 180 (radian)
MULT	101	3.141592	(60° is converted to radian and the result is
DIV	101	180	assigned to variable 101.)
COS	*1	*101	Assign the sine of the content of variable 101, or 0.5, to the variable that corresponds to the content of variable 1, or 100.

● TAN (Tangent operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TAN	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle × π / 180

[Example 1]

TAN	100	0.785398	Assign the tangent of 0.785398 (1) to variable 100.

[Example 2]

LET	1	100	Assign 100 to variable 1.
LET	101	45	45 × π / 180 (radian)
MULT	101	3.141592	(45° is converted to radian and the result is
DIV	101	180	assigned to variable 101.)
TAN	*1	*101	Assign the sine of the content of variable 101, or 1, to the variable that corresponds to the content of variable 1, or 100.

● ATN (Inverse-tangent operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ATN	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the inverse tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of inverse tangent is radian.

(Note 1) Radian = Angle × π / 180

[Example 1]	ATN	100	1	Assign the inverse tangent of 1 (0.785398) to variable 100.
[Example 2]	LET	1	100	Assign 100 to variable 1.
	LET	101	1	Assign 1 to variable 101.
	ATN	*1	*101	Assign the inverse arc tangent of the content of variable 101, or 0.785398, to the variable that corresponds to the content of variable 1, or 100.

● SQR (Root operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SQR	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the root of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

SQR	1	4	Assign the root of 4 (2) to variable 1.

[Example 2]

LET	1	10	Assign 10 to variable 1.
LET	2	4	Assign 4 to variable 2.
SQR	*1	*2	Assign the square root of the content of variable 2, or 4, to the variable that corresponds to the content of variable 1, or 10.

[4] Logical Operation

● AND (Logical AND)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	AND	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the logical AND operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	204	Assign 204 to variable 1.
AND	1	170	Assign the logical AND operation result (136) of the content of variable 1 (204) and 170, to variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	204	Assign 204 to variable 2.
LET	3	170	Assign 170 to variable 3.
AND	*1	*3	Assign the logical product 136 of the content 204 of the variable that corresponds to the content of variable 1, or 2, and the content of variable 3, or 170, to the variable that corresponds to the content of variable 1, or 2.

Decimal	Binary
204	11001100
AND 170	AND 10101010
136	10001000

● OR (Logical OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OR	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the logical OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	204	Assign 204 to variable 1.
OR	1	170	Assign the logical OR operation result (238) of the content of variable 1 (204) and 170, to variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	204	Assign 204 to variable 2.
LET	3	170	Assign 170 to variable 3.
OR	*1	*3	Assign the logical sum 238 of the content 204 of the variable that corresponds to the content of variable 1, or 2, and the content of variable 3, or 170, to the variable that corresponds to the content of variable 1, or 2.

Decimal		Binary	
	204		11001100
OR	170	OR	10101010
	238		11101110

● EOR (Logical exclusive-OR)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	EOR	Variable number	Data	ZR

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign the logical exclusive-OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

LET	1	204	Assign 204 to variable 1.
EOR	1	170	Assign the logical exclusive-OR operation result (102) of the content of variable 1 (204) and 170, to variable 1.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	204	Assign 204 to variable 2.
LET	3	170	Assign 170 to variable 3.
EOR	*1	*3	Assign the exclusive logical sum 102 of the content 204 of the variable that corresponds to the content of variable 1, or 2, and the content of variable 3, or 170, to the variable that corresponds to the content of variable 1, or 2.

```
   Decimal            Binary

       204            11001100
   EOR 170       EOR  10101010
   -------       -------------
       102            01100110
```

[5] Comparison Operation

● CP☐☐ (Compare)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CP☐☐	Variable number	Data	EQ NE GT GE LT LE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] The output will be turned ON if the comparison result of the content of the variable specified in operand 1 and the value specified in operand 2 satisfies the condition.
The value in the variable does not change.
The output will be turned OFF if the condition is not satisfied.

(Note) The output will not be turned OFF when the command is executed.

```
CP☐☐
  ├─ EQ ............... Operand 1  =  Operand 2
  ├─ NE ............... Operand 1  ≠  Operand 2
  ├─ GT ............... Operand 1  >  Operand 2
  ├─ GE ............... Operand 1  ≥  Operand 2
  ├─ LT ............... Operand 1  <  Operand 2
  └─ LE ............... Operand 1  ≤  Operand 2
```

[Example 1]

	LET	1	10		Assign 10 to variable 1.
	CPEQ	1	10	600	Turn ON flag 600 if the content of variable 1 is 10.
600	ADD	2	1		Add 1 to variable 2 if flag 600 is ON.

[Example 2]

	LET	1	2		Assign 2 to variable 1.
	LET	2	10		Assign 10 to variable 2.
	LET	3	10		Assign 10 to variable 3.
	CPNE	*1	*3	310	Turn ON output 310 if the variable that corresponds to the content of variable 1, or 2, is not equal to the content of variable 3.

[6] Timer

● TIMW (Timer)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TIMW	Time	Prohibited	TU

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Stop the program and wait for the time specified in operand 1.
The setting range is 0.01 to 99, and the unit is second.
The output will turn ON when the specified time has elapsed and the program proceeds to the next step.

[Example 1] TIMW 1.5 Wait for 1.5sec.

[Example 2]

```
LET   1    10     Assign 10 to variable 1.
TIMW  *1          Wait for the content of variable 1 (10sec).
```

● TIMC (Cancel timer)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TIMC	Program number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Cancel a timer in other program running in parallel.

(Note) Timers in TIMW, WTON, WTOF and READ commands can be cancelled. In the case of WTON, WTOF and READ commands, even if timeout is not specified it is assumed that an unlimited timer has been specified and the wait time will be cancelled.

[Example 1]

TIMC	10		Cancel the wait time in program 10.

[Example 2]

LET	1	10	Assign 10 to variable 1.
TIMC	*1		Cancel the wait time in the content of variable 1 (program 10).

[Example 3]

Program 1		Program 10	
:		:	
:		WTON 8 20	Program 10 waits for input 8 for 20 seconds.
:		(Wait for input 8)	
TIMC	10	(Wait for input 8)	Cancel the wait time in program 10.
:		:	

(Note) The steps shown in the above example represent those executed simultaneously in different programs.

● GTTM (Get time)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTTM	Variable number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read system time to the variable specified in operand 1. The time is specified in units of 10msec.
The time obtained here has no base number. Therefore, this command is called twice and the difference will be used to calculate the elapsed time.

[Example 1]

GTTM	1		Read the reference time to variable 1.
ADD	1	500	Set the ending time to 5sec later.
GTTM	2		Read the current system time to variable 2.
DWLE	2	*1	Proceed to the step next to EDDO when 5sec elapsed.
:			
:			The above process will be repeated for 5sec.
GTTM	2		Read the current system time to variable 2.
EDDO			

[Example 2]

LET	1	5	Assign 5 to variable 1.
GTTM	*1		Store the current system time in the content of variable 1 (variable 5).

[7] I/O, Flag Operation

● BT□□ (Output port, flag operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	BT□□	Output, flag	(Output, flag)	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Reverse the ON/OFF status of the output ports or flags from the one specified in operand 1 through the other specified in operand 2.

(Note) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operands 1 and 2.

BT□□
- ON .... Switch the status to ON.
- OF .... Switch the status to OFF.
- NT .... Reverse the status.

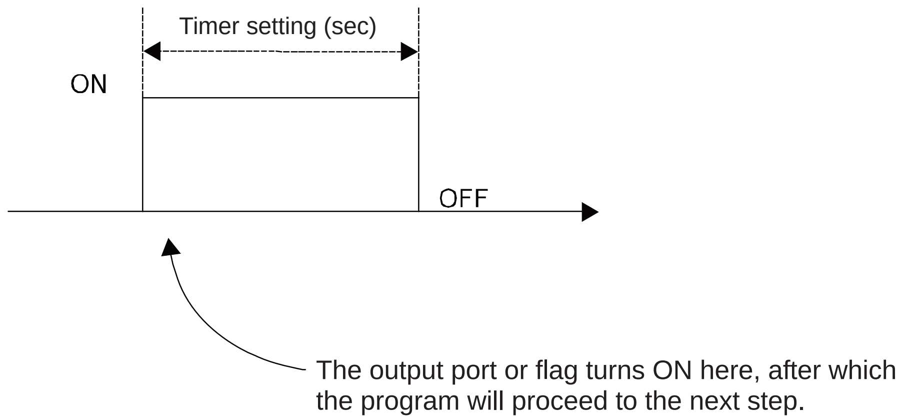
[Example 1]	BTON	300		Turn ON output port 300.
[Example 2]	BTOF	300	307	Turn OFF output ports 300 through 307.
[Example 3]	LET	1	600	Assign 600 to variable 1.
	BTNT	*1		Reverse the content of variable 1 (flag 600).
[Example 4]	LET	1	600	Assign 600 to variable 1.
	LET	2	607	Assign 607 to variable 2.
	BTON	*1	*2	Turn ON the flags from the content of variable 1 (flag 600) through the content of variable 2 (flag 607).

● BTPN (Output ON pulse)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	BTPN	Output port, flag	Timer setting	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Turn ON the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned ON and then the program will proceed to the next step. The output port or flag will be turned OFF automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00sec (including up to two decimal places).



(Note 1) If this command is executed with respect to an output port or flag already ON, the output port or flag will be turned OFF upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned OFF.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program.
(There is no limitation as to how many times these timers can be used in a single program.)

(Note 5) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operand 1.

(Note 6) If other task interrupts after a port is turned ON until it is subsequently turned OFF, an error will generate in pulse output time, in which case pulse output cannot be used for a specified period.

[Example]

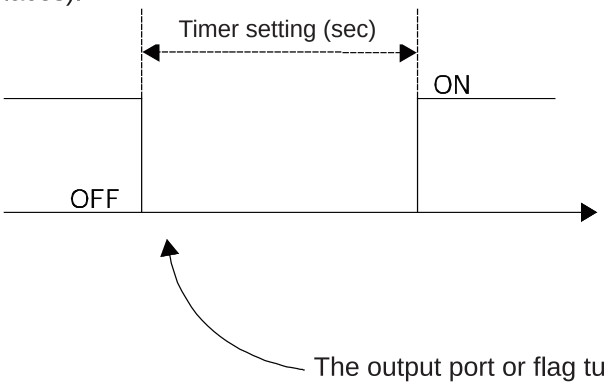
BTPN	300	1	Turn ON output port 300 for 1sec.
BTPN	600	10	Turn ON flag 600 for 10sec.

● BTPF (Output OFF pulse)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	BTPF	Output port, flag	Timer setting	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Turn OFF the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned OFF and then the program will proceed to the next step. The output port or flag will be turned ON automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00sec (including up to two decimal places).



The output port or flag turns OFF here, after which the program will proceed to the next step.

(Note 1) If this command is executed with respect to an output port or flag already OFF, the output port or flag will be turned ON upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned ON.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program.
(There is no limitation as to how many times these timers can be used in a single program.)

(Note 5) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operand 1.

(Note 6) If other task interrupts after a port is turned ON until it is subsequently turned OFF, an error will generate in pulse output time, in which case pulse output cannot be used for a specified period.

[Example]

BTPF	300	1	Turn OFF output port 300 for 1sec.
BTPF	600	10	Turn OFF flag 600 for 10sec.

● WT□□ (Wait for I/O port, flag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WT□□	I/O, flag	(Time)	TU

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Wait for the I/O port or flag specified in operand 1 to turn ON/OFF.
The program can be aborted after the specified time by setting the time in operand 2.
The setting range is 0.01 to 99sec.
The output will turn ON upon elapse of the specified time (only when operand 2 is specified).

(Note) A local flag cannot be entered in operand 1.

```
WT□□
   ├ON ··············· Wait for the applicable I/O port or flag to turn ON.
   └OF ··············· Wait for the applicable I/O port or flag to turn OFF.
```

[Example 1]	WTON	15		Wait for input port 15 to turn ON.
[Example 2]	WTOF	308	10	Wait for 10sec for output port 308 to turn OFF.
[Example 3]	LET	1	600	Assign 600 to variable 1.
	WTON	*1		Wait for the content of variable 1 (flag 600) to turn ON.
[Example 4]	LET	1	8	Assign 8 to variable 1.
	LET	2	5	Assign 5 to variable 2.
	WTOF	*1	*2	Wait for the content of variable 2 (5sec) for the content of variable 1 (input port 8) to turn OFF.

● IN (Read I/O, flag as binary)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	IN	I/O, flag	I/O, flag	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read the I/O ports or flags from the one specified in operand 1 through the other specified in operand 2, to variable 99 as a binary.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	• • • Binary
15	14	13	12	11	10	9	8	• • • Input port number
ON	OFF	OFF	OFF	OFF	ON	OFF	ON	

1	0	0	0	0	1	0	1	• • • Binary

$2^7 + 0 + 0 + 0 + 0 + 2^2 + 0 + 2^0$

$128 + 0 + 0 + 0 + 0 + 4 + 0 + 1 \quad = \quad 133$

$\boxed{133}$ • • • • • • • • • • • • • • • • • Variable 99

(Note 1) A maximum of 32 bits can be input.

(Note 2) When 32 bits have been input and the most significant bit is ON, the value read to variable 99 will be treated as a negative value.

(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

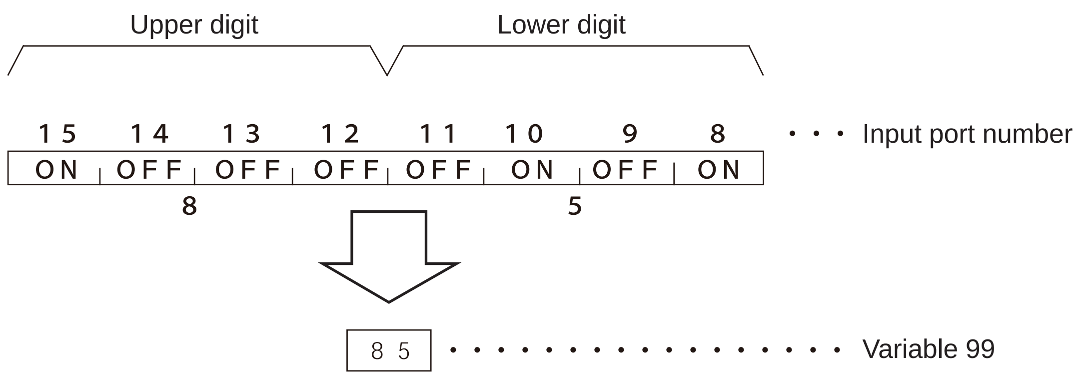
[Example 1]	IN	8	15	Read input ports 8 through 15, to variable 99 as a binary.
[Example 2]	LET	1	8	Assign 8 to variable 1.
	LET	2	15	Assign 15 to variable 2.
	IN	*1	*2	Read the input ports from the content of variable 1 (input port 8) through the content of variable 2 (input port 15), to variable 99 as a binary.

● INB (Read I/O, flag as BCD)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	INB	I/O, flag	BCD digits	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read the I/O ports or flags from the one specified in operand 1 for the number of digits specified in operand 2, to variable 99 as a BCD.



(Note 1) A maximum of eight digits (32 bits) can be input.

(Note 2) The number of I/O ports and flags that can be used is $4 \times n$ (digits).

(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

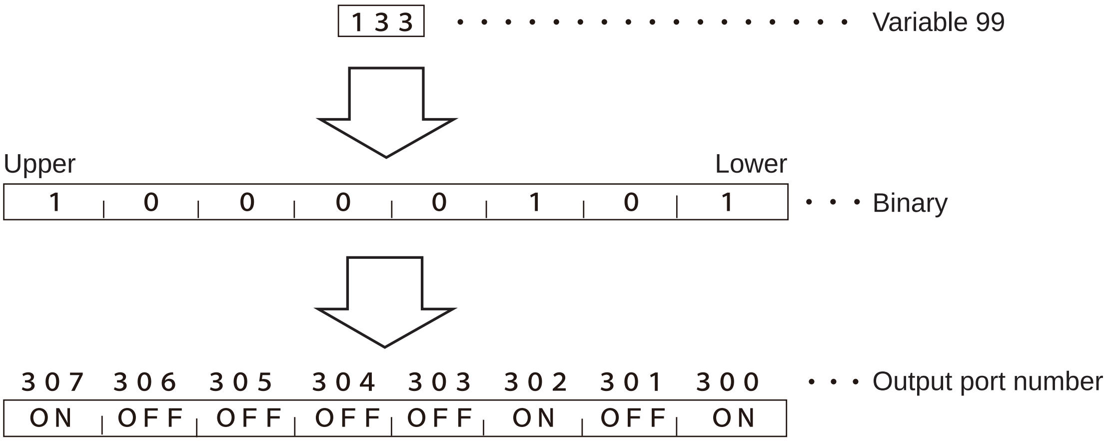
[Example 1]	INB	8	2	Read input ports 8 through 15, to variable 99 as a BCD.
[Example 2]	LET	1	8	Assign 8 to variable 1.
	LET	2	2	Assign 2 to variable 2.
	INB	*1	*2	Read the input ports from the content of variable 1 (input port 8) for the content of variable 2 (two digits) (until input port 15), to variable 99 as a BCD.

● OUT (Write output, flag as binary)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OUT	Output, flag	Output, flag	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 through the other specified in operand 2.



(Note 1) A maximum of 32 bits can be output.

(Note 2) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1] OUT 300 307 Write the value in variable 99 to output ports 300 through 307 as a binary.

[Example 2]

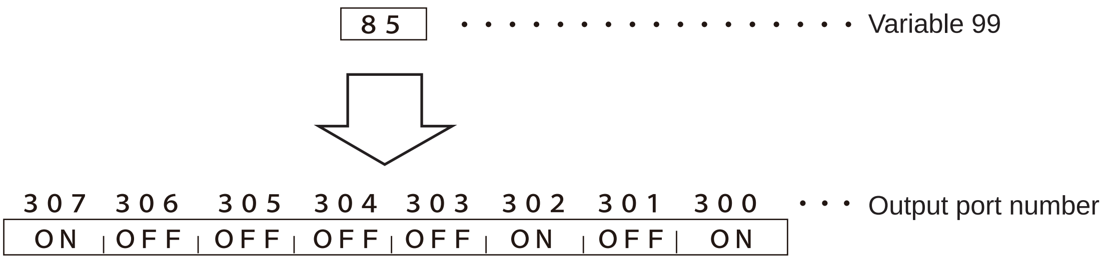
LET	1	300	Assign 300 to variable 1.
LET	2	307	Assign 307 to variable 2.
OUT	*1	*2	Write the value in variable 99 to the output ports from the content of variable 1 (output port 300) through the content of variable 2 (output port 307) as a binary.

● OUTB (Write output, flag as BCD)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OUTB	Output, flag	BCD digits	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 for the number of digits specified in operand 2 as a BCD.



(Note 1) A maximum of eight digits (32 bits) can be output.

(Note 2) The number of output ports and flags that can be used is 4 × n (digits).

(Note 3) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]	OUTB	300	2	Write the value in variable 99 to the output ports from 300 for two digits (until output port 307) as a BCD.
[Example 2]	LET	1	300	Assign 300 to variable 1.
	LET	2	2	Assign 2 to variable 2.
	OUTB	*1	*2	Write the value in variable 99 to the output ports from the content of variable 1 (output port 300) for the content of variable 2 (two digits) (until output port 307) as a BCD.

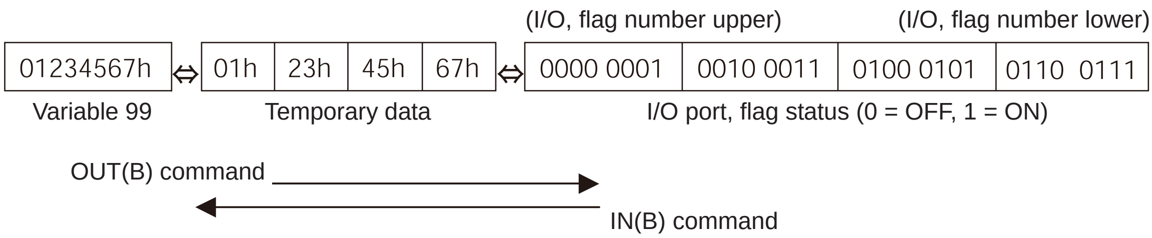
● FMIO (Set IN, INB, OUT, OUTB, OTPS command format)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	FMIO	Format type	Prohibited	CP

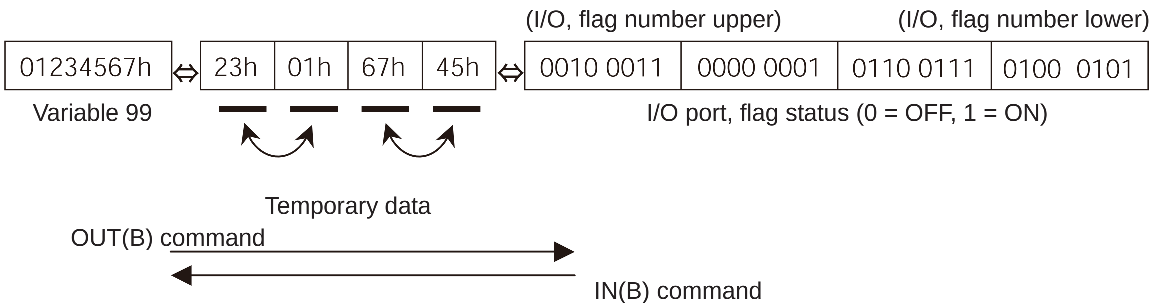
Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the data format for reading or writing I/O ports and flags with an IN, INB, OUT OUTB or OTPS command.
Details of data in each format type are shown for the IN, INB, OUT and OUTB commands. Data details of the OTPS command are the same as those of the OUT command, where the only difference is that variable 99 in the OUT command is replaced with current position data in the OTPS command.

1) Operand 1 = 0 (Default status when a FMIO command has not been executed)
Data is read or written without being reversed.



2) Operand 1 = 1
Data is read or written after its upper 8 bits and lower 8 bits are reversed every 16 bits.



3) Operand 1 = 2
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits.

(I/O, flag number upper) (I/O, flag number lower)

01234567h	⇔	45h	67h	01h	23h	⇔	0100 0101	0110 0111	0000 0001	0010 0011

Variable 99 | I/O port, flag status (0 = OFF, 1 = ON)

Temporary data

OUT(B) command →

← IN(B) command

4) Operand 1 = 3
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits and its upper 8 bits and lower 8 bits are reversed every 16 bits.

(I/O, flag number upper) (I/O, flag number lower)

01234567h	⇔	67h	45h	23h	01h	⇔	0110 0111	0100 0101	0010 0011	0000 0001

Variable 99 | I/O port, flag status (0 = OFF, 1 = ON)

Temporary data

OUT(B) command →

← IN(B) command

(Note) The FMIO command is supported by: Main application Ver.0.56 or later
PC software Ver.2.0.45 or later
Teaching pendant Ver.1.13 or later

[Example 1] Variable 99 = 00123456h (Decimal: 1193046, BCD: 123456)

00123456h ← OUT(B) command / IN(B) command → Variable 99: 1193046 (IN/OUT command); 123456 (INB/OUTB command)

OUT(B) command ↓ ↑ IN(B) command

	Temporary data				⇔	(I/O, flag number upper)			(I/O, flag number lower)
FMIO=0	00h	12h	34h	56h	⇔	0000 0000	0001 0010	0011 0100	0101 0110
FMIO=1	12h	00h	56h	34h	⇔	0001 0010	0000 0000	0101 0110	0011 0100
FMIO=2	34h	56h	00h	12h	⇔	0011 0100	0101 0110	0000 0000	0001 0010
FMIO=3	56h	34h	12h	00h	⇔	0101 0110	0011 0100	0001 0010	0000 0000

Temporary data — OUT(B) command → / ← IN(B) command — I/O port, flag status (0 = OFF, 1 = ON)

[Example 2] Variable 99 = 00001234h (Decimal: 4660, BCD: 1234)

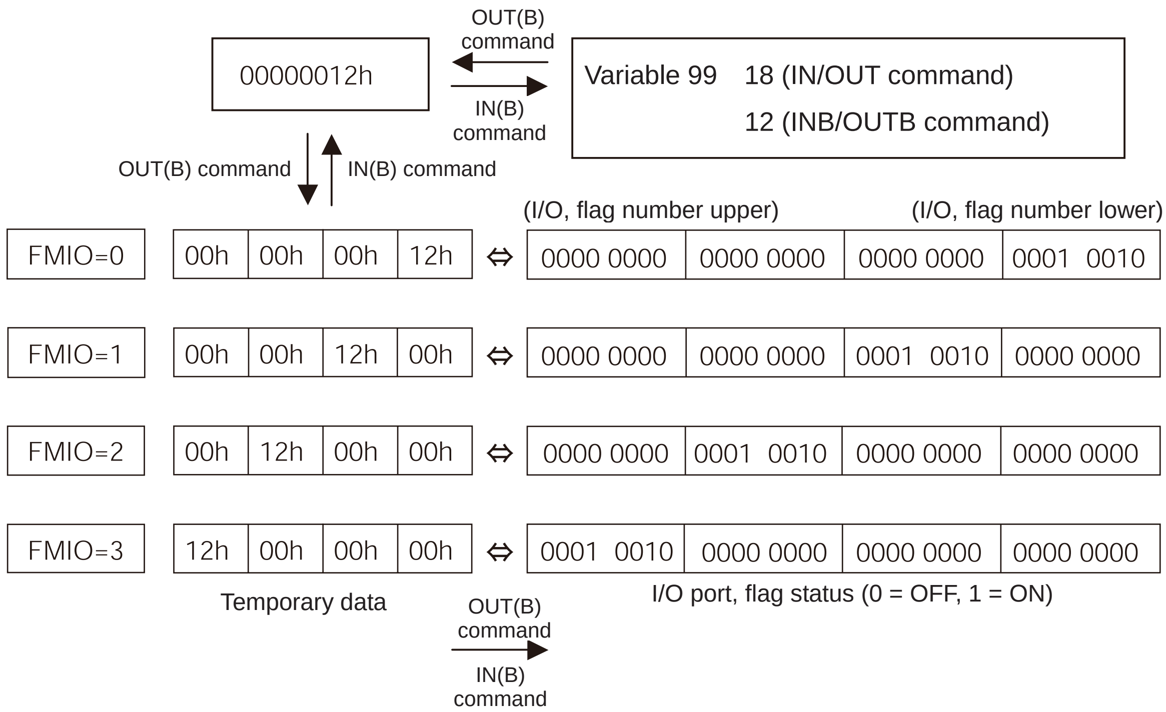
00001234h ← OUT(B) command / IN(B) command → Variable 99: 4660 (IN/OUT command); 1234 (INB/OUTB command)

OUT(B) command ↓ ↑ IN(B) command

	Temporary data				⇔	(I/O, flag number upper)			(I/O, flag number lower)
FMIO=0	00h	00h	12h	34h	⇔	0000 0000	0000 0000	0001 0010	0011 0100
FMIO=1	00h	00h	34h	12h	⇔	0000 0000	0000 0000	0011 0100	0001 0010
FMIO=2	12h	34h	00h	00h	⇔	0001 0010	0011 0100	0000 0000	0000 0000
FMIO=3	34h	12h	00h	00h	⇔	0011 0100	0001 0010	0000 0000	0000 0000

Temporary data — OUT(B) command → / ← IN(B) command — I/O port, flag status (0 = OFF, 1 = ON)

[Example 3] Variable 99 = 00000012h (Decimal: 18, BCD: 12)



● OTPS (Output current position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OTPS	Output port number	Axis number	CC

* This command is supported by SSEL controller main application Ver.0.22 or later.

Applicable models
SSEL ○ Other than SSEL ×

[Function]
- Output current position data to an output port.
- The current position data corresponding to the axis number specified in operand 2 is output to 32 bits of ports starting from the output port specified in operand 1.
- If the command is executed with 0 specified in operand 1, the command will become invalid and refreshing of current position data at the specified output will stop.
- When this command is executed, current position data will be refreshed continuously at the specified output port until the program in which this command was input is stopped or otherwise the command becomes invalid.
- 32 bits binary data (extended by sign) is output. The minimum unit is 0.001mm.

(Note)
- Only output ports of No. 300 or higher port numbers (multiples of 8) can be specified in operand 1.
- Only network output ports are supported.
- Even if this command is executed, output data remains indeterminable if home return is not yet completed.
- The output data format can be changed using the FMIO command (refer to the section on “FMIO command”). Note, however, that data is output in the FMIO-specified format when this command is called.

[Example 1]

When OTPS 300 1 is executed:
If the current position is -0.012mm, it is expressed as -12 (decimal) or FFFFFFF4 (binary) in units of 0.001mm.
Accordingly, FFFFFFF4 is output to output port No. 300 onward.

If the current position is 125.305mm, it is expressed as 125305 (decimal) or 0001E979 (binary) in units of 0.001mm.

Accordingly, 0001E979 is output to output port No. 300 onward.
The statuses of output ports are shown below.

307	306	305	304	303	302	301	300
OFF	ON	ON	ON	ON	OFF	OFF	ON
315	314	313	312	311	310	309	308
ON	ON	ON	OFF	ON	OFF	OFF	ON
323	322	321	320	319	318	317	316
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
331	330	329	328	327	326	325	324
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFf

[8] Program Control

● GOTO (Jump)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GOTO	Tag number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Jump to the position of the tag number specified in operand 1.

(Note 1) A GOTO command is valid only within the same program.
(Note 2) Do not create a program that contains an infinite series of continuous movement commands using TAG-GOTO statements. Doing so will cause coordinate conversion errors to accumulate.

[Example 1]

```
TAG    1    Set a tag.
 :
 :
 :
GOTO   1    Jump to tag 1.
```

Using a GOTO command to branch out of or into any of the syntaxes listed below is prohibited.
Since the maximum number of nests is defined for each conditional branching command or subroutine call, a nest will be infinitely repeated if an ED☐☐ is not passed, and a nest (repetition) overflow error will generate. In the case of palletizing setting, an error will generate if the second BGPA is declared after the first BGPA declaration without passing an EDPA.

(1) IF☐☐ or IS☐☐ and EDIF syntax
(2) DWXX and EDDO syntax
(3) SLCT and EDSL syntax
(4) BGSR and EDSR syntax
(5) BGPA and EDPA syntax

● TAG (Declare tag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	TAG	Tag number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the tag number specified in operand 1.

[Example 1] Refer to the section on GOTO command.

● EXSR (Execute subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EXSR	Subroutine number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Execute the subroutine specified in operand 1.
A maximum of 15 nested subroutine calls are supported.

(Note) This command is valid only for subroutines within the same program.

[Example 1]

```
EXSR    1                  Execute subroutine 1.
  :
  :
EXIT
BGSR    1                  Start subroutine 1.
  :
  :
  :
EDSR                       End subroutine 1.
```

[Example 2]

```
LET     1     10           Assign 10 to variable 1.
EXSR    *1                 Execute the content of variable 1 (subroutine 10).
```

● BGSR (Start subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	BGSR	Subroutine number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare the start of the subroutine specified in operand 1.

[Example 1] Refer to the section on EXSR command.

(Note) Using a GOTO command to branch out of or into a BGSR-EDSR syntax is prohibited.

● EDSR (End subroutine)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EDSR	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare the end of a subroutine.
This command is always required at the end of a subroutine.
Thereafter, the program will proceed to the step next to the EXSR that has been called.

[Example 1] Refer to the section on EXSR command.

● EXIT (End program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	EXIT	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] End the program.
If the last step has been reached without encountering any EXIT command, the program will return to the beginning.

(Note) Status at program end

- Output ports ················Retained
- Local flags···················Cleared
- Local variables ············Cleared
- Current values ············Retained
- Global flags·················Retained
- Global variables ··········Retained

● EXPG (Start other program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	EXPG	Program number	(Program number)	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Start the programs from the one specified in operand 1 through the other specified in operand 2, and run them in parallel. Specification in operand 1 only is allowed.

[Example 1] EXPG 10 12 Start program No. 10, 11 and 12.

Error-generation/output-operation conditions
When one EXPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	A57 “Multiple program start error”	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple EXPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range	None of programs inside the specified range are running		
Error	A57 “Multiple program start error”	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

● ABPG (Abort other program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ABPG	Program number	(Program number)	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Abort other program.

(Note 1) If an ABPG command is issued while a movement command is being executed, the axes will immediately decelerate and stop.

(Note 2) Not only the operation but also the execution of the step itself will be terminated.

[Example 1] ABPG 10 12 End program No. 10, 11 and 12.

Error-generation/output-operation conditions
When one ABPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	None	C2C "Program number error"
Output operation	ON (OFF *2)	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 2… If an own task (own program) is specified in an ABPG command, the own task will be terminated and then deleted. The output will turn OFF.

When multiple ABPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *3			Program number error *1
	Registered program exists inside the specified range *4		None of programs inside the specified range are registered	
	Running program exists inside the specified range	None of programs inside the specified range are running		
Error	None	None	None	C2C "Program number error"
Output operation	ON (OFF *5)	ON	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 3… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 4… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

* 5… If an own task (own program) is included in the specified range, the own task will be terminated, upon which the processing of the ABPG command will end. Since the own task will be deleted, the result of ending the processing of specified programs will become indeterminable. Exercise caution. The output will always turn OFF regardless of the result.

● SSPG (Pause program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SSPG	Program number	(Program number)	CC

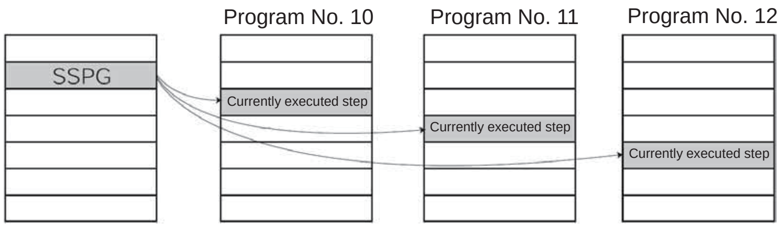
Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Pause the program from the one specified in operand 1 through the other specified in operand 2, at the current step. Specification in operand 1 only is allowed.

(Note 1) Pausing a program will also pause the operation the program has been executing.

(Note 2) Not only the operation but also the execution of the step itself will be paused.

[Example 1] SSPG 10 12 Pause program No. 10, 11 and 12 at the current step.



Error-generation/output-operation conditions

When one SSPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple SSPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range *4	None of programs inside the specified range are running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation with EXPG, ABPG, SSPG and PSPG commands. This will not affect error generation or output operation.

* 4… In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

● RSPG (Resume program)

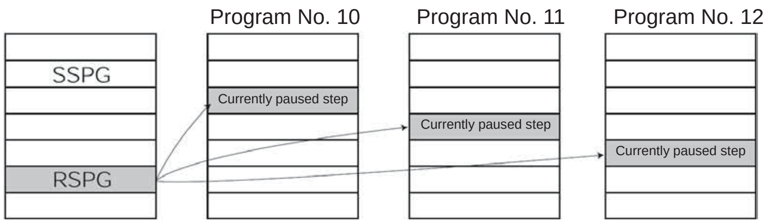
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RSPG	Program number	(Program number)	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Resume the programs from the one specified in operand 1 through the other specified in operand 2. Specification in operand 1 only is allowed.

(Note 1) Resuming a program will also resume the operation the program had been executing before the pause.

[Example 1] RSPG 10 12 Resume program No. 10, 11 and 12 from the paused step.



Error-generation/output-operation conditions

When one RSPG program is specified (only operand 1 is specified)

Status of the specified program	No program number error *1			Program number error *1
	Program already registered		Program not yet registered	
	Program running	Program not running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple RSPG programs are specified (both operands 1 and 2 are specified)

Status of the specified program	No program number error *2			Program number error *1
	Registered program exists inside the specified range *3		None of programs inside the specified range are registered	
	Running program exists inside the specified range *4	None of programs inside the specified range are running		
Error	None	None	C03 “Non-registered program specification error”	C2C “Program number error”
Output operation	ON	OFF	OFF	OFF

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

* 4… In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

[10] Position Operation

● PGET (Read position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PGET	Axis number	Position number	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read to variable 199 the data of the axis number specified in operand 1 in the position data specified in operand 2.
If the position data table contains no data to be loaded (= the position data display on the teaching pendant shows X.XXX or position data display fields in the PC software are blank) when the PGET command is executed, no data will placed in variable 199 (= the PGET command will not be executed).

[Example 1]	PGET	2	3	Read to variable 199 the data of Y-axis (axis 2) at position 3.
[Example 2]	LET	1	2	Assign 2 to variable 1.
	LET	2	3	Assign 3 to variable 2.
	PGET	*1	*2	Read to variable 199 the data of the content Y-axis (axis 2) of variable 1 at the content 3 of variable 2 at the position number.

● PPUT (Write position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PPUT	Axis number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the value in variable 199 to the axis number specified in operand 1 in the position data specified in operand 2.

[Example 1]

LET	199	150	Assign 150 to variable 199.
PPUT	2	3	Write the content 150 of variable 199 to Y-axis (axis 2) at position 3.

[Example 2]

LET	199	150	Assign 150 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2
PPUT	*1	*2	Write the content 150 of variable 199 to the content Y-axis (axis 2) of variable 1 at the content 3 of variable 2 at the position number.

● PCLR (Clear position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PCLR	Position number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Clear the position data from the one specified in operand 1 through the other specified in operand 2.
When data is cleared, the field no longer contains data, which is different from the value of 0.000. The position data display on the teaching pendant changes to ×.××× while position data fields in the PC software become blank.

(Note 1) The comment on each position data are also subject to delete. If the position data with a comment is deleted by PCLR Command and software reset is conducted or the power is turned OFF without flash ROM writing being conducted, 22B “Position Data Comment Lost Error” will occur.

[Example 1]

PCLR	10	20	Clear the data from position No. 10 through 20.

[Example 2]

LET	1	10	Assign 10 to variable 1.
LET	2	20	Assign 20 to variable 2.
PCLR	*1	*2	Clear the data of the content of variable 1 (position 10) through the content of variable 2 (position 20).

● PCPY (Copy position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PCPY	Position number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Copy the position data specified in operand 2 to the position number specified in operand 1.

(Note 1) The comment on each position data are also subject to copy. If a change is made to the position data comment of the area to recover at by PCPY Command and software reset is conducted or the power is turned OFF without flash ROM writing being conducted, 22B "Position Data Comment Lost Error" will occur.

[Example 1]	PCPY	20	10	Copy the data of position No. 10 to position No. 20.
[Example 2]	LET	1	20	Assign 20 to variable 1.
	LET	2	10	Assign 10 to variable 2.
	PCPY	*1	*2	Copy the data of the content of variable 2 (position 10) to the content of variable 1 (position 20).

● PRED (Read current position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PRED	Axis pattern	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read the current position of the axis specified in operand 1 to the position specified in operand 2.

[Example 1] PRED 11 10 Read the current positions of X and Y-axis to position No. 10.

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:

11 (binary) → 3 (decimal)

```
LET   1   3    Assign 3 to variable 1.
PRED  *1  10
```

[Example 3]

```
LET   1   10   Assign 10 to variable 1.
PRED  11  *1   Read the current positions of X and Y-axis to the
               content of variable 1 (position 10).
```

●PRDQ (Read current axis position (single-axis direct))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PRDQ	Axis number	Variable number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read the current position of the axis number specified in operand 1 to the variable specified in operand 2.

[Example] PRDQ 2 100 Read the current position of Y-axis (axis) 2 to variable 100.

● PTAM (Substitution of target arm system data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTAM	Variable No. (Two variables used in a row)	Position number	CP

Applicable models
XSEL-RX/SX/RXD/SXD ○ MSEL-PCX/PGX ○ Other than above ×

[Function] It writes the two types of arm system indications in a row from the variable number indicated in Operand 1 into the position data indicated in Operand 2.

Arm System Indication Type	Operand 1 Variable Setting
Right Arm System Substitution	1
Left Arm System Substitution	-1
Arm System Data Clear	0

Variation No. n in Operand 1 is the target arm system indication of the 1st to 4th axes or 1st to 3rd axes, and variable No. n+1 is that of the 5th to 8th axes. For the type to connect one unit of SCARA, make sure to set 0 to the indicated variable No. n+1.

Variable No.	Description	Setting Range	
		1 unit of SCARA connected	2 units of SCARA connected
n	1st to 4th axes (1st to 3rd axes) SCARA target arm system	-1, 0, 1	-1, 0, 1
n + 1	5th to 8th axes SCARA target arm system	0 (Reserved by the system)	-1, 0, 1

[Example]

LET	20	1	Set right arm system to 1st to 4th axes (1st to 3rd axes)
LET	21	0	Set 0 to 5th to 8th axes (system reservation as it is not connected)
PTAM	20	10	Write the arm system data stored in Variable No. 20 and 21 to Position No. 10.

Variable No.20	1	: Right arm system indicated in 1st to 4th axes (1st to 3rd axes)
Variable No.21	0	: System reserved (not connected) in 5th to 8th axes

No.(Name)	Axis1	Axis2	Axis3	Axis4	Arm1-4	Vel	Acc	Dcl
10( )	250.000	250.000	100.000	0.000	Right			

● PTST (Check position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTST	Axis pattern	Position number	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Check if valid data is contained in the axis pattern specified in operand 1 at the position number specified in operand 2.
The output turns ON when the data specified by the axis pattern is not entirely available (= the position data display on the teaching pendant is ×.××× or position data fields in the PC software are blank).
0 is treated as valid data.

[Example 1] PTST 11 10 300 Turn ON output 300 if there are no valid values of X and Y-axis at position 10.
Output 300 will turn OFF if the position data is given as follows:

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 1 3 Assign 3 to variable 1.
PTST *1 10 300

[Example 3] LET 1 11 Assign 11 to variable 1.
PTST 1011 *1 600 Turn ON flag 600 if there are no valid values in the data of X, Y and R-axis at the content of variable 1 (position 11).
Flag 600 will turn ON if the position data is given as follows:

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
9( )							
10( )	200.000	100.000					
11( )			150.000				
12( )							

● PVEL (Assign speed data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PVEL	Speed	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the CP operation speed/linear axis speed specified in operand 1 to the position number specified in operand 2. The unit of operand 1 is [mm/sec].

(Note 1) If a negative value is written in PVEL Command, an alarm will be generated when this position is indicated in a movement.

(Note 2) If 0 is set in Operation 1, the speed setting in the indicated position number is deleted.

[Example 1]

```
PVEL   100   10     Write speed 100mm/s to position No. 10.
```

[Example 2]

```
LET    1     100    Assign 100 to variable 1.
LET    2     10     Assign 10 to variable 2.
PVEL   *1    *2     Write the content of variable 1 (speed 100mm/s) to
                    the content of variable 2 (position 10).
```

● PACC (Assign acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PACC	Acceleration	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the acceleration in CP operation/acceleration in linear axis operation specified in operand 1 to the position number specified in operand 2. The unit of operand 1 is [G] and the entered value is valid to the second decimal point.

[Example 1]	PACC	0.3	10	Write acceleration 0.3G to position No. 10.
[Example 2]	LET	100	0.3	Assign 0.3 to variable 100.
	LET	2	10	Assign 10 to variable 2.
	PACC	*100	*2	Write the content of variable 100 (acceleration 0.3G) to the content of variable 2 (position 10).

(Note 1) Range check is not performed for a PACC command.
(Note 2) If Operation 1 is set to 0, the acceleration setting on the indicated position number gets deleted.

● PDCL (Assign deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PDCL	Deceleration	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the CP operation deceleration/linear axis deceleration specified in operand 1, into the position number specified in operand 2.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

[Example 1] PDCL 0.3 3 Assign 0.3 to the deceleration data at position No. 3.

(Note 1) If Operation 1 is set to 0, the deceleration setting on the indicated position number gets deleted.

● PAXS (Read axis pattern)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAXS	Variable number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Store the axis pattern at the position specified in operand 2 to the variable specified in operand 1.

[Example 1]	PAXS	1	98	Read the axis pattern at position 98 to variable 1. If the position is given as follows, “3” (binary 0011) will be read to variable 1.
[Example 2]	LET	1	3	Assign 3 to variable 1.
	LET	2	101	Assign 101 to variable 2.
	PAXS	*1	*2	Read the axis pattern at the content of variable 2 (position 101) to the content of variable 1 (variable 3). If the point is given as follows, “8” (binary 1000) will be stored in variable 3.

The table below shows different positions and corresponding values stored in a variable.

No.(Name)	Axis1	Axis2	Axis3	Axis4
98( )	200.000	100.000		
99( )	350.000		120.000	
100( )				
101( )				180.000

0 0 1 1 = 2 + 1 = 3
0 1 0 1 = 4 + 1 = 5
0
1 0 0 0 = 8

● PSIZ (Check position data size)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PSIZ	Variable number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set an appropriate value in the variable specified in operand 1 in accordance with the parameter setting.

- When “Other parameter No. 23, PSIZ function type” = 0
  The maximum number of position data that can be stored in the controller will be set.
  (Regardless of whether the data are used or not.)
- When “Other parameter No. 23, PSIZ function type” = 1
  The number of position data used will be set.

[Example]

```
PSIZ     1
```

When “Other parameter No. 23, PSIZ function type” = 0
The maximum number of position data that can be stored in variable 1 will be set.
When “Other parameter No. 23, PSIZ function type” = 1
The number of position data currently used will be set in variable 1.

● GTAM (Acquirement of target arm system data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTAM	Variable No. (Two variables used in a row)	Position number	CP

Applicable models
XSEL-RX/SX/RXD/SXD ○
MSEL-PCX/PGX ○
Other than above ×

[Function] Acquire the target arm system indication from the position data indicated in Operand 2, and set it in the two variables in a row indicated in Operand 1.

Arm System Indication Type	Operand 1 Variable Setting
Right Arm System Substitution	1
Left Arm System Substitution	-1
Not to be indicated	0

Variation No. n in Operand 1 is the target arm system indication of the 1st to 4th axes or 1st to 3rd axes, and variable No. n+1 is that of the 5th to 8th axes. For the type to connect one unit of SCARA, make sure to set 0 to the indicated variable No. n+1.

Variable No.	Description	Output Range	
		1 unit of SCARA connected	2 units of SCARA connected
n	1st to 4th axes (1st to 3rd axes) SCARA target arm system	-1, 0, 1	-1, 0, 1
n + 1	5th to 8th axes SCARA target arm system	Indefinite	-1, 0, 1

[Example] GTAM 20 10 Set the arm system data in Position No. 10 to Variable No. 20.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Arm1-4	Vel	Acc	Dcl
10( )	250.000	250.000	100.000	0.000	Right			

↓

Variable No.20 | 1 | : 1st to 4th axes (1st to 3rd axes) arm system

Variable No.21 | 0 | : 5th to 8th axes arm system (not connected)

● GVEL (Get speed data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GVEL	Variable number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Obtain speed data from the speed item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GVEL 100 10 Set the speed data at position No. 10 in variable 100.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
9( )							
10( )	250.000	100.000	100.000	30.000	100	0.80	0.80
11( )							

If the position data is set as above when the command is executed, 100 will be set in variable 100.

● GACC (Get acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GACC	Variable number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Obtain acceleration data from the acceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GACC 100 10 Set the acceleration data at position No. 10 in variable 100.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
9( )							
10( )	250.000	100.000	100.000	30.000	100	0.80	0.80
11( )							

If the position data is set as above when the command is executed, 0.8 will be set in variable 100.

● GDCL (Get deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GDCL	Variable number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Obtain deceleration data from the deceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GDCL 100 10 Set the deceleration data at position No. 10 in variable 100.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
9( )							
10( )	250.000	100.000	100.000	30.000	100	0.80	0.80
11( )							

If the position data is set as above when the command is executed, 0.8 will be set in variable 100.

[11] Actuator Control Declaration

● VEL (Set speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	VEL	Speed	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the actuator travel speed in the value specified in operand 1.
In the case of a SCARA, set the operating speed for CP operation.
The unit is [mm/sec].
The maximum speed will vary depending on the model of the actuator connected.
Set a speed not exceeding the applicable maximum speed.

(Note 1) Decimal places cannot be used. An error will generate
(Note 2) The minimum speed is 1mm/sec.

[Example 1]

VEL	100	Set the speed to 100mm/sec.
MOVL	1	Move to point 1 at 100mm/sec.

[Example 2]

VEL	500	Set the speed to 500mm/sec.
MOVL	2	Move to point 2 at 500mm/sec.

● VELS (Dedicated SCARA command/Set speed ratio)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	VELS	Ratio	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set in operand 1 the moving speed for SCARA PTP operation command (angular speed for all axes other than Z) as a ratio of the maximum PTP speed. Operand 1 must be set with an integer (unit: %).

(Note 1) If a RIGH or LEFT command is used, the speed must be set with VELS even when a SCARA PTP operation command is not used.

[Example 1]

VELS	50	Set the moving speed for PTP operation command to 50% of the maximum value.
MOVP	1	Move to position No. 1 via PTP at 50% of the maximum speed.

● OVRD (Override)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OVRD	Speed ratio	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Reduce the speed in accordance with the ratio specified in operand 1 (speed coefficient setting). The speed ratio is set in a range from 1 to 100%.
A speed command specifying a speed below 1mm/sec can be generated using OVRD.
Speed-command smoothing limit speed : 1pulse/msec
Speed-command generation limit speed : 1pulse/256msec
(Smoothing in actual operation is not guaranteed. It must be confirmed with the actual equipment.)
1pulse : Lead [mm]/16384 … Standard model with 1 : 1 gear ratio
(The speed set by a PAPR command (push-motion approach speed) is clamped at the lower-limit speed 1mm/sec.)

[Example 1]

VEL	100	Set the speed to 100mm/s.
OVRD	50	Reduce the speed to 50%. As a result, the actual speed will become 50mm/s.

[Example 2: SCARA robot]

VEL	150	Set the SCARA CP operation speed/linear axis speed to 150mm/sec.
VELS	90	Set the SCARA PTP operation speed ratio to 90%.
OVRD	50	Lower the speed to 50%. The SCARA CP operation speed/linear axis speed becomes 75mm/sec, while the SCARA PTP operation speed ratio becomes 45%.

Command limit speed for smooth operation:
Travel distance per encoder pulse [mm/pulse]/time [msec]
Command limit speed that can be generated:
Travel distance per encoder pulse [mm/pulse]/time [msec]
(Smoothness of actual operation cannot be guaranteed. Movement must be checked on the actual machine.)

[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001μm/pulse) × 1000
/ (2 ^ Encoder division ratio)

(Reference) Use the values of the following parameters for the above calculation formulas:

Encoder resolution:	Axis-specific parameter No. 42
Encoder division ratio:	Axis-specific parameter No. 43
Screw lead:	Axis-specific parameter No. 47
Gear ratio numerator:	Axis-specific parameter No. 50
Gear ratio denominator:	Axis-specific parameter No. 51

● ACC (Set acceleration)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ACC	Acceleration	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the acceleration for actuator operation in operand 1.
For SCARA robot, the setting is the operational acceleration speed for CP operation.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

(Note) [Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX]
If no acceleration is set in the position data or by an ACC command when the actuator moves, the actuator uses the default value registered in all-axis parameter No. 11, “Default acceleration”.
[XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX]
If no acceleration is set in the position data or by an ACC command during CP operation, a SCARA robot uses the default value registered in all-axis parameter No. 11, “Default CP acceleration for SCARA axis”, while a linear axis uses the default value registered in all-axis parameter No. 200, “Default acceleration for linear axis”.

[Example 1] ACC 0.3 Set the acceleration to 0.3G.

(Note) Setting an acceleration exceeding the specified range for the actuator may generate an error. It may also result in a failure or shorter product life.

● ACCS (Dedicated SCARA command/Set acceleration ratio)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ACCS	Ratio	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set in operand 1 the acceleration for movement by SCARA PTP operation command (angular acceleration for all axes other than Z) as a ratio of the maximum PTP acceleration. Operand 1 must be set with an integer (unit: %).

(Note 1) For the acceleration ratio setting, make sure to refer to “Caution for Use” in Vertical Articulated Robot IX Series Instruction Manual provided separately.

[Example] ACCS 50 Set the acceleration for movement by PTP operation command to 50% of the maximum value.

● DCL (Set deceleration)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DCL	Deceleration	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the deceleration for actuator operation in operand 1.
For SCARA robot, the setting is the operational deceleration speed for CP operation.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

(Note) [Other than XSEL- JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX]
If the position data contains no deceleration AND deceleration is not set by a DCL command, the actuator will move based on the default value set in "All-axis parameter No. 12, Default deceleration".
A DCL command cannot be used with CIR and ARC commands.
[XSEL- JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX]
If no deceleration is set in the position data or by a DCL command during CP operation, a SCARA robot uses the default value registered in all-axis parameter No. 12, "Default CP deceleration for SCARA", while a linear axis uses the default value registered in all-axis parameter No. 201, "Default acceleration for linear axis".
DCL is invalid with respect to a CIR or ARC command.

[Example] DCL 0.3 Set the deceleration to 0.3G.

(Note) Setting a deceleration exceeding the specified range for the actuator may generate an error. It may also result in a failure or shorter product life.

● DCLS (Dedicated SCARA command/Set deceleration ratio)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DCLS	Ratio	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set in operand 1 the deceleration for movement by SCARA PTP operation command (angular deceleration for all axes other than Z) as a ratio of the maximum PTP deceleration. Operand 1 must be set with an integer (unit: %).

(Note 1) For the deceleration ratio setting, make sure to refer to “Caution for Use” in Vertical Articulated Robot IX Series Instruction Manual provided separately.

[Example] DCLS 50 Set the acceleration for movement by PTP operation command to 50% of the maximum value.

● SCRV (Set sigmoid motion ratio) • • • 1/3

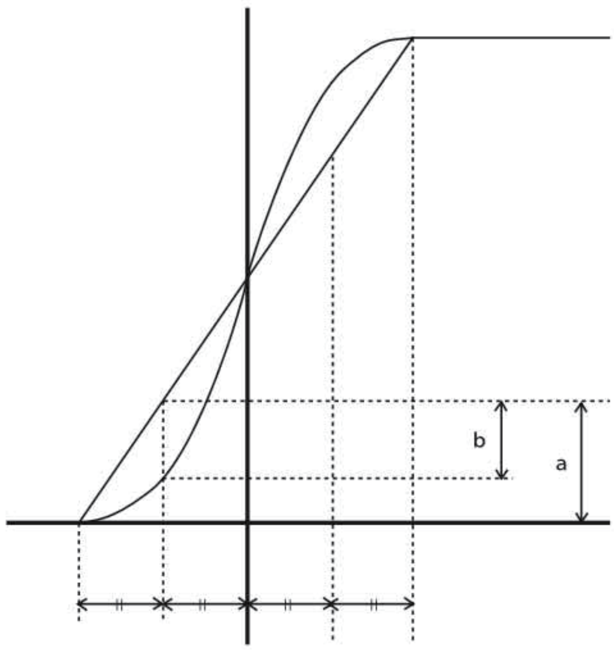
Extension Condition (LD, A, O, AB, OB)	Input condition (I/O flag)	Command,declaration			Output (Output,flag)
		Command, declaration	Operand1	Operand2	
E	N, Cnd	Cmnd	Operand1	Operand2	Pst
Optional	Optional	SCRV	Ratio	Prohibited	CP

Applicable models (Refer to the following pages for the models marked with × in the table below.)								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
○	×	×	○	×	○	○	TT:○, TTA:×	×

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)

$$\frac{b}{a} \times 100\ (\%)$$

If the ratio is not set using this command or 0% is set,a trapezoid motion will be implemented.
A SCRV command can be used with the following commands :
MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, TMPI, TMLI, RIGH, LEFT



[Example] SCRV 30 Set the sigmoid motion ratio to 30%

● SCRV (Set sigmoid motion ratio) • • • 2/3

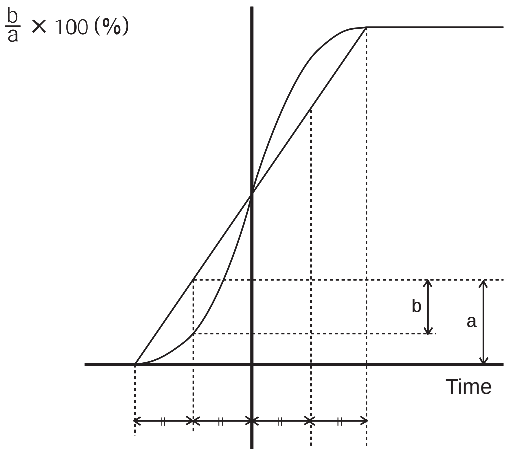
Extension Condition (LD, A, O, AB, OB)	Input condition (I/O flag)	Command,declaration			Output (Output,flag)
		Command, declaration	Operand1	Operand2	
E	N, Cnd	Cmnd	Operand1	Operand2	Pst
Optional	Optional	SCRV	Ratio	(S-motion type)	CP

Applicable models (Refer also to the previous and following pages for the models marked with × in the table below.)								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	○	×	○	○	×	×	×

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)
If the ratio is not set using this command or 0% is set, a trapezoid motion will be implemented.
A SCRV command can be used with the following commands：
MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, TMPI, TMLI, RIGH, LEFT

Value set in operand2	Description
0 or no specification	S-motionA
1	S-motionB (Recommended)

- S-motion A (Operand 2 = Not specified or 0)



- S-motion B (Operand 2 = 1)
If S-motionB is selected,the speed pattern becomes smoother(than the equivalent S-motion control ratio based on S-motionA).(The divergence peak relative to trapezoid motion becase smaller).

[Example] SCRV 30 Set the sigmoid motion ratio to 30%

[11] Actuator Control Declaration

● SCRV (Set sigmoid motion ratio) • • • 3/3

Extension Condition (LD, A, O, AB, OB)	Input condition (I/O flag)	Command,declaration			Output (Output,flag)
		Command, declaration	Operand1	Operand2	
E	N, Cnd	Cmnd	Operand1	Operand2	Pst
Optional	Optional	SCRV	Ratio	(S-motion type)	CP

Applicable models (Refer to the previous two pages for the models marked with × in the table below.)								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	×	×	×	×	×	○	○

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)
If the ratio is not set using this command or 0% is set,a trapezoid motion will be implemented.

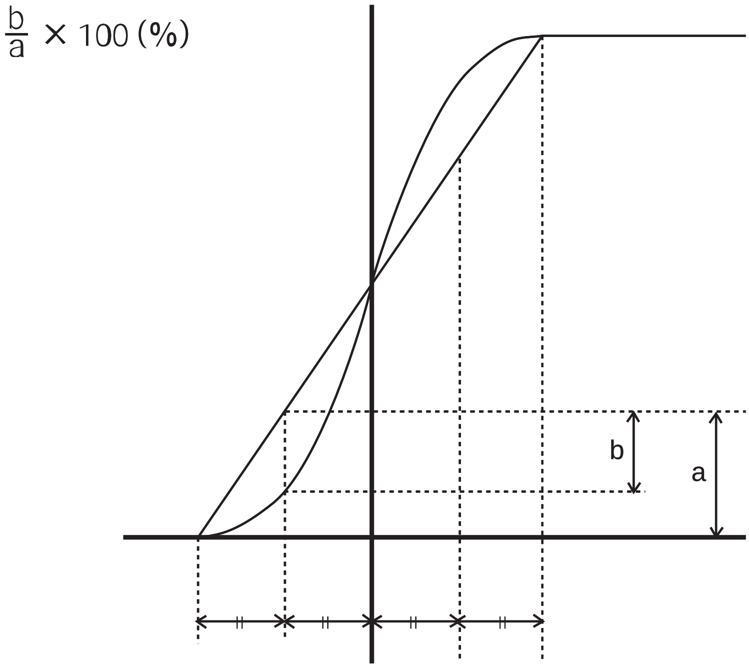
XSEL-P/Q/PCT/QCT are available to select operand2.(S-shaped type).
(Main application Ver.1.25 and later)
Operand 2 can be inputted from IA-T-X(D):Ver.1.52 or subsequent ones after teaching box SEL-T(D):Ver.1.12 after PC software:Ver.7.7.12.0.

Model Name	Setting in Operand 2 (S-shaped Type)	S-shaped Motion Class	S-shaped Motion Effective Command Group (See the table below)
XSEL -P/Q	Not set, 0	A	1)
	1	B	1)
	2	A	2) (Note 2)
	3	B	2) (Note 2)
XSEL -PCT/ QCT	Not set, 0	B (Note 1)	1)
	1		1)
	2		2) (Note 2)
	3		2) (Note 2)

Note 1 The class of S-shaped motion is compulsorily B.
Note 2 S-shaped Motion is effective also at the speed change point (position joint point) during PATH Command. If S-shaped Motion is activated, constant velocity or track could be lost. Use the unit with S-shaped Motion ineffective in such processes as applying paint or glue, in which the constant velocity and track are important.

Effective Command Group	SCRV Effective Command
1)	MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN
2)	MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, PATH, ARCH, PACH

- S-motion A



- S-motion B
  In this class, operates with a speed pattern smoother than the control of S-shaped Motion Class A. (Estrangement peak with Trapezoid Motion becomes small.)

[Example]SCRV	30	1	Set S-shaped motion ratio 30% and S-shaped motion class A.

● OFST (Set offset)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OFST	Axis pattern	Offset value	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Add the offset in operand 2 to the target value for the axis pattern specified in operand 1 when the actuator moves, to reset the target value and operate the actuator accordingly.
The offset is set in mm, and the effective resolution is 0.001mm.
A negative offset may be specified as long as the operation range is not exceeded.

(Note) An OFST command cannot be used outside the applicable program. To use OFST in multiple programs, the command must be executed in each program.
An OFST command cannot be used with MVPI, MVTI, TMLI and TMPI commands.

[Example 1]

```
OFST    110    50        Add 50mm to the specified positions of Y-axis and
                         Z-axis.
```

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
110 (binary) → 6 (decimal)

```
LET     1      6         Assign 6 to variable 1.
OFST    *1     50
```

[Example 3]

```
LET     1      30        Assign 30 to variable 1.
OFST    1000   *1        Add the content of variable 1, or 30°, to the
                         specified position of R-axis.
```

● DEG (Set arc angle)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DEG	Angle	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set a division angle for the interpolation implemented by a CIR (move along circle) or ARC (move along arc) command.
When CIR or ARC is executed, a circle will be divided by the angle set here to calculate the passing points.
The angle is set in a range from 0 to 120°.
If the angle is set to “0”, an appropriate division angle will be calculated automatically so that the actuator will operate at the set speed (maximum 180°).
The angle is set in degrees and may include up to one decimal place.

(Note) If a CIR or ARC command is executed without setting an angle with this command, the default value registered in “All-axis parameter No. 30, Default division angle” will be used.

[Example 1] DEG 10 Set the division angle to 10°.

● BASE (Specify axis base)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	BASE	Datum axis number	Prohibited	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Count the axes sequentially based on the axis number specified in operand 1 being the first axis.
BASE Command is available in PRED, PRDQ, AXST, actuator control commands, ARCH, PACH, PMVP, PMVL, zone commands, actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Note that each zone range is assigned to the actuator via parameter.

(Note 1) For XSEL-RX/SX/RXD/SXD, GRP and BASE Commands are available in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Refer to the caution note for GRP and BASE Commands.

[Example 1]

BASE	5		Axis 5 is considered the first axis.
HOME	1		Axis 5 returns to the home.
HOME	10		Axis 6 returns to the home.

[Example 2]

LET	1	5	Assign 5 to variable 1.
BASE	*1		The content of variable 1 (axis 5) will be considered as the first axis.

Thereafter, axes 5 and 6 move according to the specifications for axes 1 and 2.

● GRP (Set group axes)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GRP	Axis pattern	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Allow only the position data of the axis pattern specified in operand 1 to become valid.
The program assumes that there are no data for other axes not specified.
When multiple programs are run simultaneously, assigning axes will allow the same position data to be used effectively among the programs.
GRP Command is available in the operand axis pattern indication SEL commands except for OFST, DFTL, DFWK, DFIF, GTTL, GTWK and GTIF or the servo operation commands to use the position data, actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH and LEFT, and the system information acquirement command GARM.
GRP Command activates in the condition before the axis number changed due to BASE Command.

(Note 1) In XSEL-RX/SX/RXD/SXD, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.

[Example] Express what axis is to be used by using either “1” or “0”.

(Superior) (Inferior)

Axis No.	8-axis	7-axis	6-axis	5-axis	4-axis	3-axis	2-axis	1-axis
Use	1	1	0	1	1	1	1	1
Unused	0	0	1	0	0	0	0	0

● When using 1st and 2nd axes;

2-axis
↓
000011… (0 in front are not necessary. Remove 0 and make it 11.)
↑
1-axis

● When using 1st and 4nd axes;

4-axis
↓
1001… (In this case, 0 are necessary to express the position of the 4th axis.)
↑
1-axis

● HOLD (Hold: Declare axis port to pause)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	HOLD	(Input port, global flag)	(HOLD type)	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare an input port or global flag to pause while a servo command is being executed.
When operation is performed on the input port or global flag specified in operand 1, the current servo processing will pause. (If the axes are moving, they will decelerate to a stop.)
If nothing is specified in operand 1, the current pause declaration will become invalid.

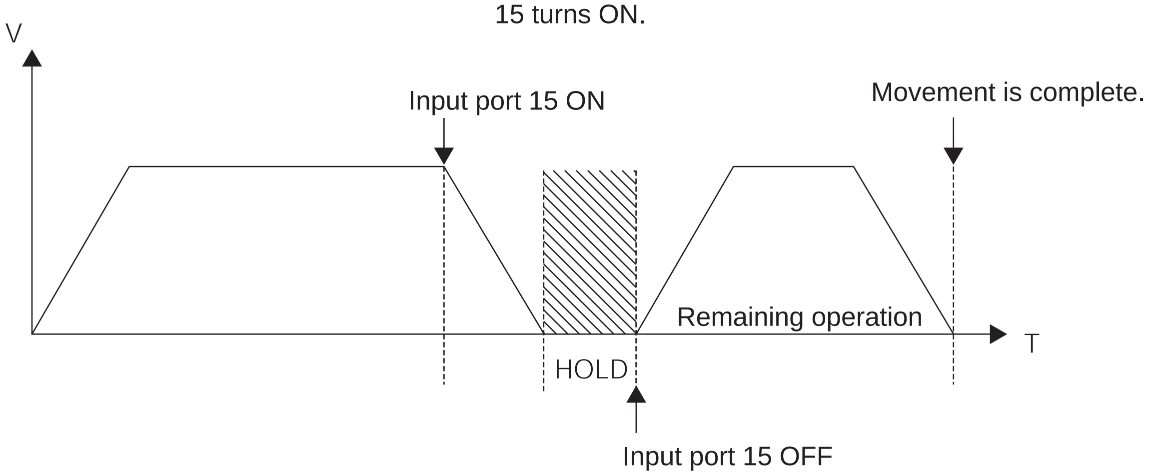
[HOLD type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)
2 = Contact b (Deceleration stop → Servo OFF (The drive source will not be cut off))

The HOLD type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current HOLD type will be used.
Using other task to issue a servo ON command to any axis currently stopped via a HOLD servo OFF will generate an “Error No. C66, Axis duplication error”. If the servo of that axis was ON prior to the HOLD stop, the system will automatically turn on the servo when the HOLD is cancelled. Therefore, do not issue a servo ON command to any axis currently stopped via a HOLD servo OFF.
If any axis currently stopped via a HOLD servo OFF is moved by external force, etc., from the stopped position, and when the servo of that axis was ON prior to the HOLD stop, the axis will move to the original stopped position when the HOLD is cancelled before resuming operation.

(Note 1) The input port or global flag specified by a HOLD declaration will only pause the axes used in the task (program) in which the HOLD is declared. The declaration will not be valid on axes used in different tasks (programs).

(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in J☐W☐ and PATH operations.)

[Example] HOLD 15 0 The axes will decelerate to a stop when input port 15 turns ON.



● CANC (Cancel: Declare axis port to abort)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CANC	(Input port, global flag)	(CANC type)	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare an input port or global flag to abort while a servo command is being executed.
When operation is performed on the input port or global flag specified in operand 1, the current servo processing will be aborted. (If the axes are moving, they will decelerate to a stop before the processing is aborted.)
If nothing is specified in operand 1, the current abort declaration will become invalid.

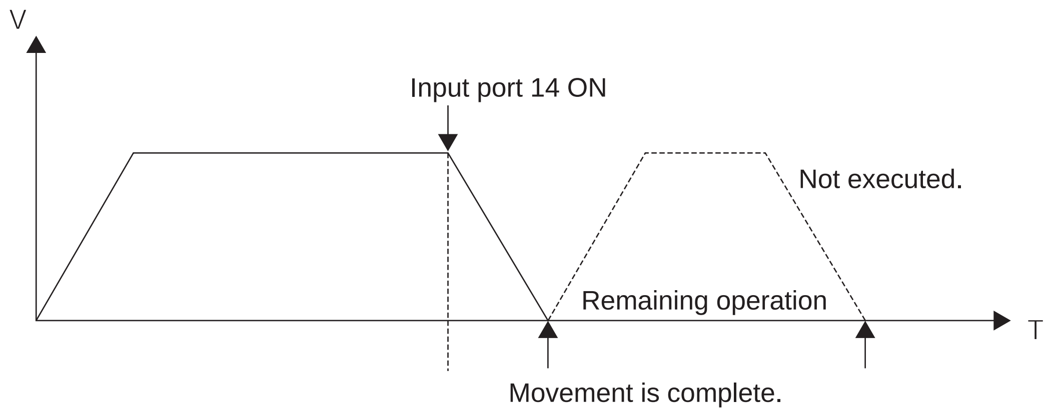
[CANC type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)

The CANC type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current CANC type will be used.

(Note 1) The input port or global flag specified by a CANC command will only abort the axes used in the task (program) in which the CANC is declared. The declaration will not be valid on axes used in different tasks (programs).

(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in JXWX and PATH operations.)

[Example] CANC 14 0 The axes will decelerate to a stop when input port 14 turns ON.



● ACMX (Indicate ACMX acceleration) (Dedicated linear axis command)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ACMX	ACMX Acceleration No.	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	○ (SSEL only)	×	○

[Function] Set the movement acceleration and deceleration of the actuator to the ACMX acceleration of the number indicated in Operand 1. Once ACMX Command is executed, the parameters registered in ACMX Acceleration No. 1 to 4 (Each Axis Parameters No. 2 to 5 and 194 to 197) are set as the movement acceleration and deceleration. It is necessary to set the parameters of ACMX acceleration in advance considering the conditions how to use the actuator (transportation weight, installation condition, etc.).

ACMX Acceleration No.	Moving Direction	Acceleration	Deceleration
1	Positive	Each Axis Parameter No.2 Setting of “ACMX + Acceleration 1”	Each Axis Parameter No.3 Setting of “ACMX - Acceleration 1”
	Negative	Each Axis Parameter No.3 Setting of “ACMX - Acceleration 1”	Each Axis Parameter No.2 Setting of “ACMX + Acceleration 1”
2	Positive	Each Axis Parameter No.4 Setting of “ACMX + Acceleration 2”	Each Axis Parameter No.5 Setting of “ACMX - Acceleration 2”
	Negative	Each Axis Parameter No.5 Setting of “ACMX - Acceleration 1”	Each Axis Parameter No.4 Setting of “ACMX + Acceleration 2”
3	Positive	Each Axis Parameter No.194 Setting of “ACMX + Acceleration 3”	Each Axis Parameter No.195 Setting of “ACMX - Acceleration 3”
	Negative	Each Axis Parameter No.195 Setting of “ACMX - Acceleration 3”	Each Axis Parameter No.194 Setting of “ACMX + Acceleration 3”
4	Positive	Each Axis Parameter No.196 Setting of “ACMX + Acceleration 4”	Each Axis Parameter No.197 Setting of “ACMX - Acceleration 4”
	Negative	Each Axis Parameter No.197 Setting of “ACMX - Acceleration 4”	Each Axis Parameter No.196 Setting of “ACMX + Acceleration 4”

(Note 1) It may generate an error is the acceleration or deceleration is set above the actuator specifications. Also, it cause a malfunction or drop of the production life.

(Note 2) The priority is put to the setting of acceleration and deceleration in the position data indicated with a movement command if there is any.

(Note 3) An operation is made within the range of the maximum acceleration and deceleration that would not exceed the ACMX acceleration/deceleration of each movement axis during the CP operation such as MOVL Command. In case constancy is required in the target acceleration/deceleration, indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 4) Do not attempt to indicate the ACMX acceleration/deceleration to the continuous movement related commands (PATH, PSPL, etc.).
It may cause a big speed drop depending on the direction of the movement position. Indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 5) Do not attempt to indicate the ACMX acceleration/deceleration to the extended motion control board axis movement commands. It would cause Error No. C89 “Acceleration/Deceleration Indication Error”. Indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 6) ACMX Command is a command dedicated for the linear drive axes.

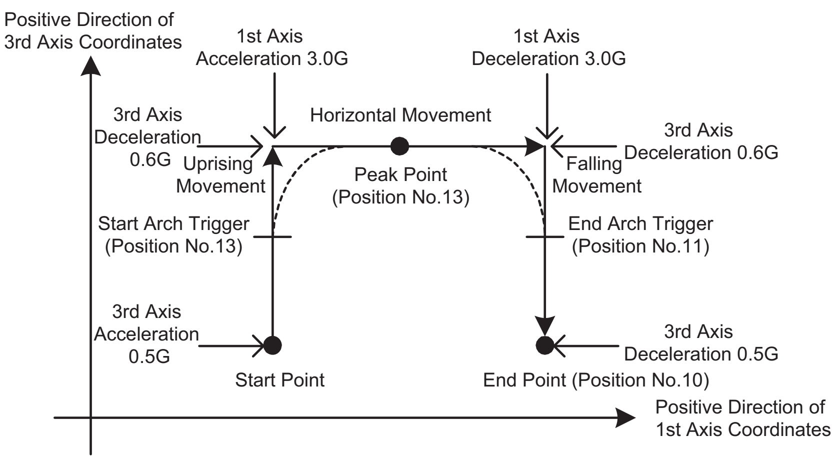
[Example 1] For arch motion movement (vertical axis to move up/down)

VLMX			Set the speed setting in VLMX Speed.
ACMX	1		Set the ACMX acceleration/deceleration of No. 1.
ACHZ	3		Indicate the 3rd axis to Z-axis for arch motion.
ATRG	13	11	
ARCH	10	12	With Position No. 12 as the peak point, move with the arch motion to Position No. 10.

- Setting for Example 1

ACMX Acceleration No.	Each Axis Parameter No.	Parameter Name	Example for Setting	
			1st Axis	3rd Axis
1	2	ACMX + Acceleration 1	300 (3.0G)	50 (0.5G)
	3	ACMX - Acceleration 1	300 (3.0G)	60 (0.6G)

- Operation of Example 1 (Acceleration/Deceleration in Arch Motion Movement)



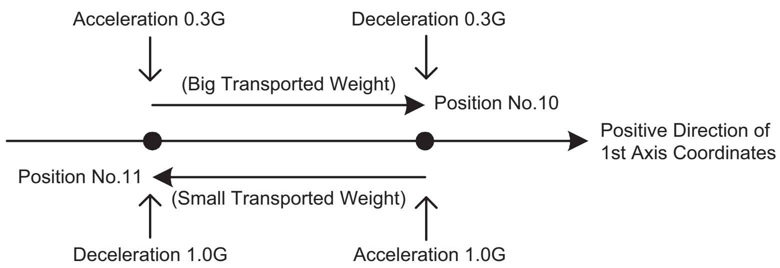
[Example 2] When the transported weight differs for going forward and backward

VLMX		Set the speed setting in VLMX Speed.
ACMX	1	Set the ACMX acceleration/deceleration of No. 1.
MOVP	10	PTP movement is made to Position No. 10.
ACMX	2	Set the ACMX acceleration/deceleration of No. 2.
MOVP	11	PTP movement is made to Position No. 11.

- Setting for Example 2

ACMX Acceleration No.	Each Axis Parameter No.	Parameter Name	Example for Setting
			1st Axis
1	2	ACMX + Acceleration 1	30 (0.3G)
	3	ACMX - Acceleration 1	30 (0.3G)
2	4	ACMX + Acceleration 2	100 (1.0G)
	5	ACMX - Acceleration 2	100 (1.0G)

- Operation of Example 2



● VLMX (Dedicated linear axis command/Specify VLMX speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	VLMX	Prohibited	Prohibited	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Set the moving speed of a linear axis to the VLMX speed (normally maximum speed).
Executing a VLMX command will set the value registered in “Axis-specific parameter No. 29, VLMX speed” as the travel speed.

(Note 1) If the VLMX speed is specified for a continuous position movement command (PATH, PSPL), the target speed to each position becomes a composite speed based on the VLMX speed to the extent that each axis does not exceed the value set in axis-specific parameter No. 28, “Maximum PTP speed (SCARA axis)/ axis-specific maximum operating speed (linear axis)”. To keep the target speed constant, you must expressly specify the speed using a VEL command.

(Note 2) Error No. C88 “Velocity Specification Error” will occur if VLMX speed is indicated in case of CP operation held on the liner axes and SCARA axes at the same time. Indicate the speed with VEL Command.

[Example]

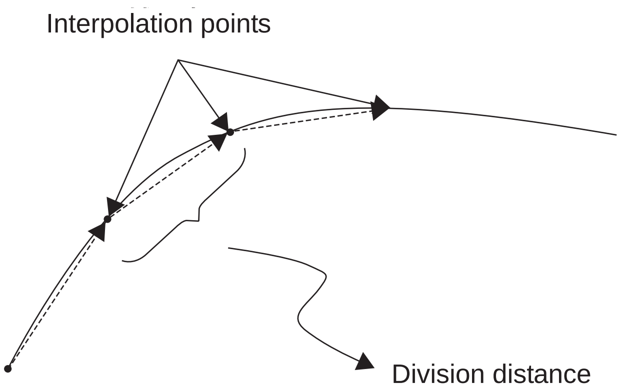
```
VEL     1000 ┐
MOVP    1    │ The speed becomes 1000mm/sec in this section.
MOVP    2    ┘
VLMX
MOVP    3    ┐ The speed becomes VLMXmm/sec in this section.
MOVP    4    ┘
```

● DIS (Set division distance at spline movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DIS	Distance	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set a division distance for the interpolation implemented by a PSPL (move along spline) command.
When a PSPL command is executed, a passing point will be calculated at each distance set here and the calculated passing points will be used as interpolation points.
If the distance is set to "0", an appropriate division distance will be calculated automatically so that the actuator will operate at the set speed.
The distance is input in mm.



(Note) If a PSPL command is executed without setting a distance with a DIS command, the default value registered in "All-axis parameter No. 31, Default division distance" will be used.

[Example] DIS 10 Set the division distance to 10mm.

● POTP (Set PATH output type)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	POTP	0 or 1	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the output type in the output field to be used when a PATH and PSPL command is executed.
When a PATH and PSPL command is executed, the output will operate as follows in accordance with the setting of the POTP command.

1) POTP [Operand 1] = 0 (ON upon completion of operation)
The output port or flag will turn ON upon completion of operation.

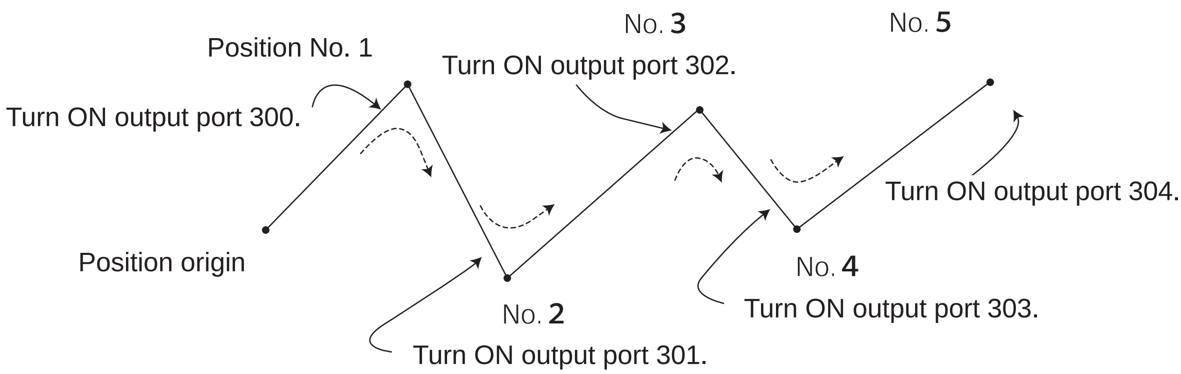
2) POTP [Operand 1] = 1 (Increment and output on approaching each position; ON upon completion of operation for the last position)
During PATH or PSPL operation, the output port number or flag number specified in the output field will be incremented and turned ON when each specified position approaches.
At the last position, however, the output will turn ON upon completion of operation. This setting provides a rough guide for output in sequence control.

(Note 1) The default value of POTP, before it is set, is "0".

(Note 2) If POTP = 1 and there is no valid data at the specified position, the output number will be incremented but the output will not turn ON. (The output number will be incremented regardless of the size of position numbers specified in operands 1 and 2 in a PATH or PSPL command.)

[Example]

POTP	1			
PATH	1	5	300	Turn ON output port No. 300 through 304 sequentially each time a specified position approaches during a pass movement from position No. 1 through 5, starting from the first position.

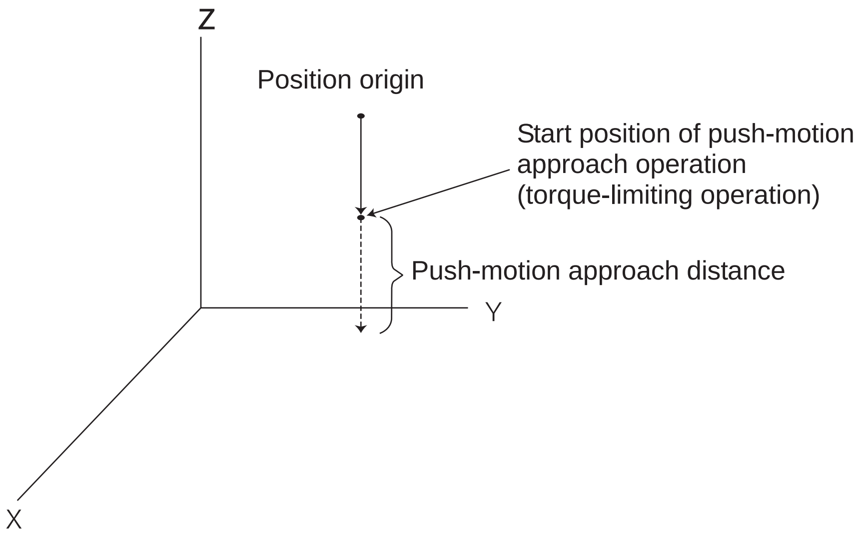


● PAPR (Set push-motion approach distance, speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAPR	Distance	Speed	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the operation to be performed when a PUSH command is executed.
Set the distance (push-motion approach distance) over which push-motion approach operation (torque-limiting operation) will be performed in operand 1 (in mm), and set the speed (push-motion approach speed) at which push-motion approach operation (torque-limiting operation) will be performed in operand 2 (in mm/sec).
The push-motion approach distance specified in operand 1 may contain up to three decimal places, while the speed specified in operand 2 cannot contain any decimal place.



[Example]

PAPR	100	30	Set the push-motion approach distance in a PUSH command to 100mm and the push-motion approach speed to 30mm/sec.
MOVP	10		
PUSH	11		

(Note) The push-motion approach speed in an OVRD command will be clamped by the minimum speed of 1mm/sec. (Correct push-motion operation is not guaranteed at the minimum speed. Operation at slow push-motion approach must be checked on the actual machine by considering the effects of mechanical characteristics, etc.)

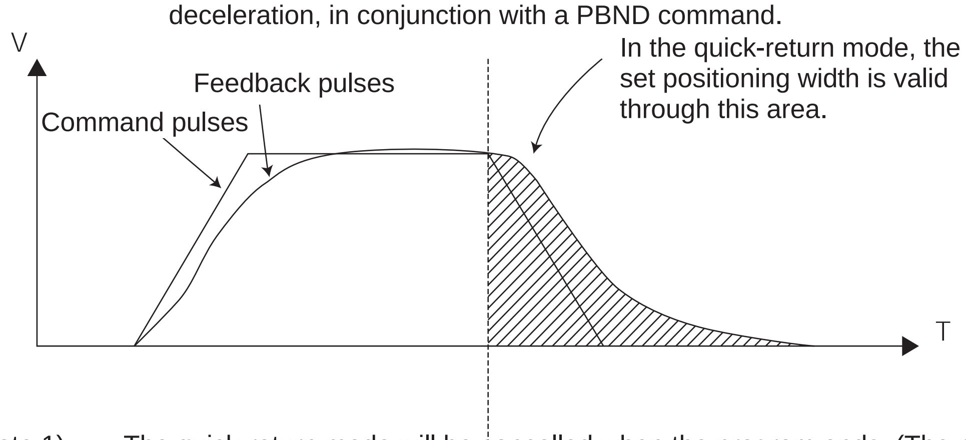
● QRTN (Set quick-return mode)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	QRTN	0 or 1	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
○	○	○	×	×	×	○	○	○ (PC/PG only)

[Function] Set and cancel the quick-return mode.

1) QRTN [Operand 1] = 0 (Normal mode)
   Positioning is deemed complete when all command pulses have been output and the current position is inside the positioning width.
   * If a deceleration command is currently executed in the quick-return mode, the system will wait for all command pulses to be output.
2) QRTN [Operand 1] = 1 (Quick-return mode)
   Positioning is deemed complete when “a normal deceleration command is currently executed (excluding deceleration due to a stop command, etc.) or all command pulses have been output” and “the current position is inside the positioning width”. This setting is used to perform other processing during deceleration, in conjunction with a PBND command.



(Note 1) The quick-return mode will be cancelled when the program ends. (The positioning width set by a PBND command will not be cancelled.)

(Note 2) If a given axis is used even once in the quick-return mode, the program will not release the right to use the axis until the QRTN is set to “0” (normal mode) or the program ends. Any attempt to use the axis from other program will generate an “Error No. C66, Axis duplication error”.

(Note 3) Following a return from a normal deceleration command in the quick-return mode, the next positioning will start after all command pulses for the previous positioning have been output. Therefore, in the quick-return mode a simple reciprocating operation will require a longer tact time because of the extra completion check. In this sense, this setting should be used only if you wish to reduce the overall tact time by performing other processing during deceleration.

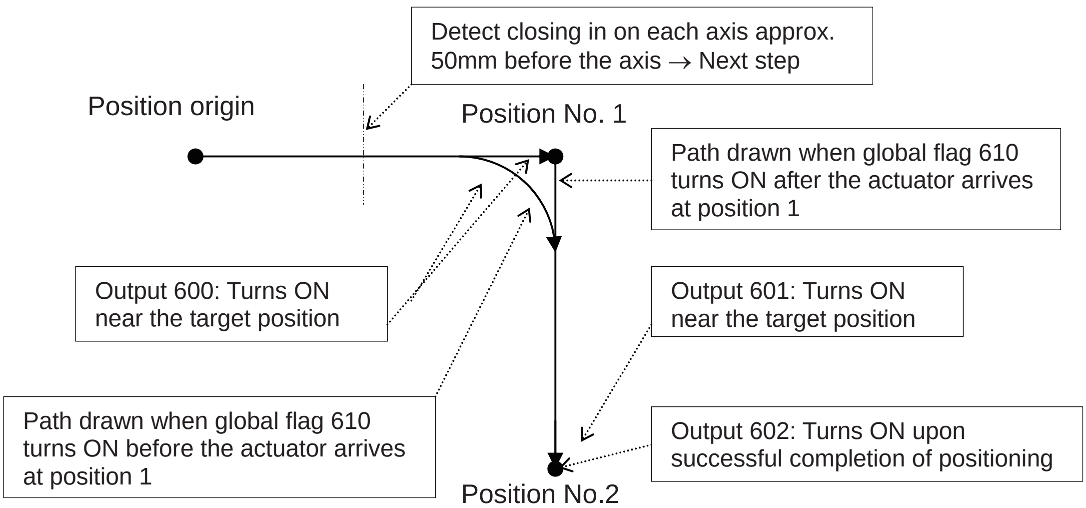
(Note 4) The quick-return mode represents very irregular processing. Therefore, be sure to revert to the normal mode when the overlay processing is completed in the necessary section.

(Note 5) The quick-return mode cannot be used with a push-motion travel command or arc interpolation command.

3) Quick return mode 2 (closeness-detection return target position addition mode)
   * XSEL-J/K only
   - When a MOVP, MOVL or PATH command (specifying the final moving position) is executed, closeness to the target position is detected when the close distance set by a NBND command is reached (or all command pulses are sent AND the positioning width is reached) while all used axes are positioning in steady state according to the applicable command, after which the command will be reset (quick return) and the SEL command in the next step will be executed.
     Set this mode if you want to perform other processing during positioning by using NBND and PEND commands together, or add a target position to operate the actuator continuously.
   - If a MOVP, MOVL or PATH command is executed again while the actuator is moving in quick return mode 2, a target position will be added and the actuator will operate continuously.

[Example]

```
  :
QRTN   2                Set quick return mode to 2
NBND   11   50          Set close position for axes 1 and 2 to 50mm
MOVL   1          600   Move to position 1 (axes 1/2)
                        (Proceed to the next step when each axis reaches
                        approx. 50mm before the position.)
WTON   610              Wait for permission of movement to position 2 (610)
MOVL   2          601   Move to position 2 (axis 3)
PEND              602   Wait for all used axes to end operation
QRTN   0                Set quick return mode to 0
  :
```



* This mode is invalid with respect to commands other than MOVP, MOVL and PATH.
  (With CIR2, ARC2, ARCC, ARCD, CIRS, ARCS, CIR, ARC, PSPL, MVPI and MVLI commands, “Error No. B24: Quick return mode error” occurs (= the command cannot be executed) because an unexpected path may be followed and a dangerous situation may result unless the start point is accurately understood.
* The close distance set by a NBND command must consider an allowance for the processing time in the next step onward following the quick return upon closeness detection (the specific processing time varies depending on the types of commands, number of steps, etc.) (this distance is not intended for use in precise processing).

* Behavior at the connection of movement commands when a new target position is added (when processing under the new movement command can be performed in time)
  If either the previous movement command (quick return) or new movement command is MOVP, the actuator starts moving to the target position under the new movement command simultaneously as the slowest axis starts decelerating under the previous movement command.
  If neither of the commands is MOVP (such as when MOVL and PATH commands are combined), the connection of operations is equivalent to what happens between normal PATH commands.
* During quick return mode 2, the output of a MOVP, MOVL or PATH command turns ON near the target position (regardless of the value set by the NBND command) (the operation is not yet complete). Use the output of a PEND command to check if the operation has completed (positioning has been successful).
* During quick return mode 2, the following tasks apply to all used axes for any operation with a MOVP command (they apply to all used axes even when specified for an individual axis):
  All stop processing including one by a STOP command
  Speed change by a CHVL command
* An attempt to switch from quick return mode 2 directly to quick return mode 1 generates “Error No. B24: Quick return mode error”.
* Software versions supporting quick return mode
  Controller main application: Ver.1.04 or later
  (excluding flash ROM 8Mbit versions)
  PC software: Ver.7.2.3.0 or later
  Teaching pendant:
  IA-T-X (D): Ver.1.44 or later
  SEL-T (D): Ver.1.02 or later

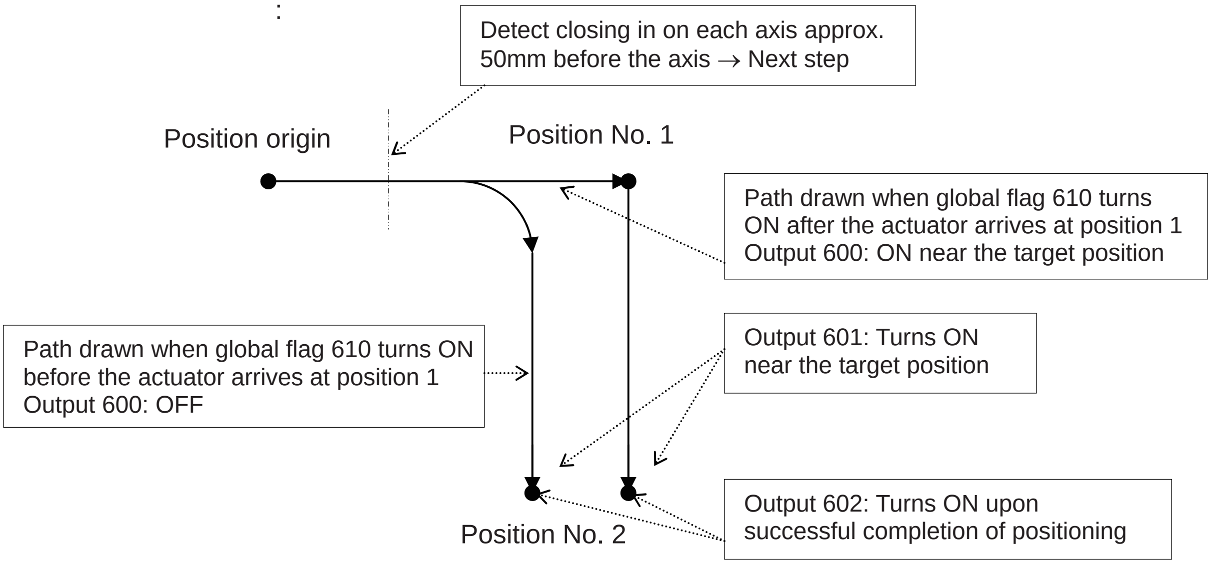
4) Quick return mode 3 (closeness-detection return target position addition mode)

* XSEL-J/K only

- When a MOVP, MOVL or PATH command (specifying the final moving position) is executed, closeness to the target position is detected when the close distance set by a NBND command is reached (or all command pulses are sent and the positioning width is reached) while all used axes are positioning in steady state according to the applicable command, after which the command will be reset (quick return) and the SEL command in the next step will be executed.
  Set this mode if you want to perform other processing during positioning by also using a NBND/PEND command or change the target position without stopping.
- If the MOVP, MOVL or PATH command is executed again while the actuator is still moving as part of quick return in quick return mode 3, the actuator changes the target position (by decelerating to stop at the previous target position to cancel the position and then starting to move to the new target position) without stopping.

[Example]

:				
QRTN	3			Set quick return mode to 3
NBND	11	80		Set close position for axes 1 and 2 to 80mm
MOVL	1		600	Move to position 1 (axes 1/2) (Proceed to the next step when each axis reaches approx. 80mm before the position.)
WTON	610			Wait for permission of movement to position 2 (610)
MOVL	2		601	Move to position 2 (axis 3)
PEND			602	Wait for all used axes to end operation
QRTN	0			Set quick return mode to 0
:				



* This mode is invalid with respect to commands other than MOVP, MOVL and PATH.
  (With CIR2, ARC2, ARCC, ARCD, CIRS, ARCS, CIR, ARC, PSPL, MVPI and MVLI commands, "Error No. B24: Quick return mode error" occurs (= the command cannot be executed) because an unexpected path may be followed and a dangerous situation may result unless the start point is accurately understood.

* The close distance set by a NBND command must consider an allowance for the processing time in the next step onward following the quick return upon closeness detection (the specific processing time varies depending on the types of commands, number of steps, etc.) (this distance is not intended for use in precise processing).

* Transition between movement commands upon target position change
  The actuator starts moving to the target position under the new movement command roughly at the same time it starts cancelling the previous movement command via forced deceleration to a stop (there is a delay corresponding to the processing time to recalculate the target position).
* During quick return mode 3, the output of a MOVP, MOVL or PATH command turns ON near the target position (regardless of the value set by the NBND command) (the operation is not yet complete). Use the output of a PEND command to check if the operation has completed (positioning has been successful).
  However, the output is invalid if the target position was changed (cancelled via forced deceleration to a stop) before the start of normal deceleration (during acceleration or constant-speed operation), and so is the S-motion mode during forced deceleration after the target position has been changed.
* During quick return mode 3, the following tasks apply to all used axes for any operation with a MOVP command (they apply to all used axes even when specified for an individual axis):
  All stop processing including one by a STOP command
  Speed change by a CHVL command
* An attempt to switch from quick return mode 3 directly to quick return mode 1 generates "Error No. B24: Quick return mode error".
* Software versions supporting quick return mode 3
  Controller main application: Ver.1.04 or later
  (excluding flash ROM 8Mbit versions)
  PC software: Ver.7.2.3.0 or later
  Teaching pendant:
  IA-T-X (D): Ver.1.44 or later
  SEL-T (D): Ver.1.02 or later

(Note 1) Following a quick return from a SEL movement command, the right to use the applicable axis is not released in the program even after the command has been reset. Accordingly, an attempt to use that axis from other program generates "Error No. C66: Multiple axis use error". To release the right to use the applicable axis, set quick return mode 0 (Normal mode = Quick return mode cancelled).

(Note 2) Quick return modes 1 to 3 are cancelled when the program ends (the close distance set by the NBND command and positioning width set by the PBND command are not cancelled).

(Note 3) At the end of combined processing requiring a quick return, be sure to reset the quick return mode to 0 (Normal mode = Quick return mode cancelled).

(Note 4) Always refer to the pages explaining the NBND and PEND commands.

● DFTL (Dedicated SCARA command/Define tool coordinate system)

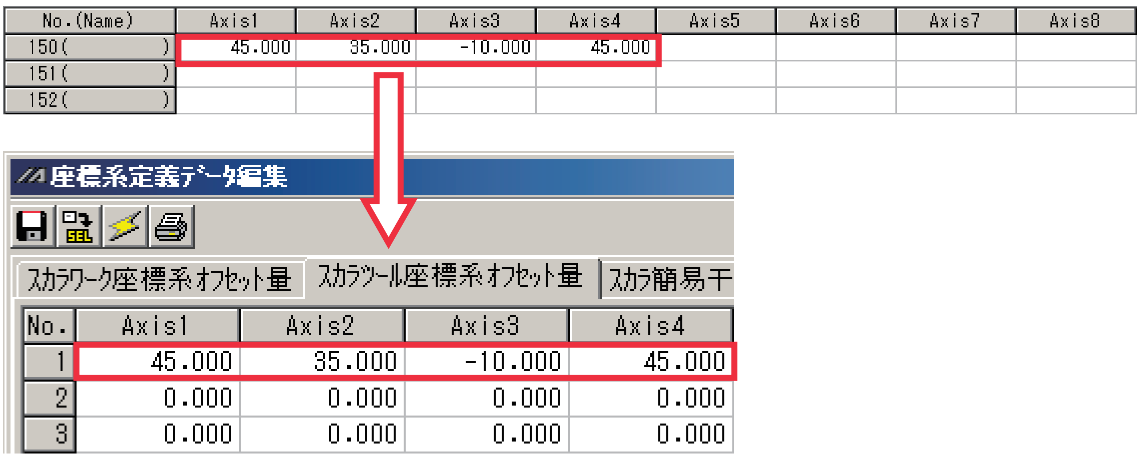
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DFTL	Tool coordinate system number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the position data in operand 2 as the tool coordinate system offset data specified in operand 1.
The position data for all the SCARA axes go into the tool coordinate system offset data, however, 0 will be set for an axis that the position data is invalid. In XSEL-RX/SX/RXD/SXD, if all the position data for the SCARA axes in one unit are invalid, data cannot be established in the tool coordinate system offset, and the data before executing DFTL Command is saved.
In MSEL-PCX/PGX, position data for four axes needs to be set in the tool coordinate system offset data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the tool coordinate system data is set from is used as the dedicated data for the tool coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the tool coordinate system in DFTL is not a target position for the additional linear axis.

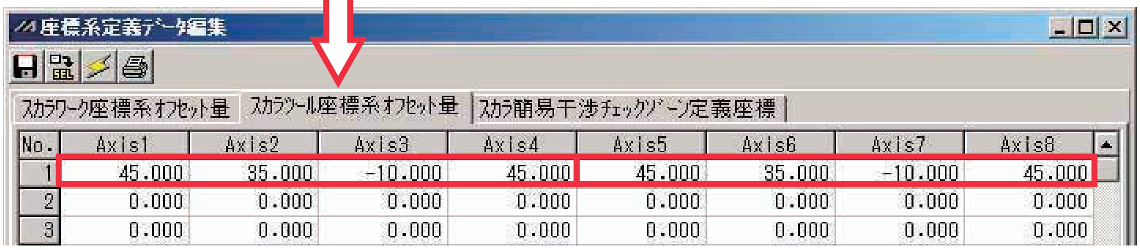
(Note 1) The tool/work coordinate systems are functions available for SCARA.
(Note 2) Since tool coordinate system No. 0 is reserved by the system as a condition specifying no tool offset, selecting this number generates “Error No. B71: Coordinate system number error”.
(Note 3) The GRP command is invalid with respect to this command.

[XSEL-JX/KX/PX/QX/RX/SX: 1 unit of SCARA connected]
[Example] DFTL 1 150



[XSEL-RXD/SXD: 2 unit of SCARA connected]
[Example 1] DFTL 1 150
In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 4 in Tool Coordinate System No. 1 as the position data in Axis 1 to 4 for the SCARA axes (1st to 4th axes) are set effective. There will be no change to Axis 5 to 8 in Tool Coordinate System No. 1 as the position data in Axis 5 to 8 for the SCARA axes (5th to 8th axes) are all set ineffective.

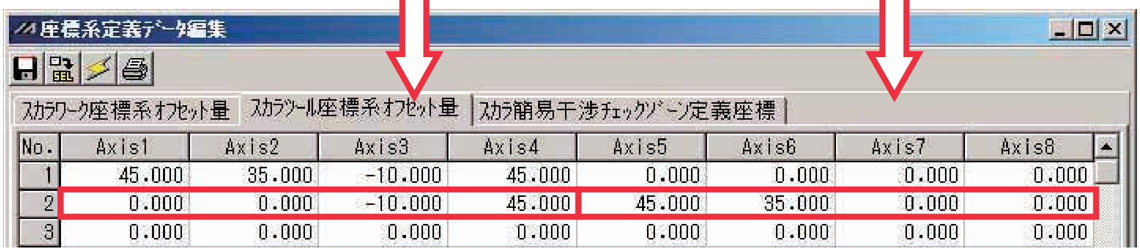
No.(Name)	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
150( )	45.000	35.000	-10.000	45.000				
151( )								
152( )								



No.	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
1	45.000	35.000	-10.000	45.000	45.000	35.000	-10.000	45.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

[Example 2] DFTL 2 152
In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 8 in Tool Coordinate System No. 2 as the position data in either of Axis 1 to 4 or Axis 5 to 8 for the SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) is set effective.
However, 0 will be set to Axis 1 to 2 and 7 to 8 that the position data is the invalid axes.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
150( )	45.000	35.000	-10.000	45.000				
151( )								
152( )			-10.000	45.000	45.000	35.000		



No.	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
1	45.000	35.000	-10.000	45.000	0.000	0.000	0.000	0.000
2	0.000	0.000	-10.000	45.000	45.000	35.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

● SLTL (Dedicated SCARA command/Select tool coordinate system)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SLTL	Tool coordinate system number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the tool coordinate system selection number in operand 1.

(Note 1) The tool/work coordinate systems are functions available for SCARA.

(Note 2) The selected number last declared in the system becomes effective.
The selected tool coordinate system number will remain effective even after the program ends, and also after the power is reconnected if the system-memory backup battery is installed (Note 6).

(Note 3) Only one tool coordinate system selection number is present within the system.

(Note 4) Expressly declare SLTL in the program to prevent unwanted problems resulting from forgetting to reset the coordinate system selection number after changing it in the PC software or on the teaching pendant.
(Execute SLTL 0, if the tool coordinate system is not used.)

(Note 5) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 6) XSEL-RX/SX/RXD/SXD save the tool coordinate system numbers without using a battery.

[Example 1]	GRP	1111	It makes the 1st to 4th axes effective.
	SLTL	1	Selected tool coordinate system of the SCARA axes (1st to 4th axes) is changed to No. 1.
[Example 2]	GRP	11111111	It makes the 1st to 8th axes effective.
	SLTL	2	Selected tool coordinate system of the SCARA axes (1st to 4th axes) and the SCARA axes (5th to 8th axes) is changed to No. 2.

● GTTL (Dedicated SCARA command/Get tool coordinate system definition data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTTL	Tool coordinate system number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the tool coordinate system offset data specified in operand 1 for the position data specified in operand 2. Tool coordinate system offset data for all SCARA axes is set for the position data.
In MSEL-PCX/PGX, tool coordinate system offset data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the tool coordinate system is acquired from is used as the dedicated data for the tool coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the tool coordinate system R-axis offset in the position data is written by execution of GTTL Command.

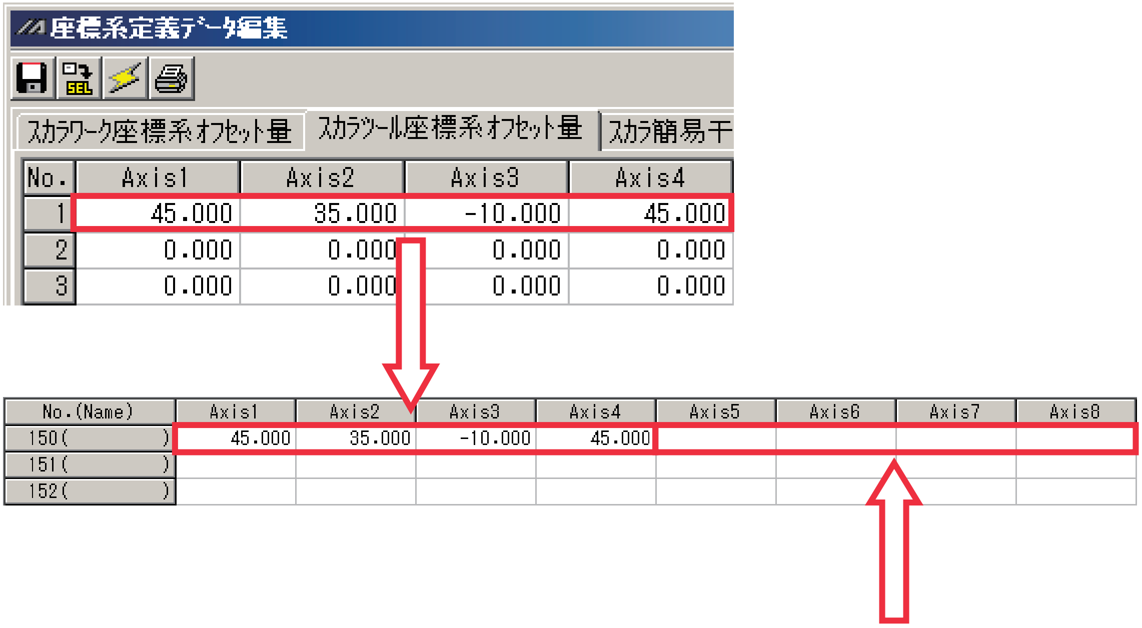
(Note 1) The tool/work coordinate systems are functions available for SCARA.

(Note 2) The position data for the liner axes (5th to 8th axes) are cleared when the command is executed.

(Note 3) Since tool coordinate system No. 0 is reserved by the system as a condition specifying no tool offset, selecting this number generates “Error No. B71: Coordinate system number error”.

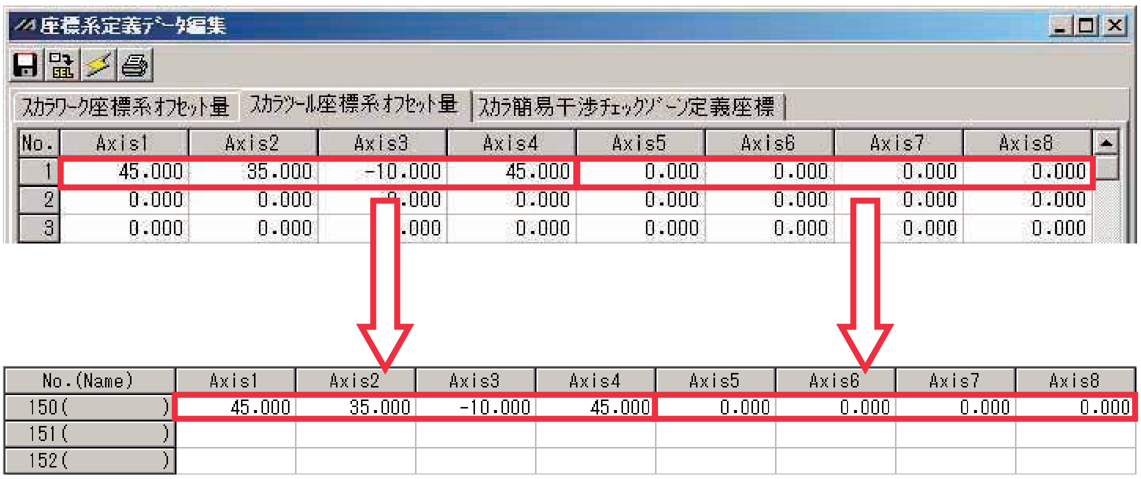
(Note 4) The GRP command is invalid with respect to this command.

[XSEL-JX/KX/PX/QX/RX/SX: 1 unit of SCARA connected]
[Example] GTTL 1 150
After the command shown above is executed, the position data for the liner axes (5th to 8th axes) are cleared.



The data before GTTL Command was executed gets cleared.

[XSEL-RXD/SXD: 2 unit of SCARA connected]
[Example] GTTL 1 150



● DFWK (Dedicated SCARA command/Define work coordinate system)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DFWK	Work coordinate system number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

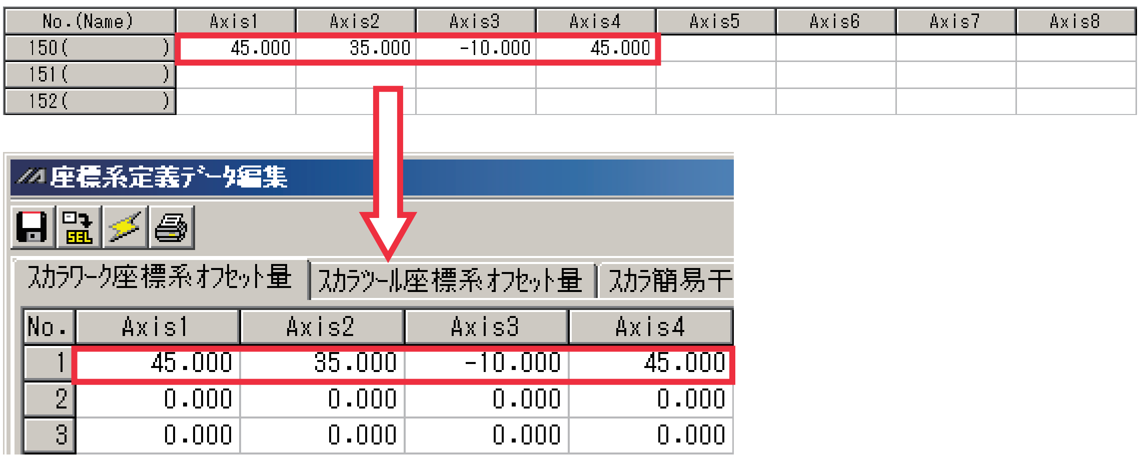
[Function] Set the position data in operand 2 for the work coordinate system offset data specified in operand 1. The position data for all the axes go into the work coordinate system offset data, however, 0 will be set for an axis that the position data is invalid.
In XSEL-RX/SX/RXD/SXD, if all the position data for the SCARA axes in one unit are invalid, data cannot be established in the tool coordinate system offset, and the data before executing DFWK Command is saved
In MSEL-PCX/PGX, position data for four axes is set in the work coordinate system offset data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the work coordinate system data is set from is used as the dedicated data for the work coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the tool coordinate system in DFWK is not a target position for the additional linear axis.

(Note 1) The tool/work coordinate systems are functions available for SCARA.

(Note 2) Since work coordinate system No. 0 is reserved by the system as the base coordinate system, selecting this number generates “Error No. B71: Coordinate system number error”.

(Note 3) The GRP command is invalid with respect to this command.

[XSEL-JX/KX/PX/QX/RX/SX: 1 unit of SCARA connected]
[Example] DFWK 1 150

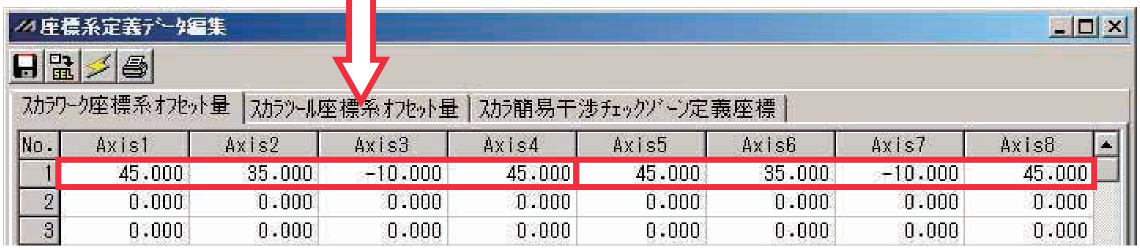


[XSEL-RXD/SXD: 2 unit of SCARA connected]

[Example 1]　　DFWK　1　　150

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 4 in Work Coordinate System No. 1 as the position data in Axis 1 to 4 for the SCARA axes (1st to 4th axes) are set effective. There will be no change to Axis 5 to 8 in Work Coordinate System No. 1 as the position data in Axis 5 to 8 for the SCARA axes (5th to 8th axes) are all set ineffective.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
150( )	45.000	35.000	-10.000	45.000				
151( )								
152( )								



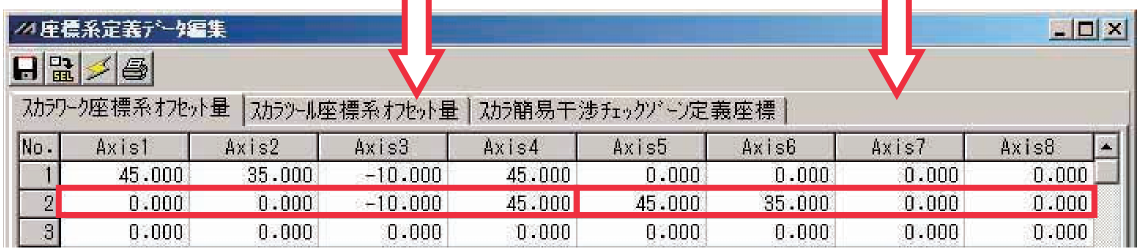
No.	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
1	45.000	35.000	-10.000	45.000	45.000	35.000	-10.000	45.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

[Example 2]　　DFWK　2　　152

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 8 in Work Coordinate System No. 2 as the position data in either of Axis 1 to 4 or Axis 5 to 8 for the SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) is set effective.

However, 0 will be set to Axis 1 to 2 and 7 to 8 that the position data is the invalid axes.

No.(Name)	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
150( )	45.000	35.000	-10.000	45.000				
151( )								
152( )			-10.000	45.000	45.000	35.000		



No.	Axis1	Axis2	Axis3	Axis4	Axis5	Axis6	Axis7	Axis8
1	45.000	35.000	-10.000	45.000	0.000	0.000	0.000	0.000
2	0.000	0.000	-10.000	45.000	45.000	35.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

● SLWK (Dedicated SCARA command/Select work coordinate system)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SLWK	Work coordinate system number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the work coordinate system selection number in operand 1.

(Note 1) The tool/work coordinate systems are functions available for SCARA.

(Note 2) The selected number last declared in the system becomes effective.
The selected work coordinate system number will remain effective even after the program ends, and also after the power is reconnected if the system-memory backup battery is installed (Note 6).

(Note 3) Only one work coordinate system selection number is present within the system.

(Note 4) Expressly declare SLWK in the program to prevent unwanted problems resulting from forgetting to reset the coordinate system selection number after changing it in the PC software or on the teaching pendant.
(Execute SLWK 0, if the work coordinate system is not used.)

(Note 5) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 6) XSEL-RX/SX/RXD/SXD save the tool coordinate system numbers without using a battery.

[Example 1]

GRP	1111	It makes the 1st to 4th axes effective.
SLWK	1	Selected work coordinate system of the SCARA axes (1st to 4th axes) is changed to No. 1.

[Example 2]

GRP	11111111	It makes the 1st to 8th axes effective.
SLWK	2	Selected work coordinate system of the SCARA axes (1st to 4th axes) and the SCARA axes (5th to 8th axes) is changed to No. 1.

● GTWK (Dedicated SCARA command/Get work coordinate system definition number)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTWK	Work coordinate system number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the work coordinate system offset data specified in operand 1 for the position data specified in operand 2. Work coordinate system offset data for all axes is set for the position data.
In MSEL-PCX/PGX, work coordinate system offset data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the work coordinate system is acquired from is used as the dedicated data for the work coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the work coordinate system R-axis offset in the position data is written by execution of GTWK Command.

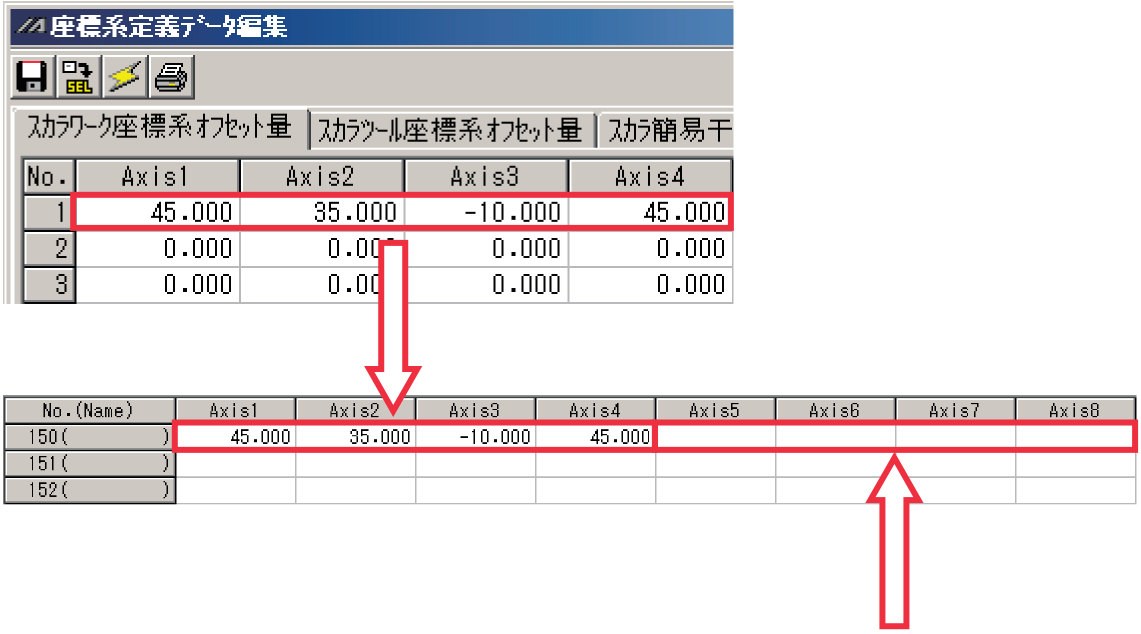
(Note 1) The tool/work coordinate systems are functions available for SCARA.

(Note 2) The position data for the liner axes (5th to 8th axes) are cleared when the command is executed.

(Note 3) Since work coordinate system No. 0 is reserved by the system as the base coordinate system, selecting this number generates “Error No. B71: Coordinate system number error”.

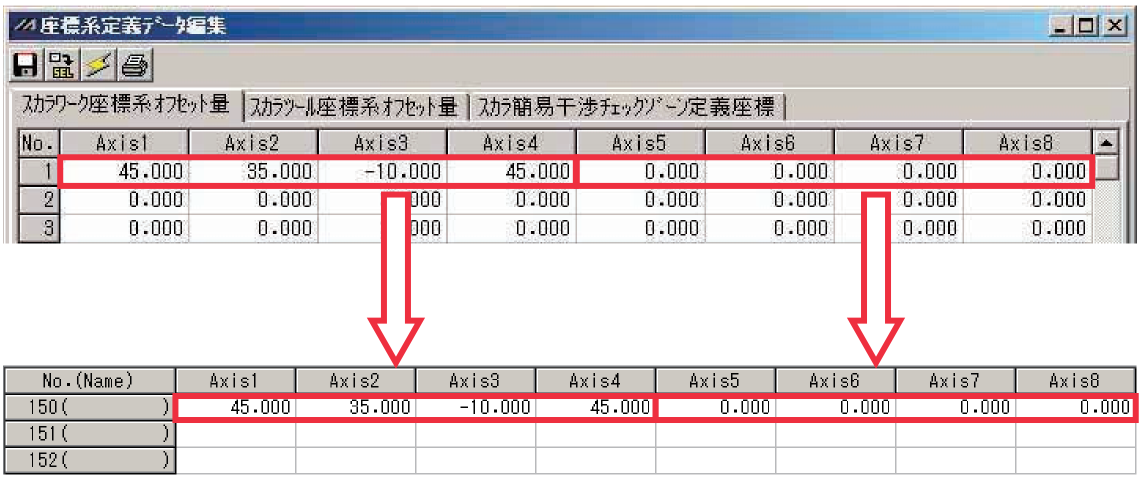
(Note 4) The GRP command is invalid with respect to this command.

[XSEL-JX/KX/PX/QX/RX/SX: 1 unit of SCARA connected]
[Example]   GTWK   1   150



The data before GTWK Command was executed gets cleared.

[XSEL-RXD/SXD: 2 unit of SCARA connected]
[Example]   GTWK   1   150



● RIGH (Dedicated SCARA command/Change current arm system to right arm (arm 2 operation involved if current arm system is opposite))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RIGH	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Change the current SCARA arm system to the right arm system. If the current arm system is the left arm system, arm 2 is moved to change it to the right arm system. After the operation, arms 1 and 2 form a straight line. No arm operation is performed if the current arm system is the right arm system.

(Note 1) To use a RIGH or LEFT command, the speed must be set with VELS even when a SCARA PTP operation command is not used.

(Note 2) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]	GRP	1111	It makes the 1st to 4th axes effective.
	RIGH		The current arm system of the SCARA axes (1st to 4th axes) is changed to the right arm system.
[Example 2]	GRP	11111111	It makes the 1st to 8th axes effective.
	RIGH		The current arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is changed to the right arm system.

● LEFT (Dedicated SCARA command/Change current arm system to left arm (arm 2 operation involved if current arm system is opposite))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	LEFT	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Change the current SCARA arm system to the left arm system. If the current arm system is the right arm system, arm 2 is moved to change it to the left arm system. After the operation, arms 1 and 2 form a straight line. No arm operation is performed if the current arm system is the left arm system.

(Note 1) To use a RIGH or LEFT command, the speed must be set with VELS even when a SCARA PTP operation command is not used.

(Note 2) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 "Axis Pattern Error" will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]	GRP	1111	It makes the 1st to 4th axes effective
	LEFT		The current arm system of the SCARA axes (1st to 4th axes) is changed to the right arm system.
[Example 2]	GRP	11111111	It makes the 1st to 8th axes effective.
	LEFT		The current arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is changed to the left arm system.

● PTPR (Dedicated SCARA command/Specify PTP target arm system to right arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTPR	Prohibited	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Specify the target arm system for SCARA PTP operation commands to the right arm system. After the PTPR command is executed, the target arm system for SCARA PTP operation commands becomes the right arm system and an error occurs if the target value cannot be achieved by operating on the right arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

GRP	1111	It makes the 1st to 4th axes effective.
PTPR		PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the right arm system.
MOVP	1	Move to Position No. 1 to become the right arm system.

[Example 2]

GRP	11111111	It makes the 1st to 8th axes effective.
PTPR		PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the right arm system.
MOVP	2	Move to Position No. 2 to become the right arm system.

● PTPL (Dedicated SCARA command/Specify PTP target arm system to left arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTPL	Prohibited	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Specify the target arm system for SCARA PTP operation commands to the left arm system. After the PTPL command is executed, the target arm system for SCARA PTP operation commands becomes the left arm system and an error occurs if the target value cannot be achieved by operating on the left arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]	GRP	1111	It makes the 1st to 4th axes effective.
	PTPL		PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the left arm system.
	MOVP	1	Move to Position No. 1 to become the left arm system.
[Example 2]	GRP	11111111	It makes the 1st to 8th axes effective.
	PTPL		PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the left arm system.
	MOVP	2	Move to Position No. 2 to become the left arm system.

● PTPD (Dedicated SCARA command/Specify PTP target arm system to current arm (Movement of opposite arm system permitted (no operation is performed) if target unachievable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTPD	Prohibited	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Specify the target arm system for SCARA PTP operation commands to the current arm system. After the PTPD command is executed, the target arm system for SCARA PTP operation commands becomes the current arm system and an error occurs if the target value cannot be achieved by operating on this arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

GRP	1111	It makes the 1st to 4th axes effective.
PTPD		PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the reversed arm system move prohibited when it is not possible.
MOVP	1	Movement is made to Position No. 1 with the current arm system. (“C73: Target Track Software Limit Excess Error” will occur when positioning cannot be performed without changing to the reversed arm system.)

[Example 2]

GRP	11111111	It makes the 1st to 8th axes effective.
PTPR		PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the reversed arm system move prohibited when it is not possible.
MOVP	2	Movement is made to Position No. 2 with the current arm system. (“C73: Target Track Software Limit Excess Error” will occur when positioning cannot be performed without changing to the reversed arm system.)

● PTPE (Dedicated SCARA command/Specify PTP target arm system to current arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTPE	Prohibited	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Specify the target arm system for SCARA PTP operation commands to the current arm system. After the PTPE command is executed, the target arm system for SCARA PTP operation commands becomes the current arm system and if the target value cannot be achieved by operating on this arm system, the target arm system is changed to the one opposite the current arm system. An error occurs if the target value cannot be achieved by operating on either the right arm system or left arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

GRP	1111	It makes the 1st to 4th axes effective.
PTPE		PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the reversed arm system move permitted when it is not possible.
MOVP	1	Movement is made to Position No. 1 with the current arm system. (Positioning is performed with the reversed arm system when positioning cannot be performed without changing to the reversed arm system.)

[Example 2]

GRP	11111111	It makes the 1st to 8th axes effective.
PTPE		PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the reversed arm system move permitted when it is not possible.
MOVP	2	Movement is made to Position No. 2 with the current arm system. (Positioning is performed with the reversed arm system when positioning cannot be performed without changing to the reversed arm system.)

● DFIF (Dedicated SCARA command/Define simple contact check zone coordinate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DFIF	Contact check zone number	Position number (2 successive positions are used)	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the data of two successive positions starting from the position number specified in operand 2, for the simple contact check zone definition coordinate data in operand 1.
The position data specified in operand 2 is set for simple contact check zone definition coordinate 1, while the data of the next position is set for definition coordinate 2. If the axis pattern does not match between the data of the two successive positions, "Error No. C30: Axis pattern error" occurs.
In MSEL-PCX/PGX, position data for four axes is set in the simple contact check zone coordinate data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the simple contact check zone coordinate data is set from is used as the dedicated data for the simple contact check zone coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the simple contact check zone coordinate in DFIF is not a target position for the additional linear axis.

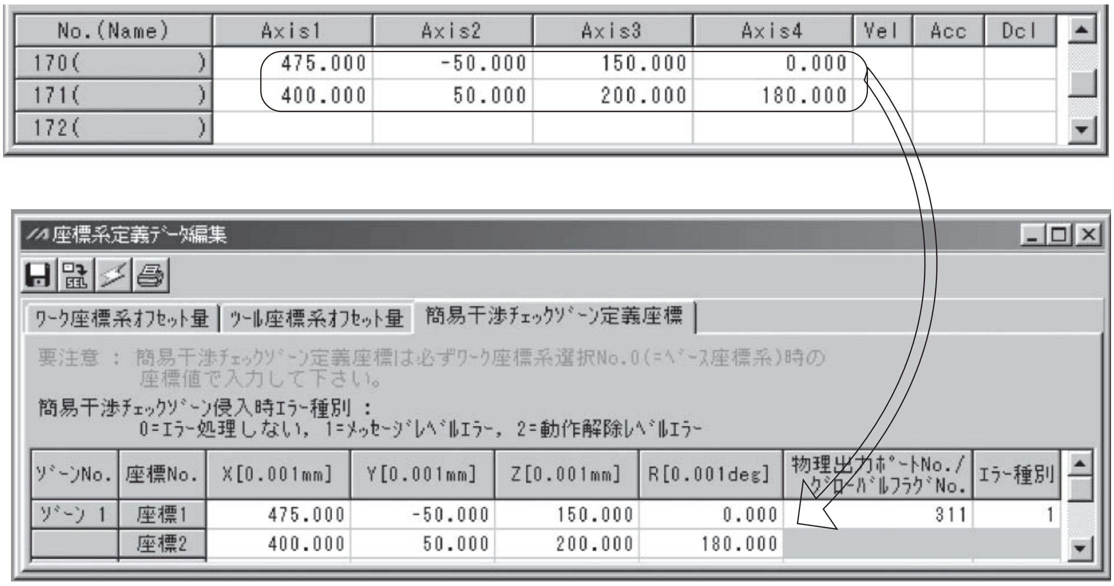
(Note 1) Simple contact check zone definition coordinates are always recognized as data on the base coordinate system (work coordinate system selection No. 0). If you are setting aside position data for use as effective definition coordinates for the DFIF command, you must set the data on the base coordinate system.

(Note 2) When the simple contact check zone definition coordinates are changed, it takes 5msec for the check result based on the new settings to be reflected.

(Note 3) The GRP command is invalid with respect to this command.

(Note 4) Indicate the position data effective either on SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) for the valid axes of the position data. "Error No. C30: Axis Pattern Error" will be issued when both of SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) are set effective for the valid axes of the position data.

[Example] DFIF 1 170



[11] Actuator Control Declaration

● SOIF (Dedicated SCARA command/Specify output for simple contact check zone)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SOIF	Contact check zone number	Output/ global flag number	CP

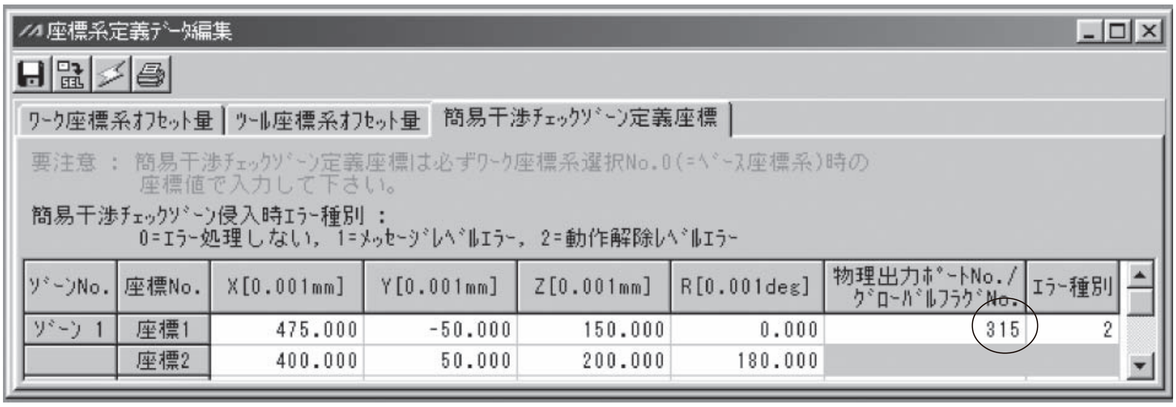
Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the output number/global flag number in operand 2 as the output specification to be applied upon entry into the simple contact check zone specified in operand 1.

(Note 1) The simple contact check zone is a function available for SCARA.

(Note 2) If duplicate physical output port numbers/global flag numbers are specified, chattering occurs and operation results become indeterminable.

[Example] SOIF 1 315



● SEIF (Dedicated SCARA command/Specify type of simple contact check zone)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SEIF	Contact check zone number	0 or 1 or 2 (error type)	CP

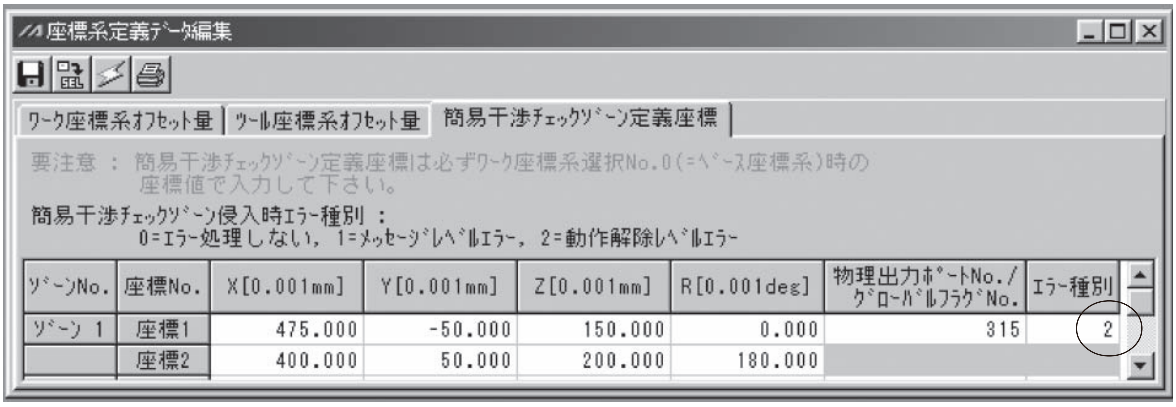
Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Specify the error type in operand 2 (see below) as the error type to be applied upon entry into the simple contact check zone specified in operand 1.

Error types applicable upon entry into simple contact check zone
0: No error
1: Message level error
2: Operation-cancellation level error

(Note 1) The simple contact check zone is a function available for SCARA.

[Example] SEIF 1 2



● GTIF (Dedicated SCARA command/Get simple contact check zone definition coordinate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTIF	Contact check zone number	Position number (2 successive positions are used)	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Set the simple contact check zone definition coordinate data in operand 1 for the data of two successive positions starting from the position number specified in operand 2.
Simple contact check zone definition coordinate 1 is set for the position data specified in operand 2, while definition coordinate 2 is set for the data of the next position. At this time, coordinate data in the position data becomes invalid for all axes, and then the simple contact check zone definition coordinate data is set.
In MSEL-PCX/PGX, simple contact check zone coordinate data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the simple contact check zone coordinate is acquired from is used as the dedicated data for the simple contact check zone coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the simple contact check zone coordinate R-axis offset in the position data is written by execution of GTIF Command.

(Note 1) The position data of the invalid SCARA axes in the liner axes or the simple interference check zone definition coordinate data is cleared when the command is executed.

(Note 2) Simple contact check zone definition coordinate is always recognized as data on the base coordinate system (work coordinate system selection No. 0). Accordingly, the position data set by a GTIF command must be handled on the base coordinate system.

(Note 3) The GRP command is invalid with respect to this command.

● WGHT (Dedicated SCARA command/Set tip load mass, inertial moment)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WGHT	Mass	(Inertial moment)	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	×	○	○	×	×	○ (PCX/PGX only)

This command is supported by controller main application Ver.0.45 or later.
It is supported by PC software of Ver.7.5.0.0 or later and teaching pendants of Ver.1.11 or later.

(Note) Conventional models such as IX-NNN5020 cannot use this command. (A “D8A: Internal parameter error of acceleration/deceleration optimization or horizontal movement Z-position optimization function” occurs.)

[Function] Set the mass and inertial moment of the tip load (tool + work).
Set the mass in operand 1, and inertial moment in operand 2. The unit of operand 1 is [g], while the unit of operand 2 is [kg•mm^2].
The tip load mass/inertial moment set by a WGHT command will be retained until a new WGHT command is set again (= the set values will be retained even after the program ends). However, they are cleared when the power is turned OFF or a software reset is performed, after which you must set the applicable values again expressly in the program.

(Note 1) For the inertial moment in operand 2, set a composite inertial moment covering the tool and work relating to the center of rotation of the R-axis.

(Note 2) Although entry of inertial moment in operand 2 is optional, if no inertial moment is set the maximum allowable inertial moment of the robot is set automatically.

(Note 3) If the tip load mass exceeds the maximum loading capacity of the robot, a “B44: Load mass setting error” occurs.

(Note 4) Executing a WGHT command updates the information of both the tip load mass and inertial moment. You cannot change only the mass or only the inertial moment.

(Note 5) Although both the tip load mass and inertial moment can be approximate values, set values slightly larger than necessary. Before setting the values, round them up to the nearest multiple of 1g or 1kg•mm^2, respectively.

(Note 6) If a WGHT command has not yet been executed, the load mass and inertial moment have been initialized to the maximum loading capacity and maximum allowable inertial moment of the robot. Set an appropriate load mass and inertial moment according to the use conditions.

(Note 7) The load mass and inertial moment set by a WGHT command are used in the SCARA PTP acceleration/deceleration optimization function, SCARA horizontal movement Z-position optimization function, etc.

(Note 8) In XSEL-RX/SX/RXD/SXD, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]	GRP	1111		It makes the 1st to 4th axes effective.
	WGHT	2000	5000	Set a tip load with 2000g of weight and 5000kgmm2 of the moment of inertia to the SCARA axes (1st to 4th axes).
[Example 2]	GRP	11111111		It makes the 1st to 8th axes effective.
	WGHT	1000	30000	Set SCARA axes (1st to 4th axes9 and SCARA axes (5th to 8th axes) so the movement is made to Position No. 2 to make it become the right arm system.

● WGT2 (Dedicated SCARA command/Tip load condition setting)

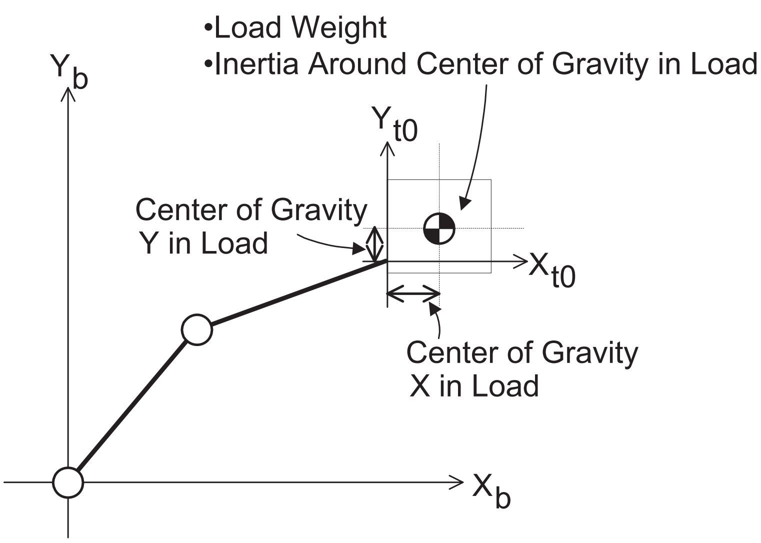
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WGT2	Mass	(Variable No.)	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	×	×	○	×	×	×

[Function] Set the weight [gr] of the load on the tip (tool + work piece) in Operation 1, and the center of gravity in the load, inertia around the center of gravity and other related parameters to the six variables in a row from the indicated variable in Operation 2.
The tip load mass/inertial moment set by a WGT2 command will be retained until a new WGT2 command is set again. However, they are cleared when the power is turned OFF or a software reset is performed, after which you must set the applicable values again expressly in the program.

• Indicated Variable in Operation 2

Variable No.	Contents of Setting	Remarks
n	Gravity Center X for Tip Load [1/1000mm unit]	Input the position at Tool Coordinate No. 0.
n+1	Gravity Center Y for Tip Load [1/1000mm unit]	
n+2	Inertia around Center of Gravity in Top Load [kgmm^2]	
n+3	Set to 0	Reservation (∗Possibility of use in future)
n+4	Set to 0	Reservation (∗Possibility of use in future)
n+5	Set to 0	Reservation (∗Possibility of use in future)



(Note 1) Inputting in Operation 2 is optional. When the setting in Operation 2 is not established, the parameters are the center of gravity in tip load X-Y = 0 and the maximum allowable moment of inertia.

(Note 2) An error will be issued when the tip load weight exceeds the maximum transportable weight of the robot.

(Note 3) When WGT2 Command is executed, the information for both the tip load weight and the moment of inertia is updated.
A change to individuals such as the weight only or center of gravity in tip load and inertia around the center of gravity only is not available.

(Note 4) Inappropriate setting of the robot tip load condition may cause vibration (abnormal noise) or error, and also may give an impact that shortens the mechanical life.
Establish the setting that reflects the actual mounted load.

(Note 5) For XSEL-RX/SX/RXD/SDX, GRP/BASE Commands become effective even in WGT2 Command. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands. When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]	GRP	1111		Indicates SCARA of 1st to 4th axes
	LET	1001	50000	Indicates Center of gravity X in tip load = 50.000mm
	LET	1002	0	Indicates Center of gravity Y in tip load = 0.000mm
	LET	1003	2000	Indicates inertia around center of gravity in load = 2000kg•mm^2
	WGT2	1000	1001	For SCARA of 1st to 4th axes sets weight of 1000g and conditions of the tip load for Variable No. 1001 to 1003

[Example 2]	GRP	11110000		Indicates SCARA of 5th to 8th axes
	LET	1001	20000	Indicates Center of gravity X in tip load = 20.000mm
	LET	1002	20000	Indicates Center of gravity Y in tip load = 20.000mm
	LET	1003	500	Indicates inertia around center of gravity in load = 500kg•mm^2
	WGT2	500	1001	SCARA of 5th to 8th axes sets weight of 500g and conditions of the tip load for Variable No. 1001 to 1003

● NBND (Dedicated linear axis command/Set close distance)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	NBND	Axis pattern	Close distance	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
○	×	×	×	×	×	×	×	×

[Function] Set in operand 2 the close distance (mm) from the target position based on the axis pattern specified in operand 1.
This command is valid only with respect to MOVP, MOVL and PATH commands in quick return mode 2 (closeness-detection return target position addition mode) or quick return mode 3 (closeness-detection return target position change mode).
A different value can be set for each axis.

(Note 1) The default value of 0 is applied if the close distance is not set with a NBND command.

(Note 2) In the case of PATH commands involving successive movements to multiple positions, the close distance becomes effective after the movement to the last position in the last movement is started and also after the processing of the previous position movement is completed. Accordingly, a dead width is created between (= at the overlap of) the movement to the last position in the PATH commands and the movement to the position immediately before it.

(Note 3) The close distance set here will remain effective even after the program ends. When building a system using NBND commands, therefore, specify the close distance expressly with a NBND command in all programs before any operation is started in each program. If you assume that the close distance will be reset after the end of operation in other programs, an unexpected close distance may be applied should the program abort due to an error, etc., in which case unforeseen problems may result.

(Note 4) Be sure to also refer to the pages that explain the QRTN command and PEND command.

(Note 5) Software versions supporting NBND
Controller main application: Ver.1.04 or later
(excluding flash ROM 8Mbit versions)
PC software: Ver.7.2.3.0 or later
Teaching pendant:
IA-T-X (D): Ver.1.44 or later
SEL-T (D): Ver.1.02 or later

[Example 1]

```
NBND    11    50    Set the close distance for axes 1 and 2 to 50mm
                    after this command.
```

[Example 2] The axis pattern can be specified indirectly using a variable. [Example 1] can be rephrased using indirect specification by variable as follows:
11 (binary) → 3 (decimal)

```
LET     1     3     Assign 3 to variable 1.
NBND    *1    50    Set the close distance for axes 1 and 2 to 50mm
                    after this command.
```

[12] Actuator Control Command

● SV☐☐ (Turn ON/OFF servo)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SV☐☐	Axis pattern	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Turn ON/OFF the servos of the axes specified by the axis pattern in operand 1.

SV☐☐
- ON .... Turn ON the servo.
- OF .... Turn OFF the servo.

(Other than SCARA robot)

[Example 1] SVON 11 — Turn ON the servos of axes 1 and 2. Nothing will occur if the axis servos are already ON.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 1 3 — Assign 3 to variable 1.
SVON *1

(SCARA robots)

The arm system of SCARA axes (1st to 4th axes or 1st to 3rd axes) is set to Local Variable No. 99 when complete in normal condition.

Right arm system = 1
Left arm system = -1
Indeterminable = 0

The angle of arm 2 is used to make judgment.
The arm system effective immediately after the servo ON is set. The arm system is not monitored continuously.

(Note) The arm system data set in Local Variable No. 99 is the arm system for SCARA axes (1st to 4th axes or 1st to 3rd axes). To acquire the arm system data for SCARA axes (5th to 8th axes), use GARM Command.

● HOME (Dedicated linear axis command/Home return)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	HOME	Axis pattern	Prohibited	PE

Applicable models
XSEL-JX/KX ×   Other than XSEL-JX/KX ○

[Function] Perform home return of the axes specified by the axis pattern in operand 1.
The servo of each home-return axis will turn ON automatically.
The output will turn OFF at the start of home return, and turn ON when the home return is completed.

(Note 1) This is a dedicated command for linear axes. If a SCARA axis (except for IXP Type Incremental specification) is specified, “Error No. B80: Specification-prohibited axis error” or “Error No. 421: SCARA/linear-axis simultaneous specification error” occurs.

(Note 2) Following a pause of home return, the operation will resume from the beginning of the home-return sequence.

(Note 3) Home-return operation for the axis using an ABS encoder makes a movement to the multi-rotation data reset position, thus it does not always make a movement to the home preset coordinate (including 0).
Use a MOVP command, instead of a HOME command, if you want to turn ON output 304 when I/O parameter No. 50, “Output function selection 304” is set to 1 (Output when all effective linear axes are home (= 0)) or 3 (Output when all effective linear axes are at home preset coordinate).

(Note 4) If an operation pause or cancel is performed during the HOME Command is executed for the axis using an ABS encoder other than the absolute reset mode provided by the PC software or teaching pendant, it may cause the “actual-position soft limit error” due to the position.
It is not recommended to perform home return other than for the purpose of adjusting an absolute-encoder axis.

[Example 1] HOME 11 Axes 1 and 2 return to the home.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    1    3        Assign 3 to variable 1.
HOME   *1
```

● MOVP (Move PTP by specifying position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MOVP	Position number	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move the actuator to the position corresponding to the position number specified in operand 1, without interpolation (PTP stands for “Point-to-Point”).
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

[Example 1]

VEL	100		Set the speed to 100mm/s.
MOVP	1		Move the axes to the position corresponding to position No. 1 (200, 100).

[Example 2]

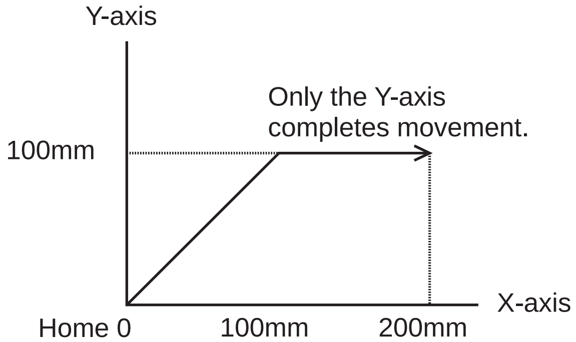
VEL	100		Set the speed to 100mm/s.
LET	1	2	Assign 2 to variable 1.
MOVP	*1		Move the axes to the position corresponding to the content of variable 1 (position No. 2, or (100, 100)).

Position Data Display in PC Software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	200.000	100.000			
2	100.000	100.000			

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration”.

Travel path from the home to the position corresponding to position No. 1 (200, 100)



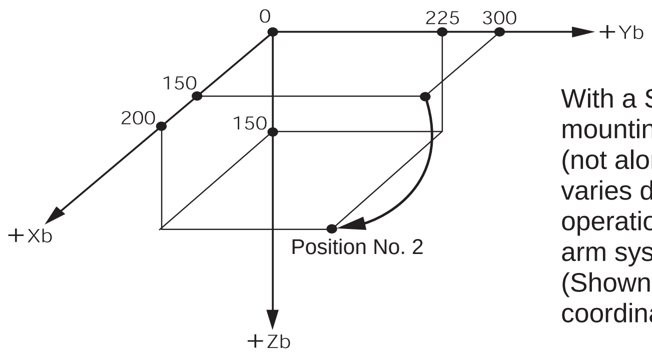
Each axis moves at 100mm/s.

(SCARA robots)
[Example 1] MOVP 2 Move the axes to the positions set under position No. 2 (200, 225, 150, 30).

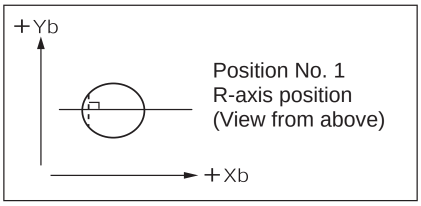
No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
1( )	150.000	300.000	0.000	0.000			
2( )	200.000	225.000	150.000	30.000			
3( )							
4( )							

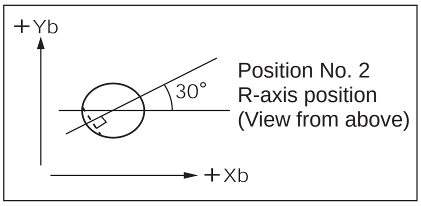
(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 47, “Default PTP acceleration for SCARA axis” or all-axis parameter No. 48, “Default PTP deceleration for SCARA axis” if the acceleration/deceleration is not specified using an ACCS (DCLS) command. In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.

Path of moving from position No. 1 to position No. 2



With a SCARA axis, the center of the tool mounting surface or tool tip moves via PTP (not along a straight line). The moving path varies depending on the start position of operation, completion position of operation, arm system, etc.
(Shown to the left are positions on the base coordinate system.)





● MOVL (Move by specifying position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MOVL	Position number	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move the actuator to the position corresponding to the position number specified in operand 1, with interpolation.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

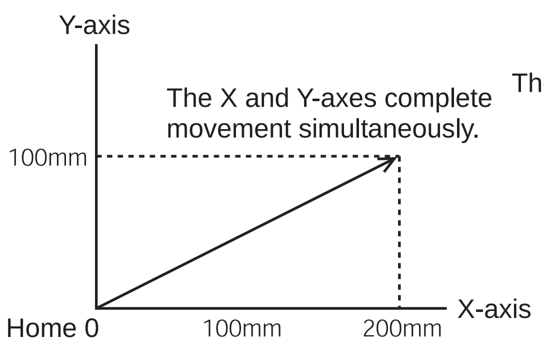
[Example 1]	VEL	100		Set the speed to 100mm/s.
	MOVL	1		Move the axes to the position corresponding to position No. 1 (200, 100), with interpolation.
[Example 2]	VEL	100		Set the speed to 100mm/s.
	LET	1	2	Assign 2 to variable 1.
	MOVL	*1		Move the axes to the position corresponding to the content of variable 1 (position No. 2, or (100, 100)), with interpolation.

Position Data Display in PC Software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	200.000	100.000			
2	100.000	100.000			

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration”.

Travel path from the home to the position corresponding to position No. 1 (200, 100)



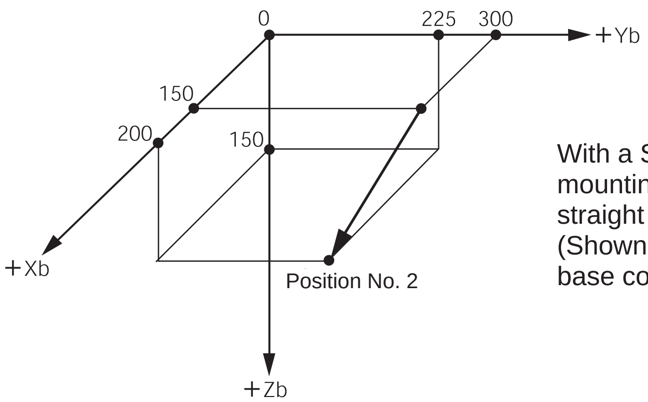
The tip of each axis moves at 100mm/s.

(SCARA robots)
[Example 1]   MOVL   2   Move the axes to the positions set under position No. 2 (200, 225, 150, 30) via interpolation.

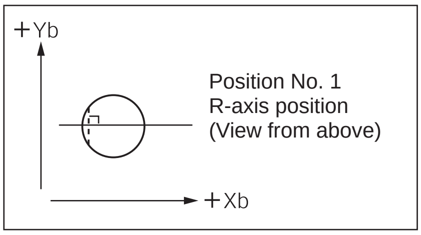
Path of moving from position No. 1 to position No. 2

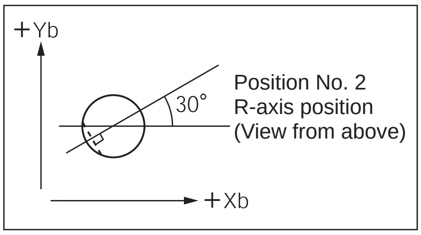
No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
1( )	150.000	300.000	0.000	0.000			
2( )	200.000	225.000	150.000	30.000			
3( )							
4( )							

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 11, “Default CP acceleration for SCARA axis” or all-axis parameter No. 12, “Default CP deceleration for SCARA axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.



With a SCARA axis, the center of the tool mounting surface or tool tip moves along a straight line.
(Shown to the left are positions on the base coordinate system.)





● MVPI (Move via incremental PTP)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MVPI	Position number	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move the actuator, without interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

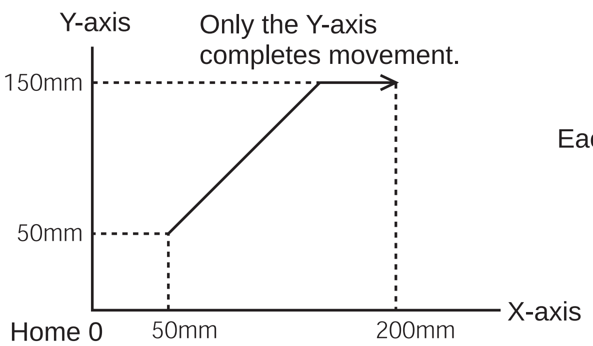
[Example 1]	VEL	100		Set the speed to 100mm/s.
	MVPI	1		If the current position is (50, 50) and position No. 1 is set to (150, 100), the axes will move 150 in the X direction and 100 in the Y direction (200, 150) from the current position.
[Example 2]	VEL	100		Set the speed to 100mm/s.
	LET	1	2	Assign 2 to variable 1.
	MVPI	*1		Move from the current position by the travel distance corresponding to the content of variable 1 (position No. 2, or (100, 100)).

Position Data Display in PC Software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	150.000	100.000			
2	100.000	100.000			

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration”.

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



Each axis moves at 100mm/s.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution (0.001μm/pulse) × 1000
/ (2 ^ Encoder division ratio)

(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution : Axis-specific parameter No. 42
Encoder division ratio : Axis-specific parameter No. 43
Screw lead : Axis-specific parameter No. 47
Gear ratio numerator : Axis-specific parameter No. 50
Gear ratio denominator : Axis-specific parameter No. 51

(SCARA robots)
(Note 1) If an incremental movement command (MVPI, MVLI, TMPI or TMLI) is used repeatedly, coordinate conversion rounding errors, etc., will accumulate. To eliminate these errors, etc., execute an absolute movement command (MOVP, MOVL, etc.) once.

[Example 1] MVPI 6 Move from the current position by the travel according to position No. 6.
If the current positions of the axes are specified by position No. 5 (200, 150, 50, 45) and travels are specified by position No. 6 (15, 30, 20, 30), the axes move to the positions (215, 180, 70, 75).

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
5( )	200.000	150.000	50.000	45.000			
6( )	15.000	30.000	20.000	30.000			
7( )							

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 47, “Default PTP acceleration for SCARA axis” or all-axis parameter No. 48, “Default PTP deceleration for SCARA axis” if the acceleration/deceleration is not specified using an ACCS (DCLS) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.

Caution
A margin of error could accumulate between each pitch if the incremental (relative position indication) movement commands are repeated continuously.
To avoid accumulation of errors, utilize the movement command to indicate the absolute position (MOVP Command).

● MVLI (Move via incremental interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MVLI	Position number	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move the actuator, with interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

[Example 1]

```
VEL    100        Set the speed to 100mm/s.
MVLI   1          If the current position is (50, 50) and position No. 1
                  is set to (150, 100), the axes will move 150 in the X
                  direction and 100 in the Y direction (200, 150) from
                  the current position, with interpolation.
```

[Example 2]

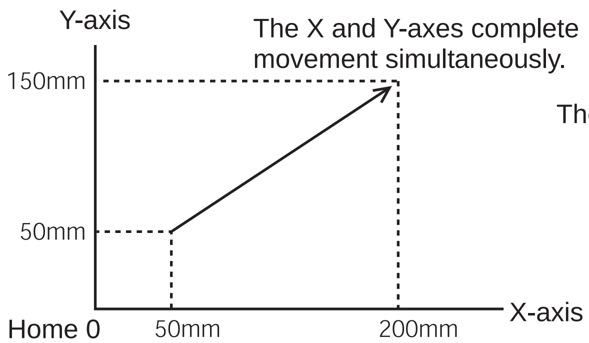
```
VEL    100        Set the speed to 100mm/s.
LET    1    2     Assign 2 to variable 1.
MVLI   *1         Move from the current position by the travel distance
                  corresponding to the content of variable 1 (position
                  No. 2, or (100, 100)).
```

Position Data Display in PC Software

No.	Axis 1 (X-axis)	Axis 2 (Y-axis)	Vel	Acc	Dcl
1	150.000	100.000			
2	100.000	100.000			

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration”.

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



The tip of each axis moves at 100mm/s.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution [0.001μm/pulse] × 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution : Axis-specific parameter No. 42
Encoder division ratio : Axis-specific parameter No. 43
Screw lead : Axis-specific parameter No. 47
Gear ratio numerator : Axis-specific parameter No. 50
Gear ratio denominator : Axis-specific parameter No. 51

(SCARA robots)
(Note 1) If an incremental movement command (MVPI, MVLI, TMPI or TMLI) is used repeatedly, coordinate conversion rounding errors, etc., will accumulate. To eliminate these errors, etc., execute an absolute movement command (MOVP, MOVL, etc.) once.

[Example 1] MVLI 6 Move from the current position by the travel according to position No. 6.
If the current positions of the axes are specified by position No. 5 (200, 150, 50, 45) and travels are specified by position No. 6 (15, 30, 20, 30), the axes move to the positions (215, 180, 70, 75).

No.(Name)	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
5( )	200.000	150.000	50.000	45.000			
6( )	15.000	30.000	20.000	30.000			
7( )							

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 11, “Default CP acceleration for SCARA axis” or all-axis parameter No. 12, “Default CP deceleration for SCARA axis” if the acceleration/deceleration is not specified in the position data or using an ACC (DCL) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data or using an ACC (DCL) command.

Caution
A margin of error could accumulate between each pitch if the incremental (relative position indication) movement commands are repeated continuously.
To avoid accumulation of errors, utilize the movement command to indicate the absolute position (MOVL Command).

● MOVD (Move via direct value specification)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MOVD	Target position	(Axis pattern)	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	×	×	×	○	×	○ (PC/PG only)

[Function] Move the axis specified by the axis pattern in operand 2, to the target position corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The target position is set in mm, and the set value is valid to the third decimal place.

[Example 1] MOVD 100 10 Move axis 2 to position 100.

[Example 2]

LET	1	100	Assign 100 to variable 1.
MOVD	*1	11	Move all axes to the content of variable 1 (100).

● MVDI (Move relatively via direct value specification)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	MVDI	Travel distance	(Axis pattern)	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	×	×	×	○	×	○ (PC/PG only)

[Function] Move the axis specified by the axis pattern in operand 2 from its current position by the travel distance corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The travel distance is set in mm, and the set value is valid to the third decimal place.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution [0.001μm/pulse] × 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution : Axis-specific parameter No. 42
Encoder division ratio : Axis-specific parameter No. 43
Screw lead : Axis-specific parameter No. 47
Gear ratio numerator : Axis-specific parameter No. 50
Gear ratio denominator : Axis-specific parameter No. 51

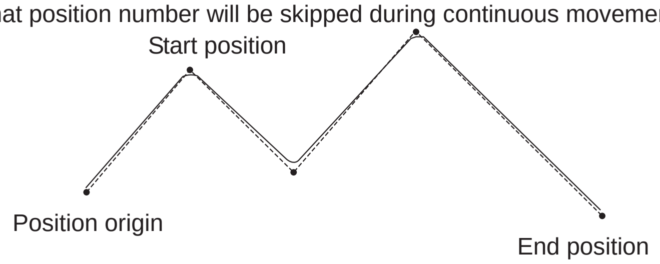
[Example 1]	MVDI	30	11	Move all axes from the current position by 30mm in the positive direction.
[Example 2]	LET	1	-100	Assign -100 to variable 1.
	MVDI	*1	1	Move axis 1 from the current position in accordance with the content of variable 1 (-100), or by 100mm in the negative direction.

● PATH (Move along path via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PATH	Start position number	End position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move continuously from the position specified in operand 1 to the position specified in operand 2.
The output type in the output field can be set using an actuator-declaration command POTP.
Increasing the acceleration will make the passing points closer to the specified positions.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.

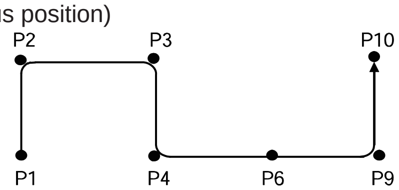


(Note 1) Multi-dimensional movement can be performed using a PATH command.
In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

(Note 2) It is possible to move through discontinuous positions or move continuously by passing the same position.
As shown in the example, specify the number corresponding to the discontinuous position for both the start position number and end position number in the PATH command. In the example, this position is No. 6.

[Example] The actuator moves continuously in the sequence of position No. 1 → 2 → 3 → 4 → 6 → 9 → 10.

PATH	1	4
PATH	6	6 (discontinuous position)
PATH	9	10



(Note 3) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 "SCARA/Linear Drive Axes Double Indication Error")
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example 1] PATH 100 120 Move continuously from position No. 100 to 120.

● J☐W☐ (Jog)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	J☐W☐	Axis pattern	Input/output/ flag number	PE

Applicable models
XSEL-JX/KX ×   Other than XSEL-JX/KX ○

[Function] The axes in the axis pattern specified in operand 1 will move forward or backward while the input or output port or flag specified in operand 2 is ON or OFF.

JBWF .... Move backward while the specified port is OFF.
JBWN.......... Move backward while the specified port is ON.
JFWF .... Move forward while the specified port is OFF.
JFWN.......... Move forward while the specified port is ON.

With SCARA axes, only one axis (X, Y, Z or R-axis) can be specified. If a SCARA axis is specified, operation (CP operation) based on the currently selected work coordinate system is applied. With linear axis, multiple axes can be specified.
(SCARA axes are supported with application Ver.0.33 or later)

(Note 1) In main application Ver.0.33 or older, “Error No. B80: Specification-prohibited axis error” or “Error No. 421: SCARA/linear-axis simultaneous specification error” occurs if a SCARA axis is specified.

(Note 2) With linear axes, this command is also effective on axes that have not yet performed home return. In this case, however, the maximum speed is limited to all-axis parameter No. 15, “Maximum jog speed before coordinate confirmation/home return”. Since the coordinate values have no meaning in this condition, pay due attention to prevent contact with the stroke end. With SCARA axes, operation by this command is disabled only before the confirmation of ABS coordinates.

(Note 3) The jog speed of a SCARA axis is limited by all-axis parameter No. 37, “Maximum SCARA-axis speed under J☐W☐ command” (default: 250mm/sec). This parameter can be edited using PC software Ver.7.0.11.0 or later, teaching pendant (IA-T-X) Ver.1.44 or later or teaching pendant (SEL-T) Ver.1.01 or later. (Although the value set in the above parameter can also be changed using an older PC software or teaching pendant version, the parameter name is not displayed and the set value is indicated in hexadecimal notation.)

(Note 4) Axes other than the Z-axis cannot be operated from other tasks while the X, Y or R-axis is jogging.

(Note 5) If the start position of operation of a SCARA axis is near the point at which arms 1 and 2 form a straight line (singular point), operation is performed at low acceleration to prevent sudden movement.

(Note 6) If the start position of operation of a SCARA axis is outside the work envelope (within the soft limit overt points of each axis, CP operation limit band, tool-reference-point entry prohibition circle (if tool offset is enabled) or back entry prohibition area), select an appropriate axis and direction and move the axis to inside the work envelope. Jogging out of the work envelope is not permitted.

(Note 7) If the R-axis generates “Error No. C74: Actual-position soft limit over error” due to a posture control component, etc., during SCARA-axis jog operation, take an appropriate action, such as bringing the R-axis position closer to the center of the R-axis stroke, using the jog function for each axis in the PC software or on the teaching pendant.

(Note 8) When tool offset for SCARA axis is enabled (tool coordinate system selection number is not 0), jogging of the R-axis involves rotation at the tool tip and thus arms 1 and 2 move. Exercise caution.

(Note 9) If the axis that moves according to J☐W☐ is a linear axis, and also if axis-specific parameter No. 1, “Axis operation type” is set to 0 (Linear movement axis) while axis-specific parameter No. 68, “Linear-axis linear movement mode selection” is set to 1 (Infinite stroke mode*), infinite stroke operation is performed.
During infinite stroke operation, the current position circulates within a range of approx. -10m to 10m.
Any positioning command to a position outside a coordinate range of approx. -9999 to +9990 generates “Error No. CBE: Target-value data boundary over error”.
If a positioning command not meeting the above condition is executed outside a coordinate range of approx. -9990 to +9990, “Error No. CC5: Positioning boundary breakout error” occurs.
(These errors are generated intentionally because the user cannot recognize the operating direction precisely around the boundary. If any of these errors occurs, axis-specific parameter No. 10, “ABS reset position movement/home return method” must be set to 1 (Current position 0 home) and, if necessary, the current value may also have to be reset with a HOME command.)
During infinite stroke operation, be sure to implement a timeout check using other task or external system.

The infinite-stroke mode can be specified only when an incremental encoder is used. If you wish to use the infinite-stroke mode, contact IAI’s Sales Engineering.

[Example 1]	VEL	100		Set the speed to 100mm/s.
	JBWF	10000	10	Move axis 5 backward while input 10 is OFF.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
10000 (binary) → 16 (decimal)

VEL	100		Set the speed to 100mm/s.
LET	1	16	Assign 12 to variable 1.
JBWF	*1	10	

[Example 3]	VEL	100		Set the speed to 100mm/s.
	LET	5	20	Assign 20 to variable 5.
	JFWN	10000	*5	Move axis 5 forward while the content of variable 5 (input 20), is ON.

● STOP (Stop movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	STOP	Axis pattern	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Decelerate and stop the axes specified by the axis pattern in operand 1.

(Note 1) A STOP command can be used with all active servo commands other than a SVOF command.

(Note 2) With a SCARA robots, all axes are decelerated to a stop regardless of the axis pattern.

(Note 3) The STOP command only issues a deceleration stop (operation cancellation) command and the program does not wait for completion of stopping. If other servo command is issued while the axes are stopping, the command becomes invalid or an "axis multiple-use" or other error occurs.
Set a timer, etc., in the program so that the next servo command will be issued after a sufficient deceleration-stop processing time elapses.
Even when a STOP command is to be issued to an axis currently stopped, provide a minimum interval of 0.1sec before the next servo command is issued.

(Other than SCARA robots)
[Example 1] STOP 11 Decelerate and stop axes 1 and 2.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET   1   3     Assign 3 to variable 1.
STOP  *1
```

(SCARA robot)
[Example 1] STOP 1 Decelerate the SCARA axes to a stop.

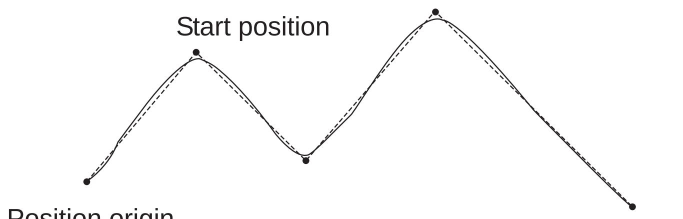
[12] Actuator Control Command

● PSPL (Move along spline via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PSPL	Start position number	End position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Continuously move from the specified start position to end position via interpolation along a spline-interpolation curve.
The output type in the output field can be set using an actuator-declaration command POTP.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.



(The diagram above is an image.)

(Note 1) If the acceleration and deceleration are different between points, the speeds will not be connected smoothly.

In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

(Note 2) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

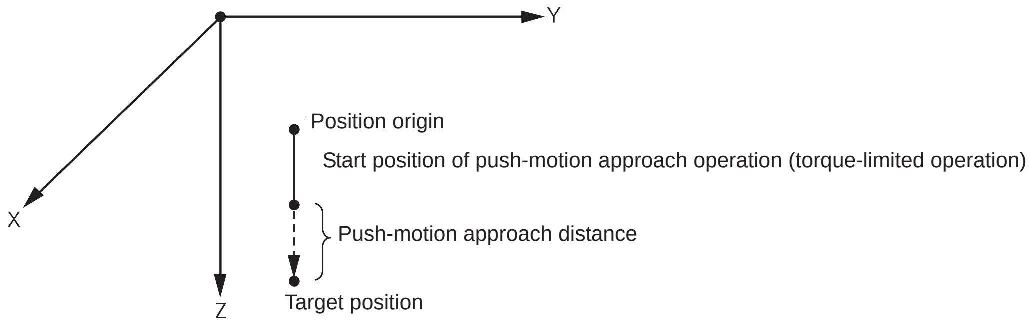
[Example] PSPL 100 120 Continuously move from position Nos. 100 to 120 along a spline-interpolation curve.

● PUSH (Move by push motion)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PUSH	Target position number	Prohibited	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Perform push-motion operation until the target position specified in operand 1 is reached.
The axes move in a normal mode from the position origin to the push-motion approach start position as determined by a PAPR command, after which push-motion approach operation (toque-limiting operation) will be performed. The speed of push-motion approach operation (toque-limiting operation) is determined by the push-motion approach speed specified by a PAPR command. If the output field is specified, the output will turn ON when a contact is confirmed, and turn OFF when a missed contact is detected.
Movement from the position origin to start position of push-motion approach conforms to the speed and acceleration/deceleration specified by VEL/ACC/DCL commands or in the position data table.



The pressing force can be adjusted in Driver Card Parameter No. 38 Limitation for pressing torque in positioning process (default value = 70%) or PTRQ Command.

(Note 1) A PUSH command only moves a single axis. If multiple axes are specified, an "Error No. C91, Multiple push-axes specification error" will generate.

(Note 2) A push-motion approach speed exceeding the maximum speed permitted by the system will be clamped at the maximum speed.
(The maximum system speed is not the maximum practical speed. Determine a practical speed by considering the impact upon contact, etc.)

[Example]

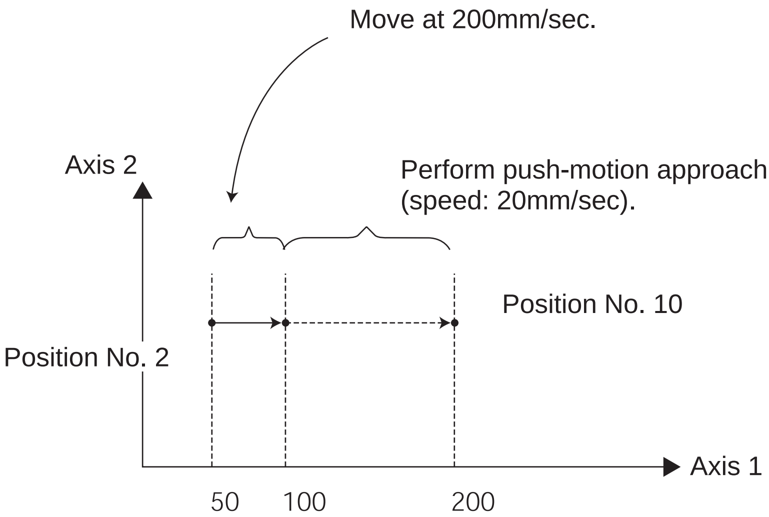
```
PAPR    100    20
MOVP    2
PUSH    10
```

Set the push-motion approach distance to 100mm and push-motion approach speed to 20mm/sec.
Move from the current position to position No. 2.
Perform push-motion movement from position No. 2 to 10.

The diagram below describes a push-motion movement based on the position data shown in the table below:

Position Data Display in PC Software

Position No.	Axis 1	Axis 2	Vel	Acc	Dcl
1					
2	50.000	100.000			
•					
•					
•					
•					
10	200.000		200	0.30	0.30
•					
•					



● PTRQ (Change push torque limit parameter)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTRQ	Axis pattern	Ratio	CC

Applicable models
XSEL-J/K × Other than XSEL-J/K ○

[Function] Change the push torque limit parameter of the axis pattern specified in operand 1 to the value in operand 2. Operand 2 is set as an integer (unit: %).
A PTRQ command temporarily rewrites “Driver parameter No. 38: Push torque limit at positioning”.

(Note 1) If a push torque limit is not set by a PTRQ command, the value set in “Driver parameter No. 38: Push torque limit at positioning” will be used.

(Note 2) The new push torque limit will remain effective even after the program ends. Therefore, when building a system using the PTRQ command, in every program explicitly specify a push torque limit using a PTRQ command before each push-motion operation. Assuming that the push torque limit will be reset to the original value when push-motion operation ends in one program can cause an unexpected problem in another program, because a different push torque limit will be used if the program is aborted due to an error, etc.

(Note 3) The new value set by a PTRQ command will become ineffective after a power ON reset or software reset.

(Note 4) A PTRQ command does not rewrite “Driver parameter No. 38: Push torque limit at positioning” (main CPU flash memory (non-volatile memory)).

[Example]

PTRQ	1	50	Change the push torque limit parameter for axis 1 to 50%.
PAPR	100	20	Set the push-motion approach distance to 100mm and the push-motion approach speed to 20mm/sec.
MOVP	2		Move to position No. 2.
PUSH	10		Move by push motion from position No. 2 to position No. 10.

● CIR2 (Move along circle via CP operation 2 (Arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CIR2	Passing position 1 number	Passing position 2 number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along a circle originating from the current position and passing positions 1 and 2, via arc interpolation.
The rotating direction of the circle is determined by the given position data.
The diagram below describes a CW (clockwise) movement. Reversing passing positions 1 and 2 will change the direction of movement to CCW (counterclockwise).

(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

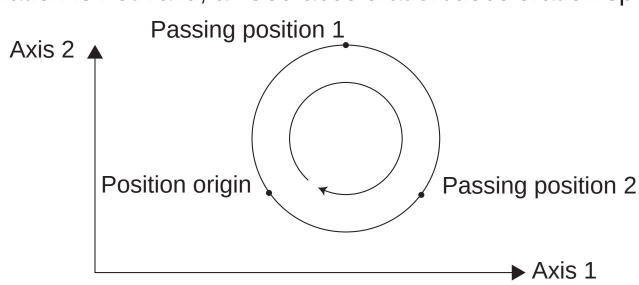
Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



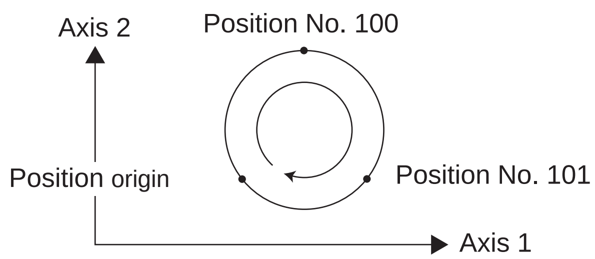
(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a CIR2 command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example]	VEL	100		Set the speed to 100mm/s.
	CIR2	100	101	Move along a circle (circular interpolation) passing position No. 100 and 101.



● ARC2 (Move along circle via CP operation 2 (Arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARC2	Passing position number	End position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along an arc originating from the current position, passing the specified position and terminating at the end position, via arc interpolation.

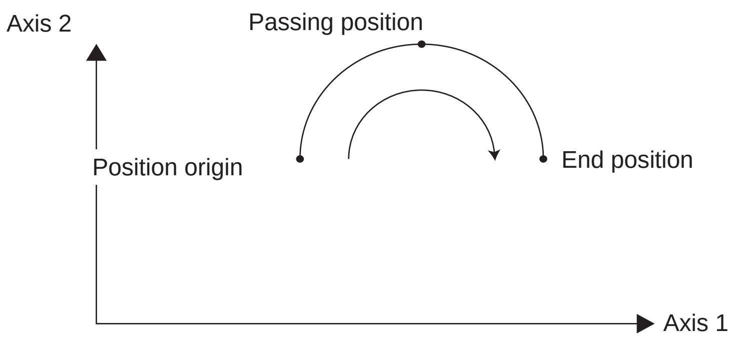
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARC2 command is executed based on the position data of axes 2 and 3.

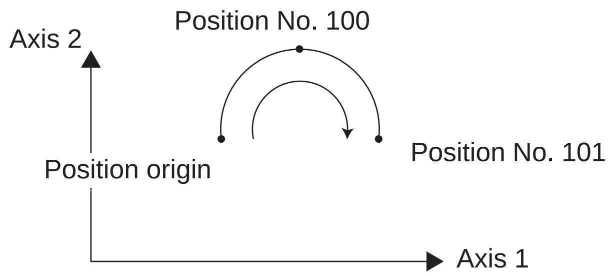
(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example]

VEL	100		Set the speed to 100mm/s.
ARC2	100	101	Move along an arc (circular interpolation) from the current position to position No. 101 by passing position No. 100.

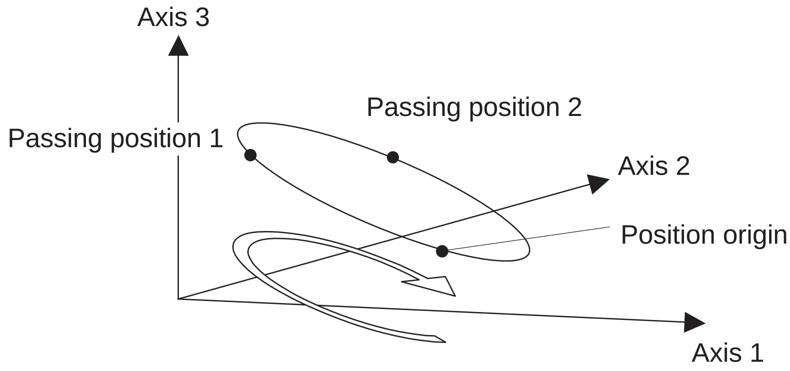


● CIRS (Move along circle three-dimensionally via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CIRS	Passing position 1 number	Passing position 2 number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along a circle by passing the passing positions 1 and 2 in this order, starting from the current position being the origin (three-dimensional movement). The direction in which to go around the circle is determined by the position data given. In the figure below, the rotating direction is reversed if passing positions 1 and 2 are swapped.



(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration	Deceleration
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1	Same as the effective value of acceleration
2	Setting value by VEL command	Setting value by ACC command	
3		Default acceleration in all-axis parameter No. 11	

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(Note 1) This command is valid on any planes in three-dimensional space. If four or more axes are set in the position data, three axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 5, for example, a CIRS command is executed based on the position data of axes 2 to 4.

(Note 2) The path tends to shift inward as the speed rises. Minor correction such as setting the position data slightly outward may be required.

(Note 3) If the diameter of the circle is smaller relative to the set speed, the speed may be limited.
(Although the extent to which the speed is limited can be reduced by raising the acceleration/acceleration, make sure the acceleration and deceleration do not exceed the range permitted by the actuator.)

(Note 4) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

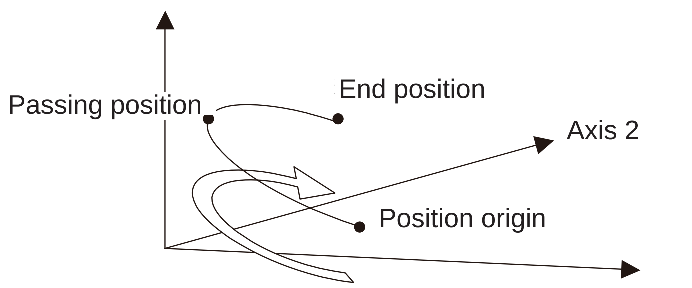
(Note 5) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

● ARCS (Move along arc three-dimensionally via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARCS	Passing position number	End position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move to the end position along an arc by passing the passing position, starting from the current position being the origin (three-dimensional movement).



(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration	Deceleration
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1	Same as the effective value of acceleration
2	Setting value by VEL command	Setting value by ACC command	
3		Default acceleration in all-axis parameter No. 11	

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(Note 1) This command is valid on any planes in three-dimensional space. If four or more axes are set in the position data, three axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 5, for example, a ARCS command is executed based on the position data of axes 2 to 4.

(Note 2) The path tends to shift inward as the speed rises. Minor correction such as setting the position data slightly outward may be required.

(Note 3) If the diameter of the circle is smaller relative to the set speed, the speed may be limited.
(Although the extent to which the speed is limited can be reduced by raising the acceleration/acceleration, make sure the acceleration and deceleration do not exceed the range permitted by the actuator.)

(Note 4) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 5) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

● CHVL (Dedicated command for linear axis: Change speed)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CHVL	Axis pattern	Speed	CP

Applicable models
XSEL-JX/KX ×   Other than XSEL-JX/KX ○

[Function] Change the speed of the axes operating in other task.
When a CHVL command is executed, the speed of the axes specified in operand 1 will change to the value specified in operand 2.

(Note 1) This command is not valid on an axis operated by a CIR, ARC, PSPL, PUSH, or ARCH command.

(Note 2) Executing a CHVL command for an axis operating in sigmoid motion (SCRV command) will generate an “Error No. CC1, Speed-change condition error”.

(Note 3) This is a temporary speed-change command issued from other task to the active packet (point). It is not affected by the data declared by VEL.

Program 1	Program 2
	VEL 300
	•
	•
	MOVP 1
CHVL 11 100	MOVP 2
	MOVP 3
	•
	•

If CHVL is executed in program 1 while MOVP 2 is executed in program 2, the travel speed of MOVP 2 will become 100mm/sec.
The speeds of other move commands will remain 300mm/sec.

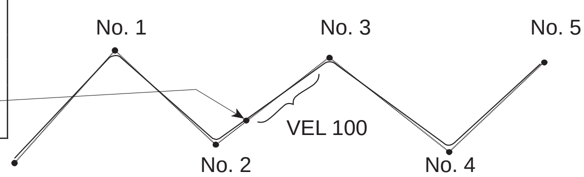
The axis pattern can be specified indirectly using a variable.
When program 1 is rephrased based on indirect specification using a variable:

11 (binary) → 3 (decimal)

LET	1	3	Assign 3 to variable 1.
CHVL	*1	100	

(Note 4) Since this command is valid only for the packet that is active at the time of execution of the command for an axis subject to continuous motion in a PATH command, etc., caution must be exercised against the timing shift. The packet handling will be put on hold during speed-change processing, so caution must also be exercised against the locus shift.

Program 1	Program 2
•	VEL 300
•	•
•	•
CHVL 11 100 →	PATH 1 5
•	
•	
•	



If CHVL is executed in program 1 while PATH is executed in program 2, or specifically during the PATH movement from point No. 2 to point No. 3, the speed specified by CHVL (100mm/sec in the above example) will become valid only during the PATH movement to point No. 3. Other travel speeds will remain at the speed specified by VEL (300mm/sec in the above example).

(Note 5) Override of the CHVL call task will be applied, so caution must be exercised.

(Note 6) The maximum speed of the specified axis completing home return will be clamped by the minimum value set in “Axis-specific parameter No. 28, Maximum operating speed of each axis” or “Axis-specific parameter No. 27, Maximum speed limited by maximum motor speed” with respect to the specified axis and related interpolation axes currently operating. To prevent the maximum speed from being limited due to the effect of other axis whose maximum speed is lower than the speed specified in the CHVL command, issue a CHVL command in multiple steps corresponding to the respective axes having different maximum speeds. In particular, specification of a CHVL command in a separate step is recommended for a rotating axis.

(Note 7) This command is dedicated for the liner axes only. “Error No. 80 Indication Prohibited Axis Error” will be issued if the SCARA axes are indicated, or SCARA axes and the liner axes are indicated at the same time.

[Example]	CHVL	11	500	⇒	CHVL	1	500
					CHVL	10	500

● ARCD (Move along arc via CP operation by specifying end position and center angle (Arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARCD	End position number	Center angle	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along a circle originating from the current position and passing positions 1 and 2, via arc interpolation.
The rotating direction of the circle is determined by the given position data.
The diagram below describes a CW (clockwise) movement. Reversing passing positions 1 and 2 will change the direction of movement to CCW (counterclockwise).
The setting unit of the center angle is degree and the set value is effective to three decimal points.

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The setting unit of the center angle is degree and the set value is effective to three decimal points.

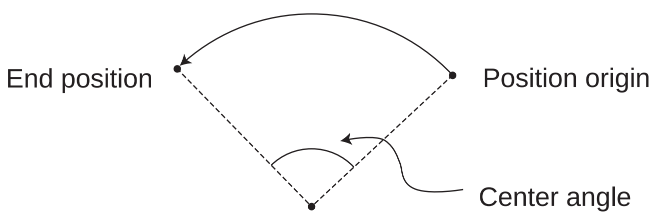
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”)

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARCD command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the center angle is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, move the path slightly inward from the soft limit boundary or make other appropriate correction. Also note that the larger the center angle, the smaller the path error becomes.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example]

VEL	100		Set the speed to 100mm/s.
ARCD	100	120	Move along an arc from the position origin to position No. 100 for a center angle of 120° (CCW direction).

● ARCC (Move along arc via CP operation by specifying center position and center angle (Arc interpolation))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARCC	Center position number	Center angle	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along an arc originating from the current position by keeping a specified radius from the center position, via arc interpolation.
Specify the center position in operand 1, and the center angle formed by the position origin and end position in operand 2. The center angle is set in a range from -3600 to 3600° (±10 revolutions). A positive value indicates CCW (counterclockwise-direction) movement, while a negative value indicates CW (clockwise-direction) movement (setting unit: °(degree)).
The setting unit of the center angle is degree and the set value is effective to three decimal points.

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The setting unit of the center angle is degree and the set value is effective to three decimal points.

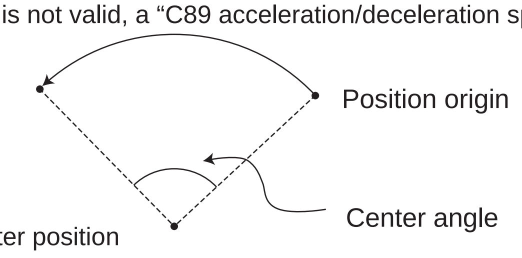
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12)

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

Priority	Speed	Acceleration (deceleration)
1	Setting value in the position data specified in operand 1	Setting value in the position data specified in operand 1
2	Setting value by VEL command	Setting value by ACC (DCL) command
3		All-axis parameter No. 11, "Default acceleration for SCARA axis" (All-axis parameter No. 12, "Default deceleration for SCARA axis") All-axis parameter No. 200, "Default acceleration for linear axis" (All-axis parameter No. 201, "Default deceleration for linear axis")

If speed is not set, a "C88 speed specification error" will generate.
If acceleration/deceleration is not valid, a "C89 acceleration/deceleration specification error" will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARCC command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the center angle is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, move the path slightly inward from the soft limit boundary or make other appropriate correction. Also note that the larger the center angle, the smaller the path error becomes.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example]	VEL	100		Set the speed to 100mm/s.
	ARCC	100	120	Move along an arc from the position origin for a center angle of 120° around position No. 100 being the center (CCW direction).

● PBND (Set positioning width)

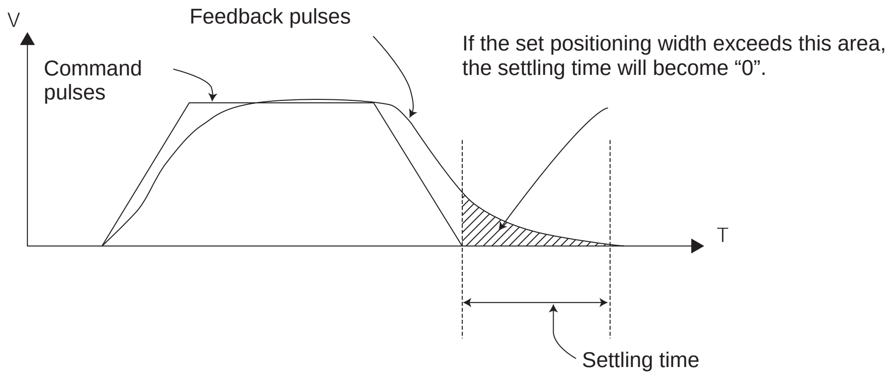
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PBND	Axis pattern	Distance	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the positioning complete width for the axes that correspond to the axis pattern specified in operand 2. The unit of operand 2 is as follows.

	Unit of operand 2
SCARA	X, Y, R: deg / Z: mm
Linear	mm / RS: deg

[Function] As a rule, positioning is deemed complete when all command pulses have been sent and the current position is within the positioning complete width. Accordingly, this command provides an effective way to shorten the tact time by shortening the settling time after rough positioning. (Normally a desired effect can be achieved with approx. 3 to 5°, but you must check on the actual equipment.)



(Note 1) If positioning width is not set with a PBND command, the value set in “Axis-specific parameter No. 58, Positioning width” will be used.

(Note 2) If the positioning width is changed, the new setting will remain valid even after the program ends. Therefore, to build a system using PBND commands, a positioning band must be expressly specified with a PBND command before operation of each program. An assumption that the positioning width will be reset to the original value when the operation ends in other program may lead to an unexpected problem, because the positioning width will become different from what is anticipated in case the applicable program is aborted due to error, etc.

(Note 3) The value set in “Axis-specific parameter No. 58, Positioning width” will not be written by a PBND command.

[Example 1] PBND 11 5 Set the positioning width for X-axis and Y-axis to 5° after this command.

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    1    3        Assign 3 to variable 1.
PBND   *1   5
```

● TMPI (Dedicated SCARA command/Move incrementally to position on tool coordinate system via PTP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TMPI	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Move incrementally on the tool coordinate system without interpolation (= via PTP operation), by the travel from the current position corresponding to the position data in operand 1.

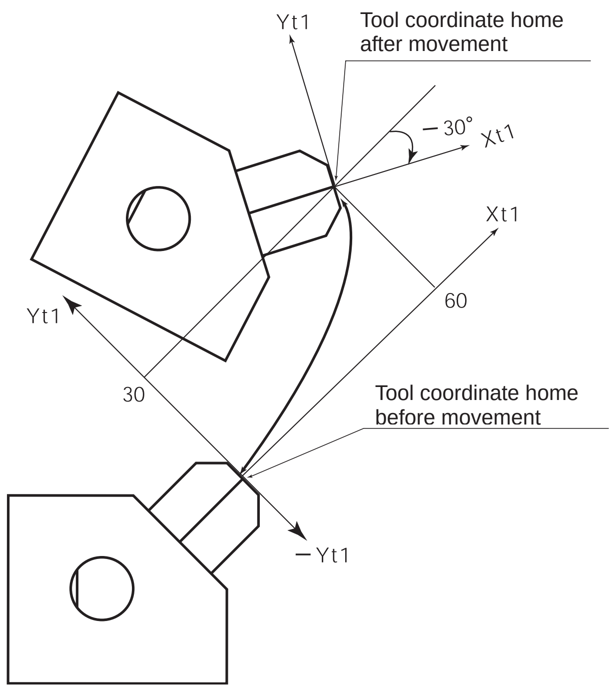
(Note 1) This command is dedicated for the SCARA axes only. “Error No. B80 Indication Prohibited Axis Error” will be issued if the liner axes are indicated.

(Note 2) If an incremental movement command is used repeatedly, coordinate conversion rounding errors, etc., will accumulate.

[Example] TMPI 120

Position data

No.	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
120	60.000	30.000	0.000	-30.000			
121							
122							



● TMLI (Dedicated SCARA command/Move incrementally to position on tool coordinate system via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TMLI	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Move incrementally on the tool coordinate system without interpolation (= via CP operation), by the travel from the current position corresponding to the position data in operand 1.

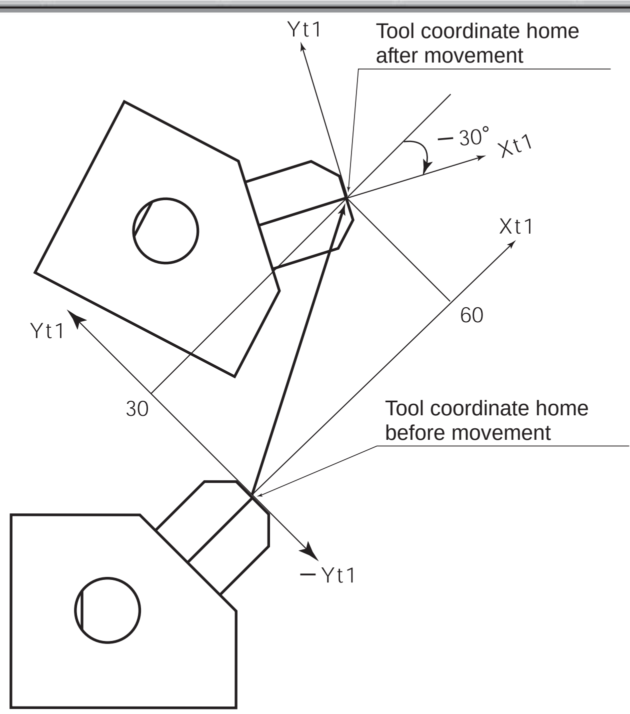
(Note 1) This command is dedicated for the SCARA axes only. “Error No. B80 Indication Prohibited Axis Error” will be issued if the liner axes are indicated.

(Note 2) If an incremental movement command is used repeatedly, coordinate conversion rounding errors, etc., will accumulate.

[Example] TMLI 120

Position data

No.	Axis1	Axis2	Axis3	Axis4	Vel	Acc	Dcl
120	60.000	30.000	0.000	-30.000			
121							
122							



● CIR (Move along circle via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CIR	Passing position 1 number	Passing position 2 number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along a circle originating from the current position and passing the positions specified in operands 1 and 2.
Therefore, reversing the settings of operands 1 and 2 will implement a circular movement in the reverse direction.
The output will turn OFF at the start of circular movement, and turn ON when the movement is complete.
Difference from CIR2:
CIR processing resembles moving along a polygon with a PATH command, while CIR2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command. (Normally CIR2 is used.)

(Note 1) If the division angle is set to “0” with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than “0” with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)
In the case of circular movement, the axes will return from passing position 2 to the start position at the speed declared by a VEL command.
Therefore, a VEL command must always be used with a CIR command.

(Note 3) The acceleration is selected in the order of the acceleration in the data at passing position 1, followed by the value in “All-axis parameter No. 11, Default acceleration”.
The deceleration will become the same value as the valid acceleration selected above. Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a CIR command is executed based on the position data of axes 2 and 3.

(Note 5) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 6) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example 1]	VEL	100		Set the speed to 100mm/s.
	CIR	100	101	Move along a circle from the current position by passing positions 100 and 101 sequentially.

[Example 2]	VEL	100		Set the speed to 100mm/s.
	LET	1	5	Assign 5 to variable 1.
	LET	2	6	Assign 6 to variable 2.
	CIR	*1	*2	Move along a circle from the current position by passing the contents of variables 1 and 2 (positions 5 and 6) sequentially.

● ARC (Move along arc via CP operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARC	Passing position number	End position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Move along an arc from the current position to the position specified in operand 2, by passing the position specified in operand 1.
The output will turn OFF at the start of arc movement, and turn ON when the movement is complete.
Difference from ARC2:
ARC processing resembles moving along a polygon with a PATH command, while ARC2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command. (Normally ARC2 is used.)

(Note 1) If the division angle is set to “0” with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than “0” with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)

(Note 3) The acceleration is selected in the order of passing position 1 data, ACC command, and all-axis parameter No. 11, “Default acceleration for SCARA axis” or all-axis parameter No. 200, “Default acceleration for linear axis”.
The deceleration will become the same value as the valid acceleration selected above. Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARC command is executed based on the position data of axes 2 and 3.

(Note 5) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 6) XSEL-PX/QX/RX/SX/RXD/SXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example 1]	VEL	100		Set the speed to 100mm/s.
	ARC	100	101	Move along an arc from the current position to position 101 by passing position 100.

[Example 2]	VEL	100		Set the speed to 100mm/s.
	LET	1	5	Assign 5 to variable 1.
	LET	2	6	Assign 6 to variable 2.
	ARC	*1	*2	Move along an arc from the current position to the content of variable 2 (position 6) by passing the content of variable 1 (position 5).

● PEND (Wait for end of operation by axes currently used by program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PEND	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
○	×	×	×	×	×	×	×	×

[Function] When a PEND command is executed, the program waits for the end of operation by the axes it is currently using. The output turns ON only when a MOVP, MOVL or PATH command has been successfully executed (positioning has been successful) in quick return mode 2 (closeness-detection return target position addition mode) or quick return mode 3 (closeness-detection return target position change mode) (the output does not turn ON if any other servo command is executed).

(Note 1) To check if the operation has been successful (positioning has been successful), execute a PEND command before the quick return mode is cancelled.

(Note 2) Be sure to also refer to the pages that explain the QRTN command and NBND command.

(Note 3) Software versions supporting PEND
Controller main application: Ver.1.04 or later
(excluding flash ROM 8Mbit versions)

PC software: Ver.7.2.3.0 or later
Teaching pendant:
IA-T-X (D): Ver.1.44 or later
SEL-T (D): Ver.1.02 or later

[13] IF structure

● IF☐☐ (Structural IF)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	IF☐☐	Variable number	Data	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
If the input condition is not satisfied and the IF☐☐ command is not executed, the program will proceed to the step next to the corresponding EDIF.
A maximum of 15 nests are supported when IS☐☐ and DW☐☐ are combined.

```
IF☐☐
  ┌ EQ  ..........  Operand 1 = Operand 2
  │ NE  ..........  Operand 1 ≠ Operand 2
──┤ GT  ..........  Operand 1 > Operand 2
  │ GE  ..........  Operand 1 ≥ Operand 2
  │ LT  ..........  Operand 1 < Operand 2
  └ LE  ..........  Operand 1 ≤ Operand 2
```

[Example 1]

```
             SVON   1111                 Set the current arm system in variable 99.
             PRDQ   1      100           Read the current X coordinate value into
                                         variable 100.
             CPNE   99     0      600    Turn OFF flag 600 if the arm system is
                                         indeterminable.
┌─ 600       IFEQ   99     1             Determine the arm system. The
│                                        processing ends if the arm system is
│                                        indeterminable.
│    ┌──     IFGE   100    0             Move to position No. 1 via PTP if the X
│    │       MOVP   1                    coordinate value is 0 or greater.
│    ├──     ELSE
│    │       MOVP   2                    Move to position No. 2 via PTP.
│    └──     EDIF
├─────       ELSE
│    ┌──     IFGE   100    0             Move to position No. 3 via PTP if the X
│    │       MOVP   3                    coordinate value is 0 or greater.
│    ├──     ELSE
│    │       MOVP   4                    Move to position No. 4 via PTP.
│    └──     EDIF
└─────       EDIF
             EXIT
```

If the current arm system is the right arm and X coordinate is 0 or greater, the axis moves to position No. 1. If the X coordinate is smaller than 0, it moves to position No. 2. If the left arm system is currently used and X coordinate is 0 or greater, the axis moves to position No. 3. If the X coordinate is smaller than 0, it moves to position No. 4.

(Note) Using a GOTO command to branch out of or into an IF☐☐-EDIF syntax is prohibited.

● IS□□ (Compare strings)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	IS□□	Column number	Column number, character literal	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Compare the character strings in the columns specified in operands 1 and 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.
If the input condition is not satisfied and the IS□□ command is not executed, the program will proceed to the step next to the EDIF.
A maximum of 15 nests are supported when IF□□ and DW□□ are combined.

```
IS□□
 ├ EQ ............. Operand 1 = Operand 2
 └ NE ............. Operand 1 ≠ Operand 2
```

[Example 1]

```
        SCPY    10    'GOFD'
                      (Move forward)
        SCPY    14    'GOBK'
                      (Move backward)
        SLEN    4                        Set the number of comparing
                                         characters to 4.
  600   ISEQ    1     'XAXS' (X-axis)    Select an axis.
        ISEQ    5     10                 Select a moving direction.
        MOVL    1                        Move to position 1 via CP.
        ELSE
        MOVL    2                        Move to position 2 via CP.
        EDIF
        ELSE
        ISNE    5     14                 Select a moving direction.
        MOVL    3                        Move to position 3 via CP.
        ELSE
        MOVL    4                        Move to position 4 via CP.
        EDIF
        EDIF
```

CP operation is performed based on position No. 1 and 2 selected in columns 1 to 4, or position No. 3 and 4 selected in columns 5 to 8.
Nothing will happen if flag 600 is OFF, in which case the program will proceed to the step next to the last EDIF.
If the following data is stored in columns 1 to 8, CP movement to position No. 1 occurs.

1 2	3 4	5 6	7 8	
XA	XS	GO	FD	

● ELSE (Else)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	ELSE	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] An ELSE command is used arbitrarily in conjunction with an IF☐☐ or IS☐☐ command to declare the command part to be executed when the condition is not satisfied.

[Example 1] Refer to the sections on IF☐☐ and IS☐☐.

● EDIF (End IF□□)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EDIF	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare the end of an IF□□ or IS□□ command.

[Example 1] Refer to the sections on IF□□ and IS□□.

[14] Structural DO

● DW☐☐ (DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	DW☐☐	Variable number	Data	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and execute the subsequent commands up to EDDO while the condition is satisfied.
The program will proceed to the step next to the corresponding EDDO if the condition is no longer satisfied.
A LEAV command can be used to forcibly end a loop.
If the input condition is not satisfied and the DW☐☐ command is not executed, the program will proceed to the step next to the corresponding EDDO.
A maximum of 15 nests are supported when IF☐☐ and IS☐☐ are combined.

DW☐☐

EQ	....	Operand 1 = Operand 2
NE	....	Operand 1 ≠ Operand 2
GT	....	Operand 1 > Operand 2
GE	....	Operand 1 ≥ Operand 2
LT	....	Operand 1 < Operand 2
LE	....	Operand 1 ≤ Operand 2

[Example 1]

```
008    DWEQ   1    0     Repeat the command up to an EDDO
         :               command while variable 1 contains "0".
         :
       EDDO
```

If DW☐☐ is specified at the start and input 8 is OFF, nothing will occur and the program will proceed to the step next to EDDO.

(Note) Using a GOTO command to branch out of or into a DW☐☐-EDDO syntax is prohibited.

● LEAV (Pull out of DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	LEAV	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Pull out of a DO☐☐ loop and proceed to the step next to EDDO.

[Example 1]

	DWEQ	1	0	Repeat the commands up to an EDDO command while variable 1 contains "0".
	:			
600	LEAV			Forcibly end the loop if flag 600 is ON and proceed to the step next to an EDDO command.
	:			
	EDDO			

● ITER (Repeat)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ITER	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Forcibly switch the control to EDDO while in a DO☐☐ loop.

[Example 1]

```
        DWEQ    1    0
          :
600     ITER
          :
        EDDO
```

Repeat the commands up to an EDDO command while variable 1 contains “0”.
Forcibly switch the control to an EDDO command and perform end judgment, if flag 600 is ON.

● EDDO (End DO WHILE)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EDDO	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare the end of a loop that began with DW☐☐.
If the DW☐☐ condition is not satisfied, the program will proceed to the step next to this command.

[Example 1] Refer to the section on DW☐☐.

[15] Multi-Branching

● SLCT (Start selected group)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SLCT	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Branch to the step next to any WH☐☐ or WS☐☐ command that exists before an EDSL command and whose condition is satisfied, or to the step next to an OTHE command if none of the conditions are satisfied.
A SLCT command must be followed by a WH☐☐, WS☐☐ or EDSL command.
A maximum of 15 nests are supported.

(Note) Using a GOTO command to branch out of or into a SLCT-EDSL syntax is prohibited.

[Example 1]

	SCPY	1	'Right'	Assign 'right' to columns 1 and 2.
	:			
600	SLCT			Jump to a W☐☐☐ whose condition is satisfied.
	WSEQ	1	'Right'	If 'right' is stored in columns 1 and 2, this command will be executed.
	:			
	WSEQ	1	'Left'	If 'left' is stored, this command will be executed.
	:			
	OTHE			If the content of columns 1 and 2 is neither of the above, this command will be executed.
	:			
	EDSL			If flag 600 is OFF, the processing will move here upon execution of any of the conditions.

● WH☐☐ (Select if true; variable)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	WH☐☐	Variable number	Data	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W☐☐☐ command or an OTHE or EDSL command when the comparison result of the content of the variable specified in operand 1 with the value specified in operand 2 satisfies the condition.

```
DW☐☐
     EQ  ............  Operand 1 = Operand 2
     NE  ............  Operand 1 ≠ Operand 2
     GT  ............  Operand 1 > Operand 2
     GE  ............  Operand 1 ≥ Operand 2
     LT  ............  Operand 1 < Operand 2
     LE  ............  Operand 1 ≤ Operand 2
```

[Example 1]

LET	1	20	Assign 20 to variable 1.
LET	2	10	Assign 10 to variable 2.
:			
SLCT			Execute multi-branching.
WHEQ	1	10	(1) will be executed if the content of variable 1 is 10. Since variable 1 contains 20, however, the next condition will be referenced.
:			
(1)			
:			
WHGT	1	*2	This command will be executed if the content of variable 1 is greater than the content of variable 2. Since variable 1 (= 20) > variable 2 (=10), (2) will be executed.
:			
(2)			
:			
OTHE			This command will be executed if none of the conditions are satisfied. In this example, since (2) was executed, (3) will not be executed.
:			
(3)			
:			
EDSL			The processing will move here if any of the conditions were satisfied and the applicable command executed. In this example, (2) and (4) will be executed.
:			
(4)			
:			

* If multiple conditions are likely to be satisfied, remember that the first W☐☐☐ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

● WS☐☐ (Select if true; character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	WS☐☐	Column number	Column number, character literal	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W☐☐☐ command or an OTHE or EDSL command when the comparison result of the character strings in the columns specified in operands 1 and 2 satisfies the condition.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

```
WS☐☐
   ├─ EQ  ...........  Operand 1 = Operand 2
   └─ NE  ...........  Operand 1 ≠ Operand 2
```

[Example 1]

SLEN	3		Set the number of comparing characters to 3.
SCPY	1	‘ABC’	Assign ‘ABC’ to column 1.
LET	1	2	Assign 2 to variable 1.
:			
SLCT			Execute multi-branching.
WSEQ	1	‘XYZ’	(1) will be executed if columns 1 to 3 contain ‘XYZ’. Since columns 1 to 3 contain ‘ABC’, however, this command will not be executed.
:			
(1)			
:			
WSEQ	2	*1	(2) will be executed if the content of the number of characters specified by SLEN after column 2 is the same as the content of the column specified in variable 1.
:			
(2)			
:			
OTHE			This command will be executed if none of the conditions are satisfied. In this example, since (2) was executed, (3) will not be executed.
:			
(3)			
:			
EDSL			The processing will move here if any of the conditions were satisfied and the applicable command executed. In this example, (2) and (4) will be executed.
:			
(4)			
:			

* If multiple conditions are likely to be satisfied, remember that the first W☐☐☐ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

[15] Multi-Branching

● OTHE (Select other)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	OTHE	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] This command is used between SLCT and EDSL commands to declare the command to be executed when none of the conditions are satisfied.

[Example 1] Refer to the sections on SLCT, WH☐☐ and WS☐☐.

● EDSL (End selected group)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EDSL	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Declare the end of a SLCT command.

[Example 1] Refer to the sections on SLCT, WH□□ and WS□□.

[16] System Information Acquisition

● AXST (Get axis status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	AXST	Variable number	Axis number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Store in the variable specified in operand 1 the status (axis error number) of the axis specified in operand 2.

(Note 1) If the obtained result is "0", it means no axis error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.

[Example] AXST 1 2 Read the error number for axis 2 to variable 1.

If 3188 (decimal) is stored in variable 1 after the execution of this command:

3188 / 16 = 199 ...4
199 / 16 = 12 (= C) ...7

$3188 = 12\ (= C) \times 16^2 + 7 \times 16^2 + 4$
$= C74\ (HEX)$ (Hexadecimal number)

Therefore, an "Error No. C74, Actual-position soft limit over error" is present.

● PGST (Get program status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PGST	Variable number	Program number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Store in the variable specified in operand 1 the status (program error number) of the program specified in operand 2.

(Note 1) If the obtained result is “0”, it means no program error is present.

(Note 2) Although the error lists are written in hexadecimals, the status to be stored (program error number) is a decimal.
Therefore, the decimal program error numbers must be converted to hexadecimals.

[Example] PGST 1 2 Read the error number for program No. 2 to variable 1.

● SYST (Get system status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SYST	Variable number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Store the system status (top-priority system error number) in the variable specified in operand 1.

(Note 1) If the obtained result is “0”, it means no system error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.
(Note 3) Relationship of error statuses

System errors
- Program errors
- Axis errors
- Other errors

* An axis error that generates during operation with a program command will be registered both as a program error and an axis error.

[Example] SYST 1 Read the system error number to variable 1.

● GARM (Dedicated SCARA command/Get current arm system)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GARM	Variable number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	○	○	○	×	×	○ (PCX/PGX only)

[Function] Acquire the current arm system and set one of the following values corresponding to this arm system in the variable specified in operand 1:
Arm system indeterminable = 0
Right arm system = 1
Left arm system = -1

(Note 1) The arm system effective immediately after the command execution is set. It is not that the arm system is always monitored.

(Note 2) In XSEL-RX/SX/RXD/SXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 3) When GRP Command is undeclared, or GRP 11111111 (1st to 8th axes effective) is declared, the current arm system of the SCARA axes (1st to 4th axes) is set. When an acquirement of the current arm system for the SCARA axes (5th to 8th axes) is required, make only the 5th to 8th axes valid in GRP Command and execute GARM Command.

[Example 1]

GRP	1111	It makes the 1st to 4th axes effective.
GARM	200	Acquire the current arm system of the SCARA axes (1st to 4th axes) to Variable No. 200.

[Example 2]

GRP	11110000	It makes the 5th to 8th axes effective.
GARM	200	Acquire the current arm system of the SCARA axes (5th to 8th axes) to Variable No. 200.

[17] Zone

● WZNA (Dedicated linear axis command/Wait for zone ON based on AND gate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WZNA	Zone number	Axis pattern	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

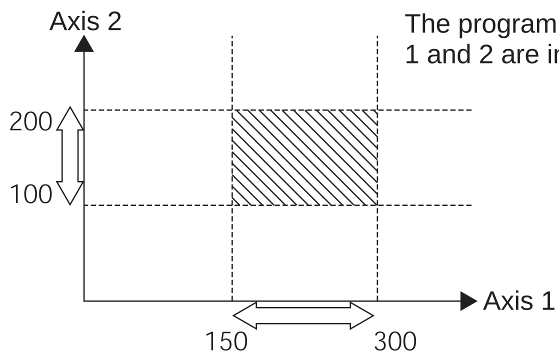
(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter No. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter No. 88, 91, 94 and 97" irrespective of this command.
(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, "Error No. B80: Specification-prohibited axis error" occurs.

[Example 1] WZNA 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3        Assign 3 to variable 5.
WZNA   1    *5
```

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001mm)	150000	100000



The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.

● WZNO (Dedicated linear axis command/Wait for zone ON based on OR gate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WZNO	Zone number	Axis pattern	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).

(Note 2) A maximum of four areas can be set as zones for each axis (“Axis-specific parameter No. 86 to 97”).

(Note 3) Zone output can be specified using “Axis-specific parameter No. 88, 91, 94 and 97” irrespective of this command.

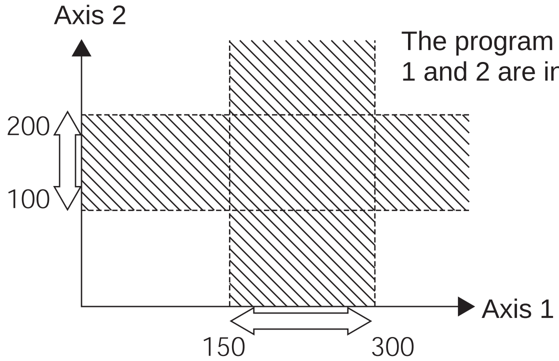
(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, “Error No. B80: Specification-prohibited axis error” occurs.

[Example 1] WZNO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3        Assign 3 to variable 5.
WZNO   1    *5
```

	Axis 1	Axis 2
“Axis-specific parameter No. 86, Zone 1 max.” (Value is set in units of 0.001mm)	300000	200000
“Axis-specific parameter No. 87, Zone 1 min.” (Value is set in units of 0.001mm)	150000	100000



The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.

● WZFA (Dedicated linear axis command/Wait for zone OFF based on AND gate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WZFA	Zone number	Axis pattern	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

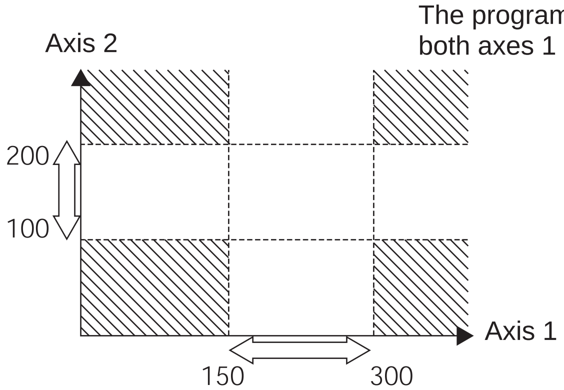
(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter No. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter No. 88, 91, 94 and 97" irrespective of this command.
(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, "Error No. B80: Specification-prohibited axis error" occurs.

[Example 1] WZFA 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes OFF (inside the shaded area shown in the diagram below)

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 5 3 Assign 3 to variable 5.
WZFA 1 *5

	Axis 1	Axis 2
"Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001mm)	300000	200000
"Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded



● WZFO (Dedicated linear axis command/Wait for zone OFF based on OR gate)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WZFO	Zone number	Axis pattern	CP

Applicable models
XSEL-JX/KX × Other than XSEL-JX/KX ○

[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).

(Note 2) A maximum of four areas can be set as zones for each axis (“Axis-specific parameter No. 86 to 97”).

(Note 3) Zone output can be specified using “Axis-specific parameter No. 88, 91, 94 and 97” irrespective of this command.

(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, “Error No. B80: Specification-prohibited axis error” occurs.

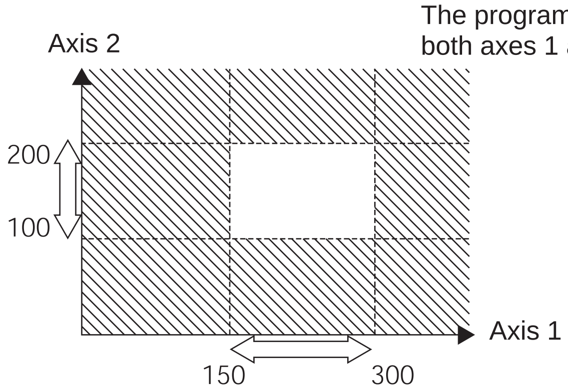
[Example 1] WZFO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes OFF (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3      Assign 3 to variable 5.
WZFO   1    *5
```

	Axis 1	Axis 2
“Axis-specific parameter No. 86, Zone 1 max.” (Value is set in units of 0.001mm)	300000	200000
“Axis-specific parameter No. 87, Zone 1 min.” (Value is set in units of 0.001mm)	150000	100000

The program will proceed to the next step if both axes 1 and 2 are inside the shaded



[18] Communication

● OPEN (Open channel)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OPEN	Channel number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Open the channel specified in operand 1.
The specified channel will be enabled to send/receive hereafter.
Prior to executing this command, a SCHA command must be used to set an end character.

[Example 1]

```
SCHA    10
OPEN    1
```

Specify 10 (= LF) as the end character.
Open channel 1.

> Note: If "Open 0" is executed the teaching pendant connector (D-sub25pin) is cut off. (This is because channel 0 is used by both the teaching pendant and PC software.)

● CLOS (Close channel)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CLOS	Channel number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Close the channel specified in operand 1.
The specified channel will be disabled to send/receive hereafter.

[Example 1]

```
CLOS    1
Close channel 1.

LET     1    2
CLOS    *1
        Assign 2 to variable 1.
        Close the content of variable 1 (channel 2).
```

● READ (Read)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	READ	Channel number	Column number	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Read a character string from the channel specified in operand 1 to the column specified in operand 2.
Read will end when the character specified by a SCHA command is received.
Either a local or global column may be specified.
A return code will be stored in a local variable (variable 99 under the factory setting) immediately after this command is executed.
Whether or not the command has been executed successfully can be checked based on this return code. Define appropriate processing to handle situations where the command execution failed due to an error.
Setting "0" in operand 2 will specify a dummy read (receive buffer cleared and receive disabled) (the return code will indicate that the command was successfully executed). The versions of tools in which 0 can be entered in operand 2 are specified below. With tools of these versions, 0 can be specified indirectly if it cannot be entered directly from the tool:

- PC software Ver.1.1.1.0 or later
- Teaching pendant application Ver.1.06 or later

[Example 1]

SCHA	10		Set LF (= 10) as the end character.
OPEN	1		Open channel 1.
READ	1	2	Read a character string from channel 1 to column 2 until LF is received.
TRAN	1	99	Assign the return code (variable 99) to variable 1.
CLOS	1		Close the channel 1.
SLCT			The processing flow branches out in accordance with each return code. (Note) Using a GOTO command to branch out of an SLCT-EDSL syntax or to other branch processing within the syntax is prohibited.
WHEQ	1	0	If the content of variable 1 is "0" (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution.
:			
(1)			
:			
WHEQ	1	1	If the content of variable 1 is "1" (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary.
:			
(2)			
:			
WHEQ	1	2	If the content of variable 1 is "2" (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary.
:			
(3)			
:			
OTHE			If the content of variable 1 is not "0", "1" or "2", (4) will be executed. In (4), define appropriate error handling, if necessary.
:			
(4)			
:			
EDSL			Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here.

(Note 1) A READ command must be executed before the other side sends the end character.

(Note 2) Dummy read (operand 2: 0) cannot be specified for channel No. 31 to 34 (Ethernet option).

```
SCHA    10
OPEN    1
READ    1        2
     <------------------------ Other side
CLOS    1
```

- Return code of the READ command
  The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24”. The default variable number is 99.

0	: READ completed successfully (Receive complete)
1	: READ timeout (the timeout value is set by a TMRD command) (Continue to receive)
2	: READ timer cancelled (the wait status is cancelled by a TIMC command) (Continue to receive)
3	: READ SCIF overrun error (Receive disabled)
4	: READ SCIF receive error (framing error or parity error) (Receive disabled)
5	: READ factor error (program abort error) (Receive disabled) (Cannot be recognized by SEL commands)
6	: READ task ended (program end request, etc.) (Receive disabled) (Cannot be recognized by SEL commands)
7	: READ SCIF receive error due to other factor (Receive disabled)
8	: READ expansion SIO overrun error (Receive disabled)
9	: READ expansion SIO parity error (Receive disabled)
10	: READ expansion SIO framing error (Receive disabled)
11	: READ expansion SIO buffer overflow error (Receive disabled)
12	: READ expansion SIO receive error due to other factor (Receive disabled)
13 to 20	: Used only in Ethernet (optional)
21	: READ SIO receive temporary queue overflow error (Receive disabled)
22	: READ SIO slave receive queue overflow error (Receive disabled)

● TMRD (Set READ timeout value)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TMRD	Timer period	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
○	×	×	○	×	×	×	TT:○, TTA:×	×

[Function] Set the timeout to be applied to a READ command.
The timer setting specified in operand 1 will set the maximum time the program will wait for the character string read to end when a READ command is executed.
If the end character could not be read before the timer is up during the execution of the READ command, a timeout will occur and the program will move to the next step.
(You can check if a timeout has occurred by checking the return code which is stored in a local variable (factory setting: variable 99) immediately after the READ command has been executed. If necessary, program an appropriate processing to be performed when a timeout occurs.)
Setting the timer to “0” will allow the READ command to wait infinitely, without timeout, until the end character is read.
The timer setting is input in seconds (setting range: 0 to 99.00sec) including up to two decimal places.

(Note) TMRD is set to “0” in the default condition before TMRD setting is performed.

[Example]

SCHA	10		Set LF (=10) as the end character.
TMRD	30		Set the READ timeout value to 30sec.
OPEN	1		Open channel 1.
READ	1	2	Read the character string from channel 1 to column 2 until LF is read.
TRAN	1	99	Assign the return code to variable 1.
CLOS	1		Close the channel.
SLCT			The processing flow branches out in accordance with each return code. (Note) Using a GOTO command to branch out of an SLCT-EDSL syntax or to other branch processing within the syntax is prohibited.
WHEQ : (1) :	1	0	If the content of variable 1 is “0” (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution.
WHEQ : (2) :	1	1	If the content of variable 1 is “1” (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary.
WHEQ : (3) :	1	2	If the content of variable 1 is “2” (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary.
OTHE : (4) :			If the content of variable 1 is not “0”, “1” or “2”, (4) will be executed. In (4), define appropriate error handling, if necessary.
EDSL			Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here.

Read completes successfully within 30sec → Variable No. 1 = 0
Timeout occurs → Variable No. 1 = 1
* The return code of READ command may not be limited to 0 or 1. The variable to store the return code can be set in “Other parameter No. 24”. (Main application Ver.0.21 or later) For details, refer to the explanation of the READ command.

● TMRW (Set READ/WRIT timeout value)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TMRW	Read timer setting	(Write timer setting)	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	○	TT:×, TTA:○	○

[Function] Set the timeout to be applied to a READ/WRIT command.
The timer setting specified in operand 1 will set the maximum time the program will wait for the character string read to end when a READ command is executed.
If the end character could not be read before the timer is up during the execution of the READ command, a timeout will occur and the program will move to the next step.
(You can check if a timeout has occurred by checking the return code which is stored in a local variable (factory setting: variable 99) immediately after the READ command has been executed.)
If the timer period is set to 0, the READ command causes the program to wait infinitely until the end characters are read, by assuming that there is no timeout.
The timer setting is input in seconds (setting range: 0 to 99.00sec) including up to two decimal places.
A variable can be specified indirectly in operand 1.

(Note) TMRW is set to “0” in the default condition before TMRW setting is performed.

[Example]

SCHA	10		Set LF (=10) as the end character.
TMRW	30		Set the READ timeout value to 30sec.
OPEN	1		Open channel 1.
READ	1	2	Read the character string from channel 1 to column 2 until LF is read.
TRAN	1	99	Assign the return code to variable 1.
CLOS	1		Close the channel.

Read completes successfully within 30sec → Variable No. 1 = 0
Timeout occurs → Variable No. 1 = 1
* The return code of READ command may not be limited to 0 or 1. The variable to store the return code can be set in “Other parameter No. 24”. Refer to the explanation of READ command for details.
For the time period specified in operand 2, set the timeout value to be applied when a WRIT command is executed (maximum wait time for completion of send).
(Maximum wait time for end based on flow control)
The write timer period is valid only for standard SIO (channels 1 and 2 supporting flow control).

For the time period specified in operand 2, set the timeout value to be applied when a WRIT command is executed (maximum wait time for completion of send).
(Maximum wait time for end based on flow control) (Arbitrary)
The write timer setting is available only on standard SIO (flow control support channels 1 and 2).
This command is recognized as a TMRD on XSEL-JX/KX controllers, and as TMRW on XSEL-PX/QX controllers. If a program created for an XSEL-JX/KX controller is transferred to an XSEL-PX/QX controller, the PC software automatically converts “TMRD” to “TMRW” before the file is transferred. This command is recognized as a TMRD on XSEL-JX/KX controllers, and as TMRW on XSEL-PX/QX controllers. If a program created for an XSEL-JX/KX controller is transferred to an XSEL-PX/QX controller, the PC software automatically converts “TMRD” to “TMRW” before the file is transferred.

● WRIT (Write)

Extension condition (LD,A,O,AB,OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	WRIT	Channel number	Column number	CC(Note 1)

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Write the character string in the column specified in operand 2 to the channel specified in operand 1.
The operation will end when the character specified by a SCHA command is written.
Either a local or global column can be specified.

[Example]

SCHA	10		Set LF (= 10) as the end character.
OPEN	1		Open channel 1.
WRIT	1	2	Write the character string in column 2 to channel 1 until LF is written.
CLOS	1		Close the channel.

With a standard SIO (channel 1 or 2), WRIT is supported by (can be sent in) a task other than the one that opened the channel, as long as the channel is currently open. Accordingly, by sending WRIT in other task after executing READ in a task that opened the channel, a response can be received from the other side without delay after sending from XSEL.

(Note 1) CP is performed if the channel is other than 1 and 2.

Return code of WRIT command (channels 1 and 2 only)
The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24”. The default variable number is 99.
0 : WRIT completed successfully
1 : WRIT timeout (the timeout value is set by a TMRW command)
2 : WRIT timer cancelled (the wait status is cancelled by a TIMC command)
3 to 4 : Reserved by the system
5 : WRIT factor error (program abort error) (Cannot be recognized by SEL commands)
6 : WRIT task ended (program end request, etc.) (Cannot be recognized by SEL commands)

● SCHA (Set end character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SCHA	Character code	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the end character to be used by a READ or WRIT command.
Any character from 0 to 255 (character code used in BASIC, etc.) can be specified.

[Example] Refer to the sections on READ and WRIT commands.

[19] String Operation

● SCPY (Copy character string)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SCPY	Column number	Column number, character literal	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Copy the character string in the column specified in operand 2 to the column specified in operand 1.
Copy will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, copy will be performed for the entire length of the literal.

[Example 1]	SCPY	1	‘ABC’	Copy ‘ABC’ to column 1.
	SLEN	10		Set the copying length to 10 bytes.
	SCPY	100	200	Copy 10 bytes from column 200 to column 100.

● SCMP (Compare character strings)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SCMP	Column number	Column number, character literal	EQ

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Compare the column specified in operand 1 with the column specified in operand 2.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

[Example 1]	SCMP	1	'ABC'	600	Flag 600 will turn ON if columns 1 to 3 contain 'ABC'.
	SLEN	5			Set the comparing length to 5 bytes.
	SCMP	10	30	999	Turn ON flag 999 if five bytes from columns 30 and 10 match.

● SGET (Get character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SGET	Variable number	Column number, character literal	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Assign one character from the column specified in operand 2 to the variable specified in operand 1.
If a character-string literal is specified in operand 2, the first character will be assigned.

[Example 1]

```
SGET   1    100
Assign one byte from column 100 to variable 1.

LET    1    3      Assign 3 to variable 1.
LET    2    1      Assign 1 to variable 2.
SCPY   1    'A'    Copy 'A' to column 1.
SGET   *1   *2     Assign 'A' from the content of variable 2 (column 1)
                   to the content of variable 1 (variable 3).
```

● SPUT (Set character)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SPUT	Column number	Data	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the data specified in operand 2 in the column specified in operand 1.

[Example 1]	SPUT	5	10	Set 10 (LF) in column 5.
	LET	1	100	Assign 100 to variable 1.
	LET	2	50	Assign 50 to variable 2.
	SPUT	*1	*2	Set the content of variable 2 (50 (‘2’)) in the content of variable 1 (column 100).

● STR (Convert character string; decimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	STR	Column number	Data	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Copy to the column specified in operand 1 a decimal character string converted from the data specified in operand 2.
The data will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a 10 digit integer including eight or more valid digits, conversion of the values in the eighth and subsequent digits will not be guaranteed (the values through the seventh digits will be converted properly.)

[Example]

SLEN	5.3		Set a length consisting of five integer digits and three decimal digits.
STR	1	123	The following values will be set in columns 1 to 9:

1	2	3	4	5	6	7	8	9
		1	2	3	.	0	0	0

LET	1	10	Assign 10 to variable 1.
LET	102	987.6543	Assign 987.6543 to variable 102.
SLEN	2.3		Set a length consisting of two integer digits and three decimal digits.
STR	*1	*102	The following values will be set in columns 10 to 15:

10	11	12	13	14	15
8	7	.	6	5	4

Since the data exceeds the specified length, 87 without 9 in the 100s place is set in the integer part, while 654 with 3 in the fourth decimal place rounded is set in the fraction part.

● STRH (Convert character string; hexadecimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	STRH	Column number	Data	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Copy to the column specified in operand 1 a hexadecimal character string converted from the data specified in operand 2.
Only the integer part will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a negative value, 8 columns will be required to covert the entire data.

[Example]

SLEN	5		Set a format consisting of 5 integer digits.
STRH	1	255	The following values will be set in columns 1 to 5:

1	2	3	4	5
			F	F

LET	1	10	Assign 10 to variable 1.
LET	102	987.6543	Assign 987.6543 to variable 102.
SLEN	2.3		Set a length consisting of 2 integer digits and 3 decimal digits.
STRH	*1	*102	The following values will be set in columns 10 and 11:

10	11
D	B

“.3” in the SLEN command and “.6543” in variable 102, which are the decimal part, will be ignored.
The integer part is expressed as ‘3DB’ in hexadecimal. Since the length is two digits, however, “3” in the third digit will be cut off.

● VAL (Convert character string data; decimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	VAL	Variable number	Column number, character literal	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Convert the decimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 18 characters or less.

[Example]

```
SCPY  10   '1234'   Set '1234' in column 10.
SLEN  4             Set the converting length to 4 bytes.
VAL   1    10       Assign 1234, which is a binary converted from
                    '1234' in column 10, to variable 1.


LET   1    100      Assign 100 to variable 1.
LET   2    20       Assign 20 to variable 2.
SCPY  20   '1234'   Copy '1234' to column 20.
SCPY  24   '.567'   Copy '.567' to column 24.
SLEN  8             Set the converting length to 8 bytes.
VAL   *1   *2       Assign 1234.567, which is a binary converted from
                    '1234.567' in the content of variable 2 (column 20)
                    to the content of variable 1 (variable 100).
```

● VALH (Convert character string data; hexadecimal)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	VALH	Variable number	Column number, character literal	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Convert the hexadecimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
Only the integer part will be converted, with the decimal part being ignored.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 8 characters or less.

[Example]

SCPY	10	‘1234’	Set ‘1234’ in column 10.
SLEN	4		Set the converting length to 4 bytes.
VALH	1	10	Assign 4660, which is a binary converted from hexadecimal ‘1234’ in column 10, to variable 1.

LET	1	100	Assign 100 to variable 1.
LET	2	20	Assign 20 to variable 2.
SCPY	20	‘ABCD’	Copy ‘ABCD’ to column 20.
SLEN	4		Set the converting length to 4 bytes.
VALH	*1	*2	Assign 43981, which is a binary converted from hexadecimal ‘ABCD’ in the content of variable 2 (column 20) to the content of variable 1 (variable 100).

● SLEN (Set length)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SLEN	Character string length	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the length to be processed by a string command.
This must always be set before using the following commands:

SCMP .... Decimal part is invalid.
SCPY .... Decimal part is invalid.
IS☐☐ .... Decimal part is invalid.
WS☐☐ .... Decimal part is invalid.
STRH .... Decimal part is invalid.
VAL, VALH .... Decimal part is invalid.
STR .... Decimal part is valid.

[Example] Refer to the examples of the above commands:

[20] Arch-Motion

● ARCH (Arch motion)

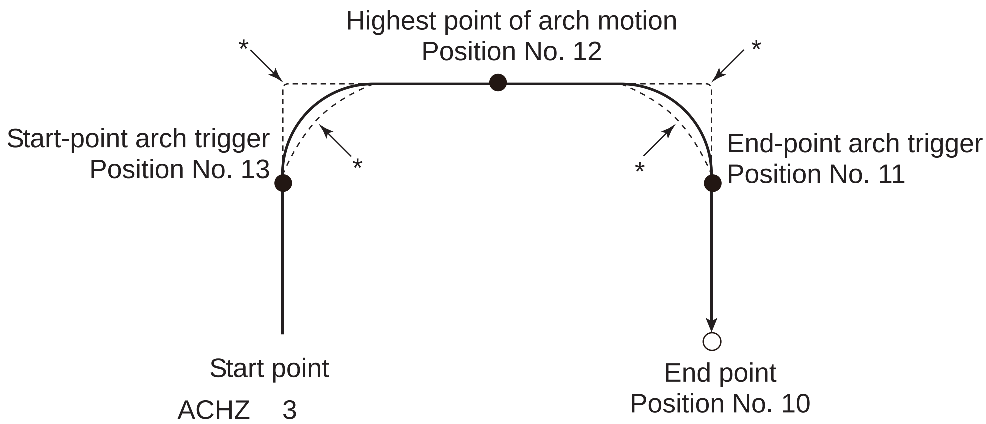
Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ARCH	Position number	Position number	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Perform arch motion from the current point and move to the specified points.
- Move to the points specified in operand 1, via arch motion.
- Movements in directions other than the arch-motion Z-axis direction will begin after rising from the current point to the start-point arch trigger. After the Z point specified in operand 2 (as the highest point) is passed and movements in directions other than the arch-motion Z-axis direction are complete, the axes will come down to the end-point arch trigger and reach the specified point.
- Palletizing arch triggers must be set using an ATRG command.

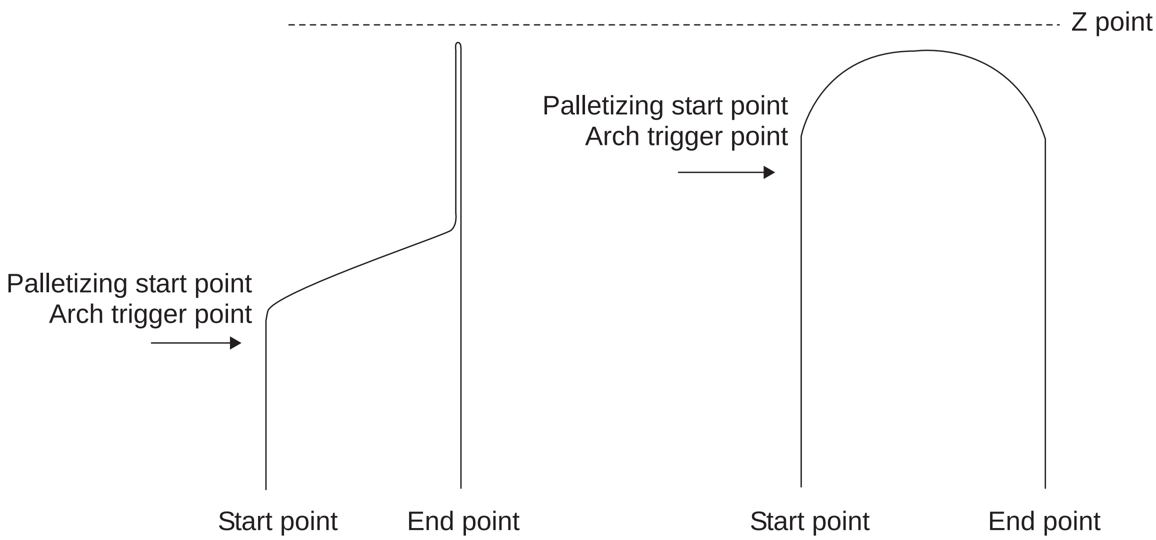
(Note 1) If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur. Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

(Note 2) The arch motion operation of SCARA axis is PTP operation and the linear drive axis is CP operation.




```
ACHZ   3
ATRG   13    11
  •
  •
  •
ARCH   10    12
```

* When the operation is resumed after a pause, depending on the position where the operation is resumed the locus may follow the lines (dotted lines) indicated by asterisks in the diagram for the composite section from ascent to horizontal movement or from horizontal movement to descent. Be careful not to cause interference.
- The arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis component of the point data specified in operand 1, if any, plus the arch-motion Z-axis offset. If there is no arch-motion Z component, the arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis coordinate of the start point plus the arch-motion Z-axis offset. (Normally the offset is added to all arch-motion positions, such as the arch triggers and Z point.)
- An error will generate if the start-point arch trigger is set below the start point or the end-point arch trigger is set below the end point. (Note: Up/down has nothing to do with +/- on the coordinate system.)
- The arch-motion Z-axis up direction refers to the direction toward the Z point from the start point (the down direction refers to the opposite direction), and has nothing to do with the size of coordinate value. Therefore, be sure to confirm the actual operating direction when using this command.
- The arch-motion Z-axis will come down after a rise-process command value is output. Therefore, one of the following operations will be performed depending on how the arch-trigger point and Z point are set.
  If the resulting operation is undesirable, change the arch trigger and/or Z point to improve the efficiency of movement.



- As for the arch-trigger end position data, movement also starts/ends above the applicable arch trigger for any effective axis, other than the arch motion Z-axis, if data of such axis is included in the position data.
- If the end position data includes R-axis data, movement of the R-axis starts/ends above the applicable arch trigger.
- If a composite arch trigger motion is set, a given effective axis, other than the arch motion Z-axis, also moves if data of such axis is included in the end point data. In this case, movement of the axis also starts/ends above the applicable arch trigger.

● ACHZ (Declare arch-motion Z-axis)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ACHZ	Axis number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Specify the axis number representing the arch-motion Z direction.
The axis number specified in operand 1 will be set as the axis number representing the arch-motion Z direction.
If the output field is specified, the output will turn ON after this command is executed.

[Example] ACHZ 3

(Note 1) The arch motion Z-axis is available for indication only on the work coordinate system Z-axis (Axis No. 3 or Axis No. 7).

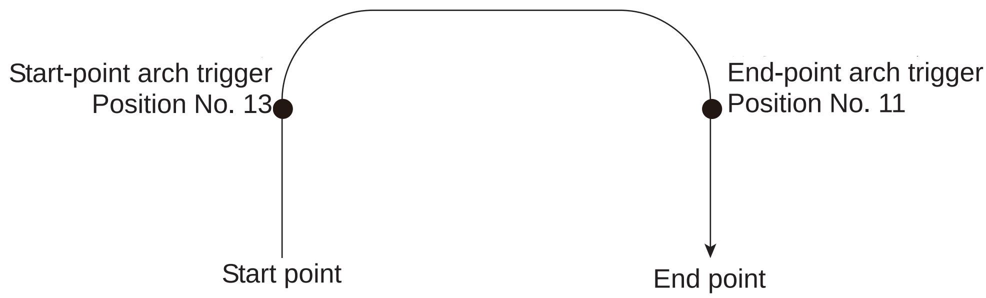
● ATRG (Set arch triggers)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ATRG	Position number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the arch triggers used for arch motion.

(This setting becomes valid when an ARCH command is executed.)
Set the arch-motion Z-axis position data in the point data specified in operand 1 as the start-point arch trigger, and set the arch-motion Z-axis position data in the point data specified in operand 2 as the end-point arch trigger.



ATRG 13 11

(Refer to “Palletizing Setting” – “Arch triggers” under “How to Use”.)
For an arch-motion operation, set it so that a horizontal movement will begin when the start-point arch trigger is reached during ascent from the start point, and that the end-point arch trigger will be reached after a horizontal movement is completed during descent.
If the output field is specified, the output will turn ON after this command is executed.

● AEXT (Set composite arch motion)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	AEXT	(Position number)	Prohibited	CP

Applicable models
ASEL/PSEL/SSEL × Other than ASEL/PSEL/SSEL ○

Set a composite arch motion. Set coordinate values other than the arch motion Z-axis at the end position of arch motion.
Use the position number specified in operand 1 for setting composite motion.
With SCARA robots, the R-axis becomes a composite arch motion axis.
When the arch motion is executed, the end coordinate of the composite axis corresponds to effective axis data, other than that of the arch motion Z-axis, included in the arch-motion end point data.
If nothing is specified in operand 1, the position number already declared for setting composite motion becomes invalid. If the output is specified, it turns ON after this command has been executed.

(Note 1) Setting of the arch motion composition axes cannot be established for linear drive axes in PX/QX.

● OFAZ (Set arch-motion Z-axis offset)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OFAZ	Offset value	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the offset in the arch-motion Z-axis direction.
The value specified in operand 1 will be set as the offset in the arch-motion Z-axis direction.
The offset amount is set in mm and the effective resolution is 0.001mm.
A negative value can also be specified as the offset, as long as the operation range will not be exceeded.
This offset is valid only at the end point of ARCH (arch motion) operation.
If the output field is specified, the output will turn ON after this command is executed.

[21] Palletizing Definition

● BGPA (Declare start of palletizing setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	BGPA	Palletizing number	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Declare the start of a palletizing setting.
Once this command is executed, palletizing setting for the palletizing number specified in operand 1 will be enabled.
(In the case of an ACHZ, AEXT, OFAZ or ATRG command, setting is enabled without declaring BGPA.)
The input range of palletizing number is from 1 to 10.
When the palletizing setting is complete, execute EDPA.
Nested BGPAs are not supported. To declare start of another palletizing setting, execute an EDPA command and then execute a BGPA command again.
If the output field is specified, the output will turn ON after this command is executed.

(Note) Using a GOTO command to branch out of or into a BGPA-EDPA syntax is prohibited.

● EDPA (Declare end of palletizing setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	EDPA	Prohibited	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Declare the end of a palletizing setting.
If a palletizing-setting command (excluding BGPA, ACHZ, ATRG, AEXT and OFAZ) is executed before another BGPA is declared following an execution of this command (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

● PAPI (Set palletizing counts)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAPI	Count	Count	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set counts in the palletizing-axis directions.
The count specified in operand 1 will apply to the preferential-axis (PX-axis) direction, while the count specified in operand 2 will apply to the PY-axis direction.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

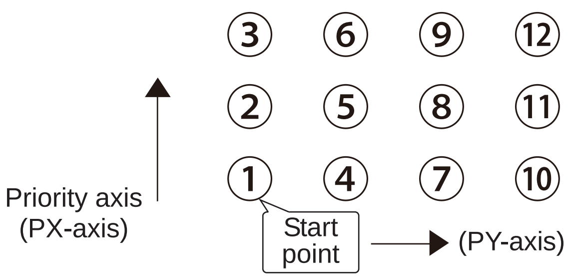
● PAPN (Set palletizing pattern)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAPN	Pattern number	Prohibited	CP

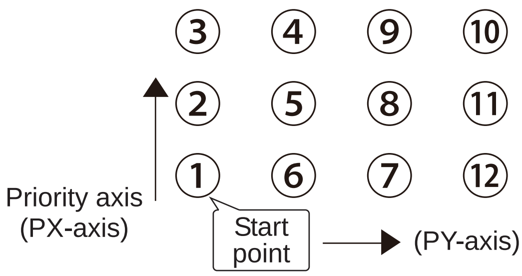
Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set a palletizing pattern.
The palletizing pattern specified in operand 1 will be set (1 = Pattern 1, 2 = Pattern 2).
If this command is not declared, pattern 1 will be used.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

Pattern 1



Pattern 2



● PASE (Declare palletizing axes)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PASE	Axis number	Axis number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the two axes to be used in palletizing (PX and PY-axes).
The axis specified in operand 1 will be set as the preferential axis (PX-axis).
The axis specified in operand 2 will be set as the PY-axis.
This command is used in conjunction with PAPT and PAST.
It cannot be used together with a 3-point teaching (PAPS) command. Whichever is set later will be given priority.

3-point teaching (PAPS) is recommended for palletizing that requires precision.

If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur.
Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur.
Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

● PAPT (Set palletizing pitches)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAPT	Pitch	Pitch	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set palletizing pitches.
The value specified in operand 1 will be set as the pitch for the preferential axis (PX-axis), while the value specified in operand 2 will be set as the pitch for the PY-axis.
This command is used in conjunction with PASE and PAST.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

● PAST (Set palletizing reference point)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAST	(Position number)	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the reference point for PX-axis (priority axis), PY-axis and PZ-axis (when palletizing Z-axis declaration is effective) to be used in palletizing calculation.
If a value is set in operand 1, that position number specified in operand 1 will be used to store the reference point data.
If no value is set in operand 1, the position-number setting for storing reference point data will become invalid.
This command is used in conjunction with PASE and PAPT.

If this command is not set, the reference point is defined as X = 0, Y = 0.
Palletizing positions are calculated as points on the palletizing plane constituted by the reference point, PX-axis and PY-axis.
Accordingly, position data of the reference point must include valid coordinate components for PX-axis, PY-axis and PZ-axis (when palletizing Z-axis declaration is effective). If these coordinate components are invalid, an error occurs during palletizing position coordinate calculation for PAPG (Get palletizing calculation data) or other palletizing movement command. Coordinate components of other axes are ignored during palletizing position coordinate calculation.

An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur.
Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur.
Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

(Note 1) In the case of SCARA robots, executing a palletizing movement command while the work coordinate system selection number is set to 0 (base coordinate system) and this command is not yet set generates an error because the palletizing start position is (0, 0) and thus movement is disabled.

(Note 2) In the case of SCARA robots, the R-axis should be excluded from the effective axes, if already set in the position data, with a GRP command.
(This is not required if the R-axis field is blank.)
Set the R-axis data at the palletizing position using a PEXT command.

● PAPS (Set palletizing points) For 3-point or 4-point teaching

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PAPS	Position number	(Palletizing position setting type)	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set palletizing positions in 3-point teaching.
It can also be used to set palletizing positions in 4-point teaching, in which case the pallet plane can be set to any quadrilateral other than a square, rectangle or parallelogram.
In operand 1, set the position number of the start point needed to set palletizing positions in 3-point teaching. If "n" is set as the position number for the start point, position data for the end point in the PX-axis direction will be stored in position No. n+1, while position data for the end point in the PY-axis direction will be stored in position No. n+2.
In the case of 4-point teaching, position data for the end point should be stored in position No. n+3.

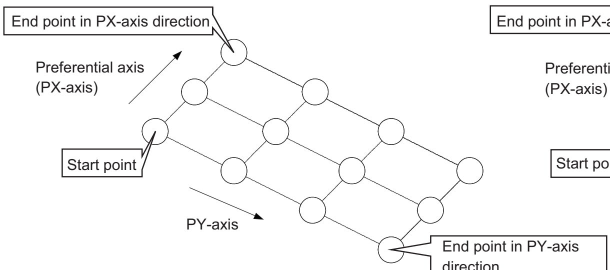
(Note 1) If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 "Indication Prohibited Axis Error" will occur. Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

In operand 2, specify the applicable palletizing position setting type.

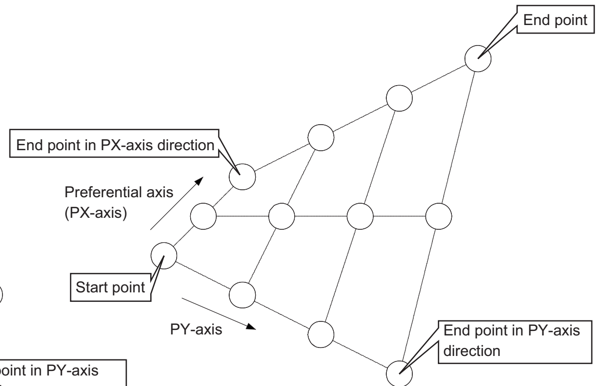
[Palletizing position setting type]
If operand 2 is "0" or blank, 3-point teaching will be specified.
As shown in Fig. 1 (a), palletizing positions will be set on the quadrilateral pallet plane determined by the three points including the start point, end point in the PX-axis direction and end point in the PY-axis direction.

If operand 2 is "2," 4-point teaching will be specified.
As shown in Fig. 1 (b), palletizing positions will be set on the quadrilateral pallet plane determined by the four points including the start point, end point in the PX-axis direction, end point in the PY-axis direction, and end point. Note, however, that whether the shape is planar or not varies depending on the end point data.

Fig. 1 shows two different arrangements of palletizing positions.



(a) 3-point teaching



(b) 4-point teaching

Fig. 1 Layout of Palletizing Positions

(Note) Since ASEL, PSEL and SSEL controllers are 2-axis controllers, setting 2 in operand 2 results in the planar type, just like 1 is set.

<u>If palletizing positions are set by 4-point teaching, it is recommended that the non-planar type be specified as long as all four points are known to be on the plane and the palletizing requires precision.</u>

If operand 2 is set to 1, 4-point teaching (planar type) is set.

Fig. 2-(a)

The plane is determined by three points including the start point, end point in PX-axis direction and end point in PY-axis direction. The end point is moved in parallel in PZ direction (vertical direction) and the point of intersection with the aforementioned plane is defined as the end point for this type of palletizing.

Palletizing positions are placed on the quadrilateral pallet surface determined by these four points.

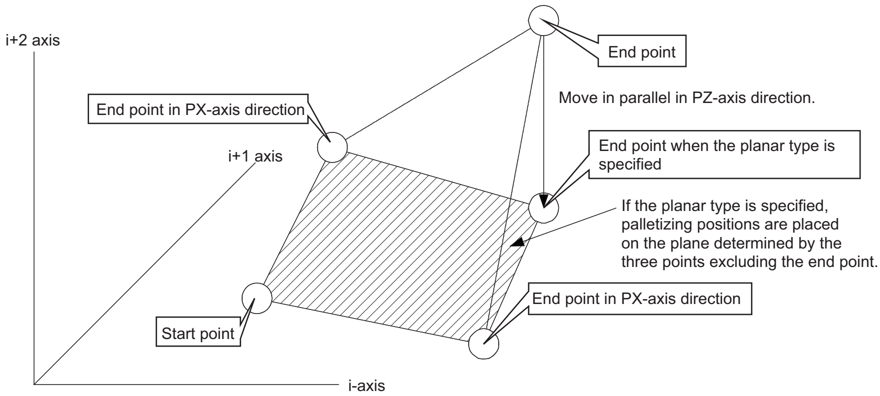


Fig. 2-(a)

Take note, however, that the moving direction of the end point varies if the three points other than the end point meet the conditions specified in Table 1. This is when the plane determined by the three points other than the end point is vertical to the ground. In this case, moving the end points in parallel with PZ direction (vertical direction) does not find a point of intersection with this plane.

Table 1 Moving Direction of End Point Based on Planar Type Specification

Condition	Moving direction of end point
Point data other than i-axis component matches among the three points other than the end point. (Refer to Fig. 2-(b))	Move in parallel in i-axis direction.
Point data other than PZ-axis component matches between the start point and end point in PX-axis direction. (Refer to Fig. 2-(c))	Move in parallel in the direction of one of the two axes other than the PZ-axis, whichever has the smaller axis number.
Point data other than PY-axis component matches between the start point and end point in PX-axis direction. (Refer to Fig. 2-(c))	
Point data other than PZ-axis component matches between the end point in PX-direction and end point in PY-axis direction. (Refer to Fig. 2-(c))	

* i indicates the axis number of one of the two axes other than the PZ-axis.

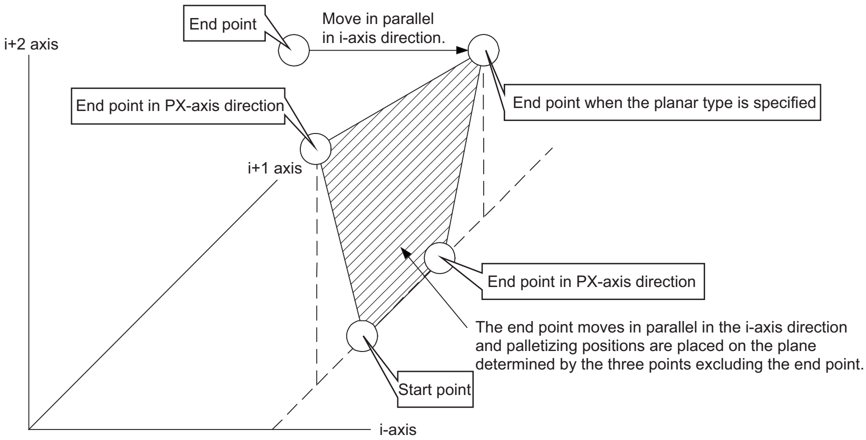


Fig. 2-(b)

The point data for i-axis component matches among the three points other than the end point:

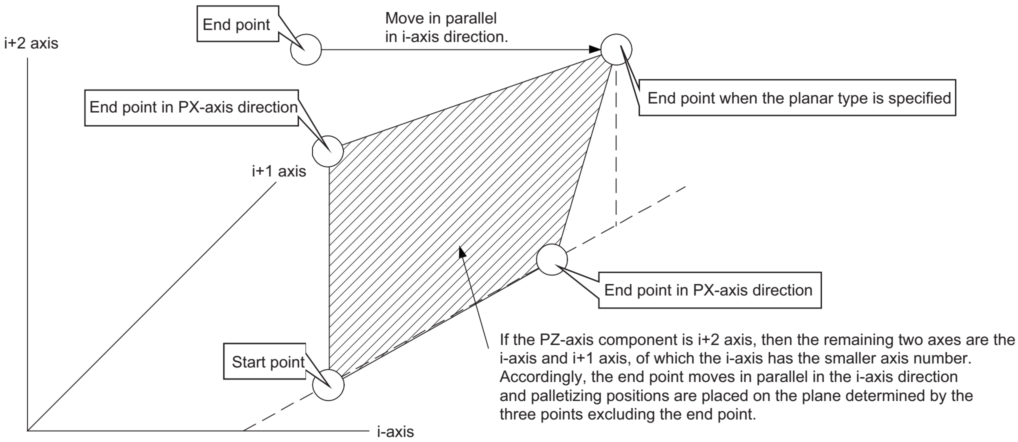


Fig. 2-(c)

The point data other than PZ-axis component matches between two of the three points other than the end point:
(In the figure above, the point data other than PZ-axis component matches between the start point and end point in PY-axis direction.)

- If the valid axis pattern does not match the point data for 3-point teaching or 4-point teaching, an error “CB0, Mismatched valid axes for palletizing 3-point teaching data” will generate. If a PAPS command is executed after specifying the applicable axes using a GRP command, only the point data corresponding to the specified axes, among all axes whose point data is valid, will be used as palletizing point data. Executing a GRP command thereafter with a different setting will have no effect.
- If the PZ-axis has been declared, there must be two effective axes other than the PZ-axis. If the PZ-axis is not yet declared, there must be two or three effective axes. If there are not enough effective axes, a “CAE: Insufficient effective axes for palletizing point data by 3-point teaching” occurs. If there are too many effective axes, on the other hand, a “CAF: Excessive effective axes for palletizing point data by 3-point teaching” occurs. If the planar type is specified and PZ-axis is not yet declared, set two effective axes. If the number of effective axes is other than 2, a “CB4: Arch motion Z-axis non-declaration error” occurs.
- This command cannot be used with a PASE (set palletizing axes) command. Whichever was set later will be given priority. (A single PAPS command can substitute a set of PASE, PAPT and PAST commands.)
- If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error, “CB5, BGPA not declared at palletizing setting” will generate.
- If the output field is specified, the output will turn ON after this command is executed.

● PSLI (Set zigzag)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PSLI	Offset amount	(Count)	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set a zigzag palletizing.
The value specified in operand 1 will be set as the offset amount for even-numbered rows.
The count specified in operand 2 will be set as the count for even-numbered rows.

[Refer to 3.6.5 Palletizing Function]

If operand 2 is not specified, the count for even-numbered rows will become the same as the count for odd-numbered rows.
If palletizing is set with PAPS (Set palletizing points) based on 3-point teaching, the PX and PY-axes need not be parallel with the corresponding axes on the work coordinate system. In this case, the offset direction is parallel with the PX-axis. If the offset value is positive, the measure in the direction of the PX-axis end point defines the offset. If the offset value is negative, the measure in the direction of the reference point defines the offset.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed

● PCHZ (Declare palletizing Z-axis): Only when there are 3 or more axes

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PCHZ	(Axis number)	Prohibited	CP

Applicable models
ASEL/PSEL/SSEL × Other than ASEL/PSEL/SSEL ○

Specify the axis number in palletizing Z direction.
Specify the axis number specified in operand 1 as the axis number in palletizing Z direction.
If operand 1 is not specified, the palletizing Z-axis which is specified and already declared becomes invalid.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed

(Note 1) Only Z-axis (either Axis No. 3 or Axis No. 7) in the work coordinate system is available to indicate for the palletizing Z-axis of the SCARA axes. Setting of the palletize Z-axis cannot be established for linear drive axes in PX/QX.

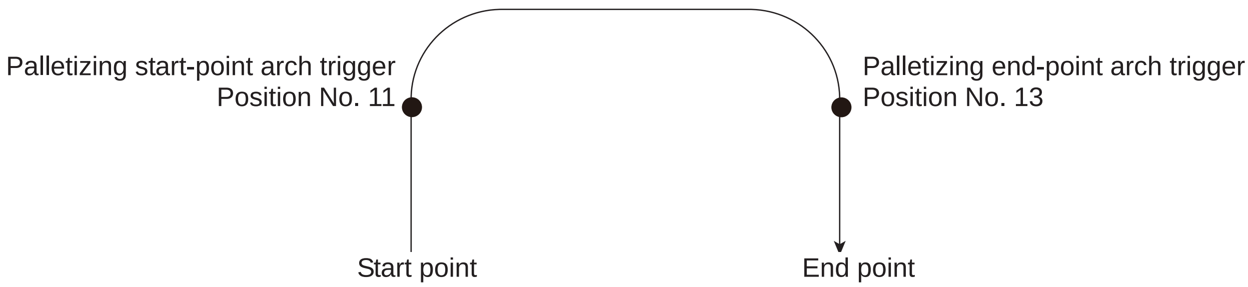
[Example] PCHZ 3

● PTRG (Set palletizing arch triggers)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTRG	Position number	Position number	CP

Applicable models
ASEL/PSEL/SSEL × Other than ASEL/PSEL/SSEL ○

Set arch triggers for arch motion to a palletizing point.
(This command is valid when a PACH command is executed.)
Set as the palletizing start-point arch trigger the palletizing Z-axis (PZ-axis) position data corresponding to the point data specified in operand 1, and set as the palletizing end-point arch trigger the PZ-axis position data corresponding to the point data specified in operand 2.



PTRG 11 13

[Refer to 3.6.5 Palletizing Function]

Among the point data, data of the PZ-axis specified by a PCHZ command must be effective.
Set the arch motion operation through palletizing points in such a way that when the axis rises from the start point, it starts parallel movement after reaching the start-point arch trigger, whereas, when descending, the axis completes parallel movement and then reaches the end-point arch trigger.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note 1) Setting of the palletize arch trigger cannot be established for linear drive axes in PX/QX.

● PEXT (Set composite palletizing)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PEXT	(Position number)	Prohibited	CP

Applicable models
ASEL/PSEL/SSEL × Other than ASEL/PSEL/SSEL ○

Set composite palletizing.
Set the position number specified in operand 1 for setting composite palletizing.
When a palletizing movement command is executed, effective axis data other than data of the PX and PY (and PZ) axes among the specified point data defines the end coordinate of the composite axis.
With SCARA robots, the R-axis becomes a composite palletizing axis.
If nothing is specified in operand 1, the position number already declared for setting composite palletizing becomes invalid.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note 1) The palletizing composition axes setting cannot be made to the linear drive axis for PX/QX.

● OFPZ (Set palletizing Z-axis offset)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	OFPZ	Offset value	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the offset in palletizing Z-axis direction.
Set the value specified in operand 1 as the offset in PZ-axis/palletizing Z-axis direction.
The setting unit of offset is mm. The effective resolution of the set value is 0.001mm.
A negative value can also be set for the offset within the range of operation.
This offset is effective only on the end point of PACH (Arch motion to palletizing point) operation.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note) Setting of the palletize Z-axis offset cannot be established for linear drive axes in PX/QX.

[22] Palletizing Calculation

● PTNG (Get palletizing position number)

Extension condition (LD,A,O,AB,OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PTNG	Palletizing number	Variable number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Assign the palletizing position number for the palletizing number specified in operand 1 to the variable specified in operand 2.
If the output field is specified, the output will turn ON after this command is executed.

● PINC (Increment palletizing position number by 1)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PINC	Palletizing number	Prohibited	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Increment by 1 the palletizing position number for the palletizing number specified in operand 1.
If the incremented value is considered normal as a palletizing position number calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully incremented, and turn OFF if the increment failed.

● PDEC (Decrement palletizing position number by 1)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PDEC	Palletizing number	Prohibited	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Decrement by 1 the palletizing position number for the palletizing number specified in operand 1. If the decremented value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully decremented, and turn OFF if the decrement failed.

● PSET (Set palletizing position number directly)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PSET	Palletizing number	Data	CC

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Set the value specified in operand 2 as the palletizing position number for the palletizing number specified in operand 1.
If the specified value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be set. If not, the value will not be set.
If the output field is specified, the output will turn ON when the palletizing position number was successfully updated, and turn OFF if the update failed.

● PARG (Get palletizing angle)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PARG	Palletizing number	Axis number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Obtain the palletizing angle.
Calculate the palletizing angle (degrees) from the physical axis specified in operand 2 for the palletizing number specified in operand 1, and store the result in variable 199.

This command need not be executed, if not necessary.
If this command is executed following PAPS (Set palletizing by 3-point teaching), the angle formed by the priority axis and specified axis on the work coordinate system is calculated automatically. An error occurs if this command is executed when PAPS is not yet executed or after PASE has been executed following PAPS.
The axis to be used with a GRP command can be specified before PAPS is executed (refer to the detailed explanation of PAPS). If the effective axis pattern for 3-point teaching data does not match, an "CB0: Mismatched effective axes for palletizing point data by 3-point teaching" error occurs.
If the number of effective point data axes (number of effective axes excluding the PZ-axis (palletizing Z-axis) if the PZ-axis is declared) is less than two, a "CAE: Insufficient effective axes for palletizing point data by 3-point teaching" error occurs. If the number of effective point data axes is greater than two, a "CB9: PX/PY-axis indeterminable error at acquisition of palletizing angle" occurs.
If the axis corresponding to the axis number in operand 2 does not specify one of the two valid axes associated with the point data, an error "CBA, Reference-axis/PX/PY-axis mismatch error at palletizing angle acquisition" will generate.
If the data other than PZ-axis component is identical between the reference point and PX-axis end point in 3-point teaching, a "Reference point/PX-axis end point identical error at acquisition of palletizing angle" occurs and angle calculation is disabled.
If the output field is specified, the output will turn ON after this command is executed.

● PAPG (Get palletizing calculation data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PARG	Palletizing number	Position number	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Store the position coordinate data of the palletizing point corresponding to the palletizing number specified in operand 1, under the position number specified in operand 2.

[23] Palletizing Movement

● PMVP (Move to palletizing points via PTP)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PMVP	Palletizing number	(Position number)	PE

Applicable models
All models [Refer to Section 5.1 for details of models]
○

Move to the calculated palletizing points via PTP.
The axes will move to the palletizing points specified in operand 1, via PTP.
Executing this command will not increment the palletizing position number by 1.

On controllers other than ASEL, PSEL and SSEL, movement does not occur in directions other than PX/PY-axis directions if the PX/PY-axis coordinates of palletizing points alone are effective (such as when the PZ-axis (palletizing Z axis) is not specified). If the PZ-axis coordinates of palletizing points are also effective, movement occurs in PZ-axis direction.
If a position number is specified in operand 2, however, the palletizing calculation result of Z-direction position is ignored and the axis moves to the height corresponding to the specified position number.
If data of any axis other than the Z-axis specified by palletizing is set under the position number specified in operand 2, such data is ignored. An error handling occurs if no PZ-axis data is available.
If composite palletizing is set, any axis whose data is available, other than the PX-axis and PY-axis (and PZ-axis), also operates.
If operand 2 is specified, the palletizing Z-axis must be declared (PCHZ) in the palletizing setting.
An error occurs if the palletizing Z-axis is not declared.

● PMVL (Move to palletizing points via interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PMVL	Palletizing number	(Position number)	PE

Applicable models
XSEL-JX/KX and MSEL-PCX/PGX × Other than XSEL-JX/KX and MSEL-PCX/PGX ○

Move to the calculated palletizing points via interpolation.
The axes will move to the palletizing points specified in operand 1, via interpolation.
Executing this command will not increment the palletizing position number by 1.

(Note 1) "Error No. B80 Indication Prohibited Axis Error" will be issued if the palletizing setting to operate the SCARA axes is indicated.
For the palletizing setting at PMVL movement, establish the setting to make the all of the operating axes the liner axes.

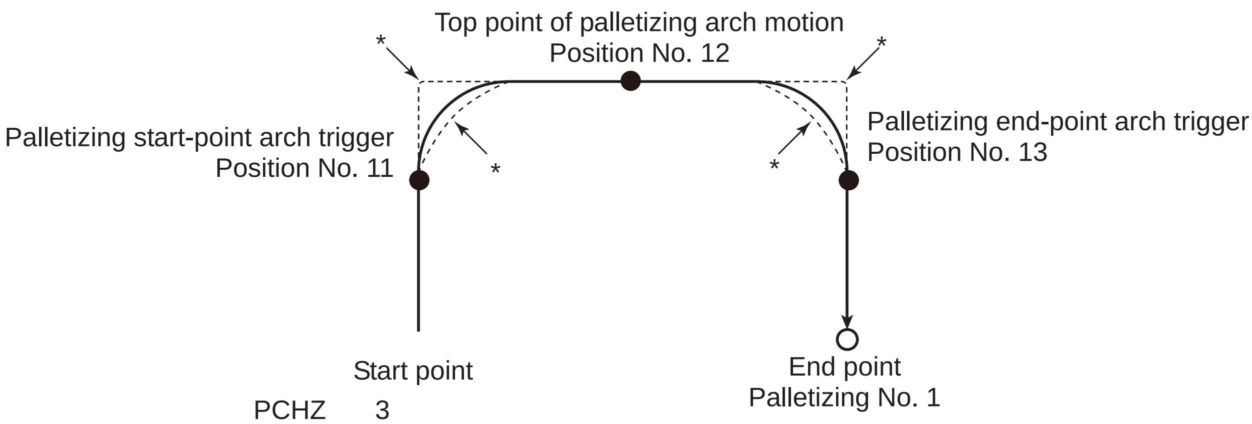
If a position number is specified in operand 2, however, the palletizing calculation result of Z-direction position is ignored and the axis moves to the height corresponding to the specified position number.
If data of any axis other than the Z-axis specified by palletizing is set under the position number specified in operand 2, such data is ignored. An error handling occurs if no PZ-axis data is available.
If composite palletizing is set, any axis whose data is available, other than the PX-axis and PY-axis (and PZ-axis), also operates.
Executing this command does not increment the palletizing position by 1.
If operand 2 is specified, the palletizing Z-axis must be declared (PCHZ) in the palletizing setting.
An error occurs if the palletizing Z-axis is not declared.

● PACH (Arch motion to palletizing point)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	PACH	Palletizing number	Position number	PE

Applicable models
ASEL/PSEL/SSEL × Other than ASEL/PSEL/SSEL ○

Perform arch motion from the current point to move to the palletizing points.
- Move via arch motion to the palletizing point specified in operand 1.
- Rise from the current point to palletizing start-point arch trigger and then start moving in PX/PY-axis directions. Pass the top point which is the Z point specified in operand 2, complete the movement in PX/PY-axis directions, and then reach the calculated palletizing point by passing near the palletizing end-point arch trigger.
- Palletizing arch triggers must be set for the PTRG command.




```
PCHZ    3
PTRG    11        13
   -
   -
   -
PACH    1         12
```

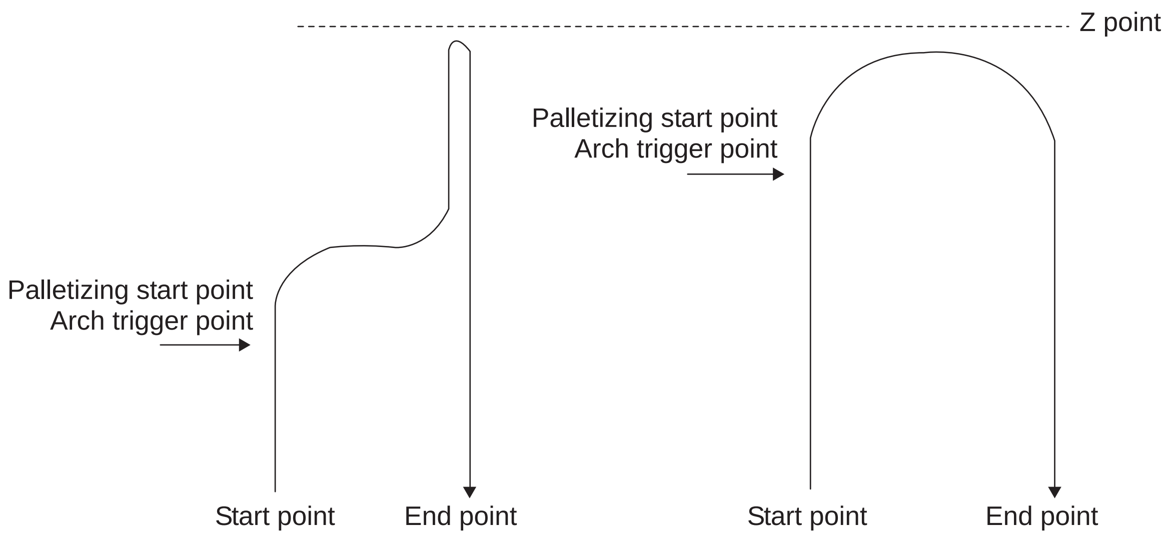
* When the operation is paused and then resumed, the rise operation → horizontal operation composite part and horizontal operation → rise operation composite part follow the paths denoted by * (dotted lines) in the figure depending on the position of resumption. Exercise caution to prevent contact.

(Note 1) When a palletizing setting with the SCARA axes and the liner axes existing together is indicated, “Error No. 421 SCARA and Liner Axes Simultaneous Indication Error” will occur. Also, when a palletizing setting with two units of SCARA axes existing together is indicated, “Error No. B80 Indication Prohibited Axis Error” will occur. Establish the setting to make all the operating axes the same SCARA axes or the linear axes for the palletizing setting at the palletizing point arch motion movement.

(Note 2) The palletize point arch motion operation of SCARA axis is PTP operation and the linear drive axis is CP operation.

(Note 3) The palletize point arch motion operation cannot be performed for linear drive axes in PX/QX.

- The PZ-axis coordinate of the end point corresponds to the PZ-axis component of the position coordinate of the palletizing point, if any, plus the palletizing Z-axis offset. If the PZ component is not available, then the PZ-axis coordinate of the start point, plus the palletizing Z-axis offset, is used. (Normally the offset is added to all applicable positions such as arch trigger and Z points.)
- An error occurs if the palletizing start-point arch trigger is set below the start point, or palletizing end-point arch trigger is set below the end point. (Note: “Above” and “below” have nothing to do with the positive and negative directions of coordinates.)
- The PZ-axis up direction refers to the direction of moving from the start point to Z point (or opposite direction in the case of down direction) and has nothing to do with the magnitude correlation of coordinate values. Accordingly, always check the actual operating directions when this command is used.
- PZ-axis down operation is performed after an up process command value has been output. Accordingly, the following operations may take place depending on how the palletizing arch trigger and Z points are set.



In these cases, change the palletizing arch triggers and PZ point to increase the efficiency of operation.
- If composite palletizing is set (PEXT), any axis whose data is available, other than the PX, PY and PZ-axes, also operates. However, the composite axis starts/ends its operation at a position above the applicable arch trigger. If the R-axis is set with a PEXT command, the R-axis starts/ends its operation above the applicable arch trigger.
- Executing this command does not increment the palletizing position by 1.

[24] Building of Pseudo-Ladder Task

● CHPR (Change task level)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	CHPR	0 or 1	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Specify "1" (User HIGH) if you wish the target task to be processed before other tasks.
This command can also be used with non-ladder tasks.
Task level change (0: User NORMAL, 1: User HIGH) is not a required component, but specifying User HIGH will require a TSLP command explained below.
(Without TSLP, tasks of the User NORMAL level will not be processed.)

● TPCD (Specify processing to be performed when input condition is not specified)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	TPCD	0 or 1	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Specify the processing to be performed when input condition is not specified.
(0: Execute, 1: Follow the input condition in the last executed step)
In a ladder task, always input “1” (Follow the input condition in the last executed step) in operand 1.
In a non-ladder task, always input “0” (Execute). (The default value is “0”.)

● TSLP (Task sleep)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Prohibited	Prohibited	TCLP	Time	Prohibited	CP

Applicable models
All models [Refer to Section 5.1 for details of models]
○

[Function] Set the time during which the applicable task will sleep, in order to distribute the processing time to other tasks.
If the task level is set to User HIGH, this command must always be specified.
The applicable task will sleep during the set time.
The time in operand 1 is set in msec.
An appropriate time setting must be examined on the actual system. (Normally approx. 1 to 3 is set.)
(If the ladder statement becomes long, state this command multiple times between steps, as necessary.)
This command can also be used with non-ladder tasks.

[25] Extended Command

● ECMD1 (Get motor current value (as percentage of rated current))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	1	Axis number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	○	×	×

[Function] Store the motor current value (percentage of the rated current) corresponding to the "axis number" specified in operand 2, in variable 99.

Note:
- The current value data (percentage of the rated current) obtained by this command has been processed by feedback current filtering and includes analog error.
  When comparing with "Constant (Non-Pressing) Torque Limit (Upper)" set in Extension Command Code 250, have 5% or more of a margin.

[Example] ECMD 1 2 Extended command 1
Store the motor current value (percentage of the rated current) of axis 2, in variable 99.

● ECMD2 (Get home sensor status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	2	Axis number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	×	×	×

[Function] Reflect in the output the status of the home sensor corresponding to the “axis number” specified in operand 2.

Note:

- The acquired home sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 14, “Home sensor input polarity”.
  If 0 (Not used) is set in axis-specific parameter No. 14, “Home sensor input polarity”, the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example]

ECMD	2	3	315	Output the home sensor status of axis 1 in output port No. 315.

● ECMD3 (Get overrun sensor status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	3	Axis number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	×	×	×

[Function] Reflect in the output the status of the overrun sensor corresponding to the "axis number" specified in operand 2.

Note:

- The acquired overrun sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 15, "Overrun sensor input polarity".
  If 0 (Not used) is set in axis-specific parameter No. 15, "Overrun sensor input polarity", the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example] ECMD 3 1 890 Output the overrun sensor status of axis 1 in global No. 890.

● ECMD4 (Get creep sensor status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	4	Axis number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	×	×	×

[Function] Reflect in the output the status of the creep sensor corresponding to the “axis number” specified in operand 2.

Note:

- The acquired creep sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 16, “Creep sensor input polarity”.
  If 0 (Not used) is set in axis-specific parameter No. 16, “Creep sensor input polarity”, the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example] ECMD 4 2 315 Output the creep sensor status of axis 2 in output port No. 315.

● ECMD5 (Get axis operation status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	5	Axis number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	○	×	×

[Function] Store the status of the axis specified in operand 2, in variable 99.
The axis status is indicated by the ON/OFF level of each bit, as shown below.
Accordingly, the obtained value must be converted to a binary value for interpretation.

Variable 99							
7	6	5	4	3	2	1	0
/	/						

- Bit 0: Servo axis in use (0 = Not in use, 1 = In use)
- Bits 2–1: Home return (00 = Not yet complete, 01 = In progress, 10 = Complete)
- Bit 3: Servo ON/OFF (0 = OFF, 1 = ON)
- Bit 4: Successful execution of movement command (0 = Not yet complete, 1 = Successful)
- Bit 5: Detection of missed work during push-motion operation (0 = Not detected, 1 = Detected)

(Note) If an invalid axis number is specified in operand 2, "C44, SEL data error" will generate.

[Example] ECMD 5 2 Store the status of axis 2 in variable 99. If 28 (decimal value) was stored in variable 99 after the command was executed, the status of axis 2 is interpreted as follows.

Variable 99
28

Binary notation

Variable 99							
0	0	0	1	1	1	0	0

Status of axis 2:
- Servo axis not in use
- Home return complete
- Servo ON
- Movement command successful
- Missed work not detected during push-motion operation

● ECMD6 (Dedicated SCARA commands/Current position acquirement on each axis system (1 axis direct)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	6	Integer Variable number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	×	×	×	○	×	×	×	○ (PCX/PGX only)

[Function] By using data stored in the four integer variables in a row from the integer variable number indicated in Operation 2, the current position expressed in each axis coordinate system of the indicated axis numbers gets read out to the variable indicated in the current position storage variable number.

• When Operand 2 = Variable number

Variable No.	Description of setting	I/O
n	Axis Number	
n+1	Current Position Storage Variable Number	
n+2	0	Reserved (to be fixed to 0)
n+3	0	Reserved (to be fixed to 0)

(Note 1) Input an integer variable number in Operation 2.
Local area : 1 to 96, 1001 to 1096
Global area : 200 to 296, 1200 to 1296

(Note 2) The units in the result of the readout of the current position for each axis system are as shown below.
SCARA 1st, 2nd and 4th Axes : deg. (degrees)
SCARA 3rd Axis : mm

[Example]

LET	200	4	Set the 4th axis (R-axis) to Variable No. 200
LET	201	300	Set Current Position Storage Variable No. (300) to Variable No. 201
LET	202	0	Set 0 to Variable No. 202
LET	203	0	Set 0 to Variable No. 203
ECMD	6	200	The current position of each coordinate system on R-axis is read out to Variable No. 300.

● ECMD20 (Get parameter value)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	20	Variable number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	○	○	×	○

[Function] Store the value of the specified parameter in variable 99, using the data stored in the three consecutive variables starting from the one corresponding to the variable number specified in operand 2.
The contents and ranges for the variable data settings are as shown below. Setting outside the specified range will generate “C44 SEL data error”.

• When Operand 2 = n

Variable No.	Description of setting	Setting value and range for each variable						
		I/O	Common to all axes	Axis-specific	Driver	Encoder	I/O device	Other
n	Parameter type	0	1	2	3	4	5	7
n+1	Device number/axis number	0	0	1 to 8* (up to number of connected axes)	1 to 8* (up to number of connected axes)	1 to 8* (up to number of connected axes)	0 to 9	0
n+2	Parameter number	1 to 999	1 to 400	1 to 250	1 to 112	1 to 30	1 to 112	1 to 200

Specify an integer variable in operand 2 (integer variables 98, 99, 298, 299, 1098, 1099, 1298 and 1299 cannot be specified, because three consecutive integer variables cannot be allocated if any of these integer variables is specified). If a variable of non-integer type is specified, “C3C, Variable number error” will generate.

(Note) Setting of Parameter Type = 10 enables to acquire parameters for the pulse I/O board. [See the next page.]

[Example]

LET	1250	0	Variable No. 1250 = Parameter type (I/O)
LET	1251	0	Variable No. 1251 = Device number (0, in the case of I/O parameter)
LET	1252	30	Variable No. 1252 = Parameter number (No. 30)
ECMD	20	1250	Extended command 20 (Use variable No. 1250 through 1252) Store the value of I/O parameter No. 30, “Input function selection 000”, in variable 99.

● ECMD20 (Get parameter value (Extension Motion Control Board Parameters)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	20	Variable number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] Store the value of the specified parameter in variable 99, using the data stored in the five consecutive variables starting from the one corresponding to the variable number specified in Operand 2.
The contents and ranges for the variable data settings are as shown below. Setting outside the specified range will generate "C44 SEL data error".
When Parameter Type = 10 (Pulse I/O board: Feature dedicated for XSEL) is selected, Parameter Detail Type (n + 3) and Channel Number (n + 4) are required.

• When Operand 2 =n

<table>
<tr><th>Variable No.</th><th>Description of setting</th><th>Setting value and range for each variable</th></tr>
<tr><td>n</td><td>Parameter type</td><td>10: Pulse I/O board</td></tr>
<tr><td>n+1</td><td>Device number</td><td>0 to 1</td></tr>
<tr><td>n+2</td><td>Parameter number</td><td>1 to 100</td></tr>
<tr><td>n+3</td><td>Parameter detail type</td><td>0 : Common
1 : Input Channel
2 : Output Channel</td></tr>
<tr><td>n+4</td><td>Channel number</td><td>The range may differ depending on parameter detail type (n+3).<table>
<tr><th>Parameter detail type (n+3)</th><th>Range for channel number (n+4)</th></tr>
<tr><td>0 (Common)</td><td>Reserved (to be fixed to 0)</td></tr>
<tr><td>1 (Input channel)</td><td>0 to 1</td></tr>
<tr><td>2 (Output channel)</td><td>0 to 7</td></tr>
</table></td></tr>
</table>

Specify an integer variable in operand 2. At this time, make sure to secure five (for XSEL) consecutive variables. C3C will be generated when a variable other than integer variables is indicated.

[Example]

LET	1250	10	Variable No. 1250 = Parameter type (Pulse I/O board)
LET	1251	1	Variable No. 1251 = Device number
LET	1252	2	Variable No. 1252 = Parameter number (No. 2)
LET	1253	2	Variable No. 1252 = Parameter Detail Type (Output Channel)
LET	1253	5	Variable No. 1252 = Channel number (Output channel 5)
ECMD	20	1250	Extended command 20 (Use variable No. 1250 through 1252) Data for the value set in No. 2 (Pulse output mode) in the pulse I/O board parameter output channel 5 is stored to Variable 99.

● ECMD250 (Set torque limit/torque limit over detection time)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	ECMD	250	Axis pattern	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Set the steady-state (non-push) torque limit (upper limit)/steady-state (non-push) torque limit over detection time. Use the data stored in three successive integer variables, starting from the integer variable number specified in operand 2, to temporarily change the applicable parameters (including internal parameters).

Operand 2 = n

Variable No. n ----- Target axis pattern (decimal entry)
* Example of decimal entry: 1 = Axis 1 only
2 = Axis 2 only
3 = Axes 1 and 2
7 = Axes 1, 2 and 3
15 = Axes 1, 2, 3 and 4

Variable No. n+1 = Set value of steady-state (non-push) torque limit (upper limit) (1% or more of the rating to the value set in driver card parameter No. 40, "Maximum torque limit (%)")
* If the set value is greater than the upper limit specific to each axis, the upper limit specific to the axis is set.

Variable No. n+2 = Set value of steady-state (non-push) torque limit over detection time
(0 to 20000msec)
* Set 1 or greater if you want to use this command to "detect a contact/heavy load" or move an axis.
* If 0 is set, the detection time becomes invalid (infinite). This setting is used mainly to "limit the torque of the supporting axis (horizontal only) in fitting application". If 0 (infinite) is set, the "steady-state (non-push) torque limit (upper limit)" is limited to a maximum of 70% to prevent overheating.

Variable No. n+3 = 0 is set. (Reserved. * May be made accessible in the future.)

Variable No. n+4 = 0 is set. (Reserved. * May be made accessible in the future.)
If a command specifying the "steady-state (non-push) torque limit (upper limit)" has remained effective for the "steady-state (non-push) torque limit over detection time" or longer in steady state (not pushing), appropriate processing is performed based on the parameter below. Note that processing based on the following parameter is not performed if the "steady-state (non-push) torque limit over detection time" is set to 0 (infinite):
All-axis parameter No. 19, "Type of processing upon steady-state (non-push) torque limit over (priority on overload and other driver errors)"
0: Operation-cancellation level error (Recommended)
(Error No. 420: Steady-state (non-push) torque limit over error)
1: Operation cancellation (SEL command output = OFF)

[Example 1]	LET	290	3	Set the target axis pattern (axes 1 and 2) in integer variable 290.
	LET	291	80	Set the steady-state torque limit in integer variable 291.
	LET	292	1000	Set the steady-state torque limit over detection time in integer variable 292.
	ECMD	250	290	Read the values of three successive variables, starting from variable 290. Set axes 1 and 2. Steady-state torque limit = 80%, steady-state torque limit over detection time = 10000msec
	MOVP	2		Move to position No. 2 under the condition set by ECMD250.

* To return to a normal state:

[Example 2]	LET	290	3	Set the target axis pattern (axes 1 and 2) in integer variable 290.
	LET	291	1000	Set the steady-state torque limit in integer variable 291 (specification of the upper limit specific to each axis).
	LET	292	20000	Clear the steady-state torque limit over detection time in integer variable 292. (Clear 20000.)
	STOP	*290		Clear the low-torque axis deviation counter.
	ECMD	250	290	Read the values of three successive variables, starting from variable 290. Steady-state torque limit = Upper limit specific to each axis (maximum torque return) Steady-state torque limit over detection time (20000msec)
	MOVP	2		Move to position 2 at the steady-state torque.

(Note 1) If the torque is set low, dropping (vertical axis, etc.) and overshooting occurs. If the torque is lowered during high-speed operation, overshooting occurs due to insufficient torque.

(Note 2) If the torque is lowered during high-speed operation, normal deceleration cannot be performed due to insufficient torque and overshooting occurs as a result, creating a dangerous situation.

(Note 3) If positioning operation is performed at low torque, the axis may remain stopped near the positioning target due to insufficient torque. When moving an axis, be sure to set the “steady-state (non-push) torque limit over detection time” to 1msec or longer to detect a steady-state (non-push) torque limit over event (timeout).

* If the “steady-state (non-push) torque limit over detection time” is set to 0 to “limit the torque of the supporting axis (horizontal only) in fitting application”, positioning operation to the coordinate of the torque-limited axis is performed when returning after the fitting operation, if the position data for return operation after the fitting operation (via a PUSH command, etc.) includes the coordinate of the supporting axis (torque-limited axis) in fitting application. As a result, the axis may remain stopped near the target position due to insufficient torque. For the position data used in the return operation after the fitting operation, set only the coordinate of the fitting operation axis (axis used by a PUSH command, etc.).

(Note 4) If the torque is set extremely low, servo ON axes may move at very slow speed due to an analog offset error, etc.

(Note 5) Even when the load is normal, the torque becomes slightly higher during acceleration/deceleration. Determine appropriate settings (steady-state torque limit and steady-state torque limit over detection time) to prevent false detection of steady-state torque limit over events.

(Note 6) “Error No. C6B: Deviation overflow error” or “Error No. CA5: Stop deviation overflow error” may be detected before “Error No. 420: Steady-state (non-push) torque limit over error”. This is normal.

(Note 7) If the torque is changed to a high level from a low level at which axis movement can no longer be guaranteed, be sure to issue a STOP command to low-torque axes and clear the deviation counter before increasing the torque (from a low level). If the torque setting is changed from low to high when deviations are still accumulated, the axes may move without their speed being limited and thus a dangerous situation may occur.

(Note 8) To return to the normal condition (maximum torque), expressly specify 1000% for the “steady-state (non-push) torque limit (upper limit)” and 20000msec for the “steady-state (non-push) torque limit over detection time”.
* If a value greater than the upper limit specific to each axis is set for the “steady-state (non-push) torque limit (upper limit)” of that axis, the upper limit specific to the axis (approx. 200 to 400%) is set.

(Note 9) The following values are used upon power ON reset, software reset and start of home return:
Steady-state (non-push) torque limit (upper limit) = Driver card parameter No. 40, “Maximum toque limit (%)”
Steady-state (non-push) torque limit over detection time = 20000msec

(Note 10) If the “steady-state (non-push) torque limit (upper limit)” and “steady-state (non-push) torque limit over detection time” are changed, the new settings will remain effective even after the SEL program ends. When building a system using this extended command, therefore, expressly set the “steady-state (non-push) torque limit (upper limit)” and “steady-state (non-push) torque limit over detection time” in all SEL programs, before any operation is started in each program, using this extended command. If you assume that the “steady-state (non-push) torque limit (upper limit)” and “steady-state (non-push) torque limit over detection time” will be reset after the end of operation in other programs, unexpected settings of “steady-state (non-push) torque limit (upper limit)” and “steady-state (non-push) torque limit over detection time” may be applied should the program abort due to an error, etc., in which case unforeseen problems may result.

(Note 11) This extended command does not rewrite the value of driver card parameter No. 40, “maximum torque limit” itself (main CPU flash memory) (in non-volatile memory).

[26] RC gateway function commands

● RPGT (Read RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPGT	RC-axis number	Position number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read the RC-axis position into variable 199.

[Example 1] RPGT 1 2 Read the position corresponding to RC position No. 2 of axis 1 into variable 199.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	200.00	300	0.3	0	0.10

→ 200.00 is stored in variable 199.

[Example 2]

LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
RPGT	*1	*2	Read into variable 199 the RC position corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2.

● RPPT (Write RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPPT	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Write the value of variable 199 to the position corresponding to the specified position data [mm].

[Example 1]

```
LET   199  150   Assign 150 to variable 199.
RPPT  1    2     Write the content of variable 199, or 150, to RC
                 position No. 2 of axis 1.
```

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	150.00	300	0.3	0	0.10

Variable 199	150

[Example 2]

```
LET   199  15    Assign 150 to variable 199.
LET   1    2     Assign 2 to variable 1.
LET   2    3     Assign 3 to variable 2.
RPPT  *1   *2    Write the content of variable 199, or 150, to the RC
                 position corresponding to the content of variable 2,
                 or 3, of the axis corresponding to the content of
                 variable 1, or 2.
```

● RPCR (Clear RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPCR	RC-axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Clear position data in the range specified by variable No. n and variable No. n+1. After the data is cleared, the fields become blank.

Variable	Description of setting
n	Clear start position number
n+1	Clear end position number

[Example 1]

LET	200	0	Assign 0 to variable 200.
LET	201	1	Assign 1 to variable 201.
RPCR	1	200	Clear 1 from position No. 0 of axis 1.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp	
0						Cleared.
1						
2	200.00	300	0.3	0	0.10	

● RPCP (Copy RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPCP	RC-axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Copy the position data specified by variable No. n and variable No. n+1.

Variable	Description of setting
n	Position number to copy data to
n+1	Position number to copy data from

[Example 1]

LET	200	2	Assign 2 to variable 200.
LET	201	0	Assign 0 to variable 201.
RPCP	1	200	Copy the data of position No. 0 of axis 1 specified by the variable, to position No. 2.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	100	0.2	0	0.20
1	380.00	300	0.3	0	0.10
2	5.00	100	0.2	0	0.20

Copy

● RPRD (Read current RC-axis position)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPRD	Position number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read into a position number the current position of each axis specified by an RAXS command.

Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a “(43B) RC-axis pattern not-set error” occurs.

[Example 1]

RAXS	0	11	Set an axis pattern consisting of axes 0, 1 and 2.
RPRD	100		Read the current positions of axes 0 to 2 into RC position No. 100.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
100	5.00	300	0.3	0	0.10

The current position of axis 1

Position data of axis 2

No.	Pos	Vel	Acc	Push	Inp
100	500.00	200	0.3	0	0.10

The current position of axis 2

Position data of axis 3

No.	Pos	Vel	Acc	Push	Inp
100	100.00	300	0.3	0	0.10

The current position of axis 3

[Example 2]

RAXS	0	111	Set an axis pattern consisting of axes 0, 1 and 2.
LET	1	100	Set 100 in variable 1.
RPRD	*1		Read the current positions of axes 0 to 2 into the RC position corresponding to the content of variable 1, or 100.

● RPRQ (Read current RC-axis position (single-axis direct))

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPDQ	RC-axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read the current position of the RC-axis into the variable specified in operand 2. The current position can be acquired faster than when a RPRD command is used.

[Example] RPRQ 2 100 Read the current position of axis 2 into variable No. 100.

[26] RC gateway function commands

● RPVL (Write RC-axis speed data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPVL	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Write the value of variable 199 to the speed [mm/s] corresponding to the position data specified in operand 2.

[Example 1]

LET	199	100	Assign 100 to variable 199.
RPVL	1	2	Write the speed in variable 199, or 100mm/s, to RC position No. 2 of axis 1.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	200.00	100	0.3	0	0.10

Variable 199	100

[Example 2]

LET	199	100	Assign 100 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
RPVL	*1	*2	Write the speed in variable 199, or 100mm/s, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2.

● RPAD (Write RC-axis acceleration/deceleration data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPAD	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Write the value of variable 199 to the acceleration/specification [G] corresponding to the position data specified in operand 2.

[Example 1]

LET	199	0.1	Assign 0.1 to variable 199.
RPAD	1	2	Write the acceleration/specification in variable 199, or 0.1G, to RC position No. 2 of axis 1.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	200.00	300	0.1	0	0.10

Variable 199	0.1

[Example 2]

LET	199	0.3	Assign 0.3 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
RPAD	*1	*2	Write the speed in variable 199, or 0.3G, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2.

● RPIP (Write RC-axis in-position width data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPIP	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Write the value of variable 199 to the in-position width [mm] corresponding to the position data specified in operand 2.

[Example 1]

LET	199	0.2	Assign 0.2 to variable 199.
RPIP	1	2	Write the in-position band in variable 199, or 0.2mm, to RC position No. 2 of axis 1.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	200.00	300	0.3	0	0.20

Variable 199	0.2

[Example 2]

LET	199	0.2	Assign 0.2 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
RPIP	*1	*2	Write the in-position width in variable 199, or 0.2mm, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2.

● RPTQ (Write RC-axis current-limiting value data for push-motion operation)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPTQ	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Write the value of variable 199 to the current-limiting value for push-motion operation [%] corresponding to the position data specified in operand 2.

[Example 1]

LET	199	50	Assign 50 to variable 199.
RPTQ	1	2	Write the current-limiting value in variable 199, or 50%, to RC position No. 2 of axis 1.

Position data of axis 1

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.3	0	0.10
2	200.00	300	0.3	50	0.10

Variable 199	50

[Example 2]

LET	199	50	Assign 50 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
RPTQ	*1	*2	Write the current-limiting value in variable 199, or 50%, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2.

● RGVL (Read RC-axis speed data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RGVL	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read into variable 199 the speed [mm/s] corresponding to the position data specified in operand 2.

[Example] RGVL 2 1 Read into variable 199 the speed specified under RC position No. 1 of axis 2.

Position data of axis 2

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	200	0.3	0	0.10

Variable 199	200

● RGAD (Read RC-axis acceleration/deceleration data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RGAD	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read into variable 199 the acceleration/deceleration [G] corresponding to the position data specified in operand 2.

[Example 1] RGAD 2 1 Read into variable 199 the acceleration/deceleration specified under RC position No. 1 of axis 2.

Position data of axis 2

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.2	0	0.10

→

Variable 199	0.2

[26] RC gateway function commands

● RGIP (Read RC-axis in-position width data)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RGIP	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read into variable 199 the in-position width [mm] corresponding to the position data specified in operand 2.

[Example] RGIP 2 1 Read into variable 199 the in-position width specified under RC position No. 1 of axis 2.

Position data of axis 2

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.2	0	0.10

Variable 199	0.10

● RGTQ (Read RC-axis current-limiting value data for push-motion operation)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used[*1]

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RGTQ	RC-axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read into variable 199 the current-limiting value for push-motion operation [%] corresponding to the position data specified in operand 2.

[Example] RGTQ 2 1 Read into variable 199 the current-limiting value specified under RC position No. 1 of axis 2.

Position data of axis 2

No.	Pos	Vel	Acc	Push	Inp
0	5.00	300	0.3	0	0.10
1	380.00	300	0.2	30	0.10

Variable 199	30

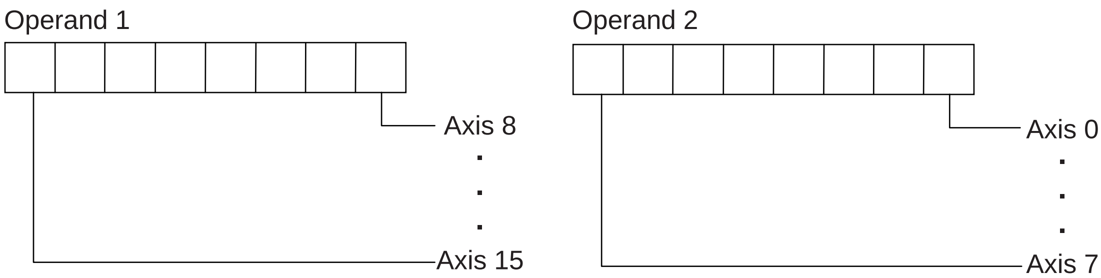
● RAXS (Set RC-axis pattern)

RC position-data use mode		
	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RAXS	Axis pattern, upper	Axis pattern, lower	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Set an axis pattern covering axes 8 to 15 in operand 1, and axis pattern covering axes 0 to 7 in operand 2.
The axes set by the axis pattern are operated simultaneously.



Always set an axis pattern if the commands listed below are used.
(Set 1 for the axis numbers used, and 0 for the axis number not used.)
If an axis pattern is not set, a "(43B) RC-axis pattern not-set error" occurs:

- RPRD : Read current RC-axis position
- RSON : Turn ON RC-axis servo
- RSOF : Turn OFF RC-axis servo
- RHOM : Return RC-axis to home
- RMVP : Move RC-axis by position specification
- RMPI : Move RC-axis incrementally by position specification
- RSTP : Decelerate RC-axis to stop

[Example]	RAXS	1010101	10101010	Set an axis pattern consisting of axes 1, 3, 5, 7, 8, 10, 12 and 14.
	RSON			Turn ON the servos of the specified axes.
	RMVP	20		Move the specified axes to the positions corresponding to position No. 20.

● RSON (Turn ON RC-axis servo)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RSON	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Turn ON the servo of each RC-axis specified by an RAXS command.

> Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example] RAXS 0 1100 Set an axis pattern that uses axes 2 and 3.

RSON Turn ON the servos of the specified axes.

● RSOF (Turn OFF RC-axis servo)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RSOF	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Turn OFF the servo of each RC-axis specified by an RAXS command.

> Important note: Before executing this command, set an axis pattern using an RAXS command.
> If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example]

RAXS	0	1100	Set an axis pattern consisting of axes 2 and 3.
RSOF			Turn OFF the servos of the specified axes.

[26] RC gateway function commands

● RHOM (Return RC-axis to home)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RHOM	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Return each RC-axis specified by an RAXS command to its home.
The servo of the axis to be returned home turns ON automatically.

[Example]	RAXS	0	1100	Set an axis pattern consisting of axes 2 and 3.
	RHOM			Return the specified axes to their home.

● RMVP (Move RC-axis by position specification)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RMVP	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Move each RC-axis specified by an RAXS command to the position number in operand 1. The output turns OFF when the axis movement is started, and turns ON when completed.

⚠ Caution: The specific operation varies between the XSEL position-data use mode and RC position-data use mode.
1) XSEL position-data use mode
→ Move via PTP to the position corresponding to the position number in operand 1.
2) RC position-data use mode
→ The specific operation varies depending on the position data in the RC controller.

No.	Position data item in RC		
	Push-motion	Incremental	Description of operation
1	0	0	Move via PTP to the position corresponding to the position number in operand 1.
2	0	1	Move incrementally (via PTP) by the travel corresponding to the position number in operand 1.
3	Other than 0	0	Move to the position corresponding to the position number in operand 1 and then perform push-motion operation. The output turns OFF if any one axis has been pushed missed the load.
4	Other than 0	1	Move to the position corresponding to the position number in operand 1 and then perform push-motion operation. The output turns OFF if any one axis has been pushed and missed the load.

ⓘ Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a “(43B) RC-axis pattern not-set error” occurs.

[Example 1]	RAXS	0	11	Set an axis pattern consisting of axes 0 and 1.
	RMVP	10		Move the specified axes to the positions corresponding to position No. 10.
[Example 2]	RAXS	0	11	Set an axis pattern consisting of axes 0 and 1.
	LET	1	10	Assign 10 to variable 1.
	RMVP	*1		Move the specified axes to the positions corresponding to position No. 10 according to the content of variable 1, or 10.

● RMPI (Move RC-axis incrementally by position specification)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RMPI	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Move each RC-axis specified by an RAXS command by the travel corresponding to the position data number in operand 1.
The output turns OFF when the axis movement is started, and turns ON when completed.

Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a “(43B) RC-axis pattern not-set error” occurs.

[Example 1]

RAXS	0	11	Set an axis pattern consisting of axes 0 and 1.
RHOM	10		Move by the travel corresponding to position No. 10.

[Example 2]

RAXS	0	11	Set an axis pattern consisting of axes 0 and 1.
LET	1	10	Assign 10 to variable 1.
RMPI	*1		Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10.

(Note 1) 405 RC Gateway Communication Type Error will occur in Fieldbus Type.

● RMVD (Move RC-axis absolutely by direct numerical specification of position)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode		
RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RMVD	RC-axis number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Perform absolute position movement using the values in variable No. n to variable No. n+3.
The output turns OFF when the axis movement is started, and turns ON when completed.

Variable number	Description of setting
n	Target position
n+1	Speed [mm/s]
n+2	Acceleration/deceleration [G]
n+3	In-position width [mm]

[Operand 1 setting type]

Operand 1	Specification of execution axis
0 to 15	The axis corresponding to the specified RC-axis number performs absolute position movement.
-1	Each RC-axis specified by an RAXS command performs absolute position movement.

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

[Example]

LET	300	100	Set the target position to 100mm.
LET	301	200	Set the speed to 200mm/s.
LET	302	0.3	Set the acceleration/deceleration to 0.3G.
LET	303	0.1	Set the in-position width to 0.1mm.
RMVD	1	300	Move RC-axis 1 absolutely to the specified position.

● RMDI (Move RC-axis incrementally by direct numerical specification of position)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RMDI	RC-axis number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Perform incremental position movement using the values in variable No. n to variable No. n+3.
The output turns OFF when the axis movement is started, and turns ON when completed.

Variable number	Description of setting
n	Travel
n+1	Speed [mm/s]
n+2	Acceleration/deceleration [G]
n+3	In-position width [mm]

[Operand 1 setting type]

Operand 1	Specification of execution axis
0 to 15	The axis corresponding to the specified RC-axis number performs incremental position movement.
-1	Each RC-axis specified by an RAXS command performs incremental position movement.

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

[Example]

LET	300	50	Set the travel to 50mm.
LET	301	200	Set the speed to 200mm/s.
LET	302	0.3	Set the acceleration/deceleration to 0.3G.
LET	303	0.1	Set the in-position band to 0.1mm.
RMDI	1	300	Move RC-axis 1 incrementally to the specified position.

(Note 1) 405 RC Gateway Communication Type Error will occur in Fieldbus Type.

● RPUS (Move RC-axis via push motion)

*1 439 RC Position Data Use Method Error when a command was executed

RC position-data use mode	XSEL	○ Can be used
	RC	× Cannot be used*1

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RPUS	RC-axis number	Position number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] The axis moves to the target position corresponding to the position number in operand 2, and then push the load over the in-position width specified by the position data.
The push force is set by the current-limiting value for push-motion operation among the position data.
The output turns ON when a push action is confirmed, and turns OFF if a miss is detected.

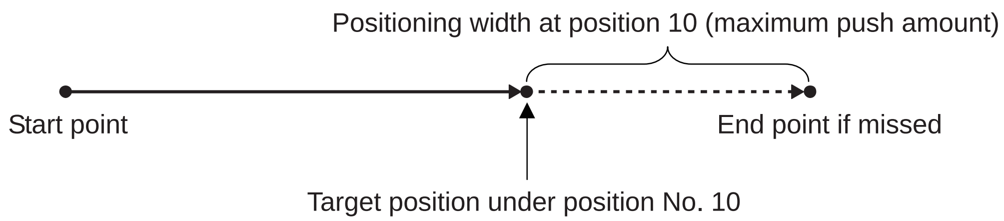
[Operand 1 setting type]

Operand 1	Specification of execution axis	Output specification
0 to 15	The axis corresponding to the specified RC-axis number moves via push motion.	The output turns ON when pushing of the command axis is confirmed.
-1	Each RC-axis specified by an RAXS command moves via push motion.	The output turns ON when pushing of all command axes is confirmed.
-2	Each RC-axis specified by an RAXS command moves via push motion.	The output turns ON when pushing of any one of all command axes is confirmed.

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

⚠ Caution:
- If a positive sign is appended to positioning width data, the load is pushed in the direction of increasing coordinates from the start point of the RPUS command toward the target position.
- If a negative sign is appended, the load is pushed in the direction of decreasing coordinates. (The operation is different from when a PUSH command is used.)

[Example] PRUS 3 10 Move RC-axis 3 to the position corresponding to position No. 10 and cause it to push the load.



● RSTP (Cancel RC-axis movement)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RSTP	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Decelerate each RC-axis specified by an RAXS command to a stop.
This command is valid with respect to all RC-axis control commands other than RSOF.

> Notice : Before executing this command, set an axis pattern using an RAXS command.
> If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example]

RAXS	0	11	Set an axis pattern consisting of axes 0 and 1.
RSTP			Decelerate the specified axes to a stop.

● RCST (Read RC-axis status)

RC position-data use mode	XSEL	○ Can be used
	RC	○ Can be used

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	RCST	Variable number	RC-axis number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	○	○	×	×	×

[Function] Read the RC-axis status into the variable number in operand 1.
Read the completed position number into variable n+1. (Refer to “Note 2”.)

Notice 1: The specific status varies between the XSEL position-data use mode and RC position-data use mode.

Variable number	Acquired data
n	RC-axis status
n+1	Completed position number

RC-axis status bit structure

Bit	XSEL position-data use mode		RC position-data use mode	
	Name	Explanation	Name	Explanation
27-31	–	Reserved	–	Reserved
26	ALMX	RC-axis alarm (Error detected by the XSEL) * When ALM turns ON, ALMX also turns ON. However, ALM may not turn ON even if ALMS turns ON depending on the error.	ALMX	RC-axis alarm (Error detected by the XSEL) * When ALM turns ON, ALMX also turns ON. However, ALM may not turn ON even if ALMS turns ON depending on the error.
25	USE	RC-axis in use	USE	RC-axis in use
24	LNK	RC-axis linked	LNK	RC-axis linked
16-23	–	Reserved	–	Reserved
15	RMDS	Operation Mode	RMDS	Operation Mode
14	ALML	Light Error Alarm	ALML	Light Error Alarm
13	ZON2	Zone 2	ZON2	Zone 2
12	ZON1	Zone 1	ZON1	Zone 1
11	–	Reserved	PZON	Position zone
10	–	Reserved	MODS	Teaching mode status
9	SFTY	Safety speed enabled	SFTY	Safety speed enabled
8	BALM	Battery voltage low	BALM	Battery voltage low
7	EMG	Emergency stop	EMG	Emergency stop
6	PSFL	Load not pushed	PSFL	Load not pushed
5	CRDY	Controller ready	CRDY	Controller ready
4	SON	Servo ON	SON	Servo ON
3	MOVE	Moving	MOVE	Moving
2	HEND	Home return complete	HEND	Home return complete
1	PEND	Positioning complete	PEND	Positioning complete
0	ALM	Operation-disabling alarm (Error detected by the RC-axis)	ALM	Operation-disabling alarm (Error detected by the RC-axis)

Notice 2: Completed position numbers are set only in the RC position-data use mode.
In the XSEL position-data use mode, this bit is always set to 0.

[Example] RCST 200 10 Acquire the status of RC-axis 10 into variable 200.

● XCRP (Clear input counter record for extension motion control board)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XCRP	Pulse input channel number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] This clears the specified pulse input channel counter to 0.

> ⚠ Caution
> The counter clear cannot be performed when the pulse I/O board axis is in synchronizing operation with the specified channel used as the master axis.

[Example 1] XCRP 0 It clears the counter for the pulse input channel No. 0.

● XGTP (Acquire the current value for extension motion control board pulse input counter)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XGTP	Pulse input channel number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the current value for the pulse input channel counter specified in Operand 1 into Variable 99.

[Example 1] XGTP 0 It acquires the pulse input channel No. 0 counter in Variable 99.

⚠ Caution
The pulse I/O board input channel is a signed 32-bit counter.

● XPGT (Read extension motion control board axis position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPGT	Axis number	Position number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data location [mm] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

[Example 1]	XPGT	2	3	It reads the position set in Position No. 3 of the 2nd axis into Variable 199.
[Example 2]	LET	1	2	Assign 2 to variable 1.
	LET	2	3	Assign 3 to variable 2.
	XPGT	*1	*2	It reads the position set in Position No. 3 (content of Variable 2) of the 2nd axis (content of Variable 1) into Variable 199.

⚠ Caution
If ineffective position data is specified in Operand 2, Variable 199 becomes non-operated and the output section is turned OFF.

● XPPT (Write extension motion control board axis position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPPT	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data location [mm] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

[Example 1]

LET	199	150	Assign 150 to variable 199.
XPPT	2	3	It writes Content 150 in Variable 199 in Position No. 3 of the 2nd axis.

[Example 2]

LET	199	150	Assign 150 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
XPPT	*1	*2	It writes Content 150 in Variable 199 in the position set in Position No. 3 (content of Variable 2) of the 2nd axis (content of Variable 1).

● XPCR (Erase extension motion control board axis position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPCR	Axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It erases the pulse I/O board axis position data specified in Operand 1 by using the two variables in a row from Variable No. n specified in Operand 2.
The erased data becomes a blank.

Variable No.	Description of setting
n	Start position number
n+1	End position number

[Example 1]

```
LET    200   10     Assign 10 to variable 200.
LET    201   20     Assign 20 to variable 201.
XPCR   1     200    It erases Positions No. 10 to 20 in the 1st axis.
```

● XPCP (Copy extension motion control board axis position data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPCP	Axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It copies the pulse I/O board axis position data specified in Operand 1 by using the two variables in a row from Variable No. n specified in Operand 2.

Variable No.	Description of setting
n	Position number to copy data to
n+1	Position number to copy data from

[Example 1]

LET	200	20	Assign 20 to variable 200.
LET	201	10	Assign 10 to variable 201.
XPCP	1	200	It copies Position No. 10 data in the 1st axis to Position No. 20.

● XPRD (Read extension motion control board axis current command position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPRD	Positio number	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the current order position of the pulse I/O board axis specified by XAXS Command into the position number specified in Operand 1.

⚠ Caution
- Make sure to set the axis pattern by XAXS Command before executing this command. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.
- he position acquired in this command is the current order position from the pulse I/O board. Make sure to perform a home-return operation before executing this command.

[Example 1]

XAXS	0	111	Set an pattern that uses axes 0,1 and 2.
XPRD	100		It reads the current order position of 0 to 2 axes into Position No. 100.

[Example 2]

XAXS	0	111	Set an pattern that uses axes 0,1 and 2.
LET	1	100	Assign 100 to variable 1.
XPRD	*1		It reads the current order position of 0 to 2 axes into the position in Content No. 100 of Variable 1.

● XPRQ (Read extension motion control board axis current command position (single-axis direct))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPRQ	Axis number	Variable number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the current order position of the pulse I/O board axis specified in Operand 1 into variable specified in Operand 2 (minimum effective digit number = 3). It enables a faster acquirement of the current order position than using XPRD Command.

> ⚠ Caution
> The position acquired in this command is the current order position from the pulse I/O board. Make sure to perform a home-return operation before executing this command.

[Example] XPRQ 2 100 It reads the current order position of the 2nd axis into Variable No. 100.

● XPVL (Write extension motion control board axis speed data)

Extension condition (LD,A,O,AB,OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPVL	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It writes the value in Variable 199 to the position data speed [mm/s] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

LET	199	100	Assign 100 to variable 199.
XPVL	2	3	It writes the speed 100mm/s in Variable 199 to Position No. 3 on the 2nd axis.

[Example 2]

LET	199	100	Assign 199 to variable 100.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
XPVL	*1	*2	It writes the speed 100mm/s in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1).

● XPAC (Write extension motion control board axis acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPAC	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It writes the value in Variable 199 to the position data acceleration [G] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

LET	199	0.3	Assign 0.3 to variable 199.
XPAC	2	3	It writes the acceleration 0.3G in Variable 199 to Position No. 3 on the 2nd axis.

[Example 2]

LET	199	0.3	Assign 0.3 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
XPAC	*1	*2	It writes the acceleration 0.3G in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1).

● XPDC (Write extension motion control board axis deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPDC	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It writes the value in Variable 199 to the position data deceleration [G] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

LET	199	0.3	Assign 0.3 to variable 199.
XPDC	2	3	It writes the deceleration 0.3G in Variable 199 to Position No. 3 on the 2nd axis.

[Example 2]

LET	199	0.3	Assign 0.3 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
XPDC	*1	*2	It writes the deceleration 0.3G in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1).

● XPIP (Write extension motion control board axis positioning complete width data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPIP	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It writes the value in Variable 199 to the position data positioning complete width [mm] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 (minimum effective digit number = 3).

[Example 1]

LET	199	0.2	Assign 0.2 to variable 199.
XPIP	2	3	It writes the positioning complete width 0.2mm in Variable 199 to Position No. 3 on the 2nd axis.

[Example 2]

LET	199	0.2	Assign 0.2 to variable 199.
LET	1	2	Assign 2 to variable 1.
LET	2	3	Assign 3 to variable 2.
XPIP	*1	*2	It writes the positioning complete width 0.2mm in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1).

[27] Electronic Cam Control System

● XGVL (Read extension motion control board axis speed data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XGVL	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data speed [mm/s] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGVL 2 3 It reads the speed in Position No. 3 on the 2nd axis into Variable 199.

● XGAC (Read extension motion control board axis acceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XGAC	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data acceleration [G] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGAC 2 3 It reads the acceleration in Position No. 3 on the 2nd axis into Variable 199.

● XGDC (Read extension motion control board axis deceleration data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XGDC	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data deceleration [G] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGDC 2 3 It reads the deceleration in Position No. 3 on the 2nd axis into Variable 199.

● XGIP (Read extension motion control board axis positioning complete width data)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XGIP	Axis number	Position number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the position data positioning complete width [mm] specified in Operand 2 on the pulse I/O board axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

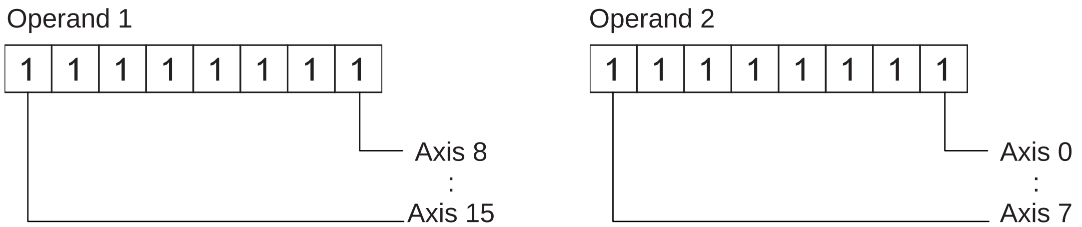
[Example] XGIP 2 3 It reads the positioning complete width in Position No. 3 on the 2nd axis into Variable 199.

● XAXS (Extension motion control board axis pattern setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XAXS	Axis pattern, upper	Axis pattern, lower	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It performs a setting for the axis patterns of axes 8 to 15 on the pulse I/O board axis in Operand 1 and the axis patterns of axes 0 to 7 in Operand 2.



After the program execution is started, make sure to set the axis patterns using this command before the following commands are executed.
If the axis pattern setting command XAXS is not conducted, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

XPRD : Read extension motion control board axis current command position
XSON : Servo ON of extension motion control board axis
XSOF : Servo OFF of extension motion control board axis
XHOM : Home return of extension motion control board axis
XMVP : Move extension motion control board axis to indicated position
XMPI : Perform extension motion control board axis position relative movement
XMVL : Move extension motion control board axis for position indicated interpolation
XMLI : Move extension motion control board axis for position relative interpolation
XSTP : Cancel operation of extension motion control board axis

[Example]

XAXS	1010101	10101010	Set an axis pattern consisting of axes 1, 3, 5, 7, 8, 10, 12 and 14.
XSON			It turns the servo ON for axes 1, 3, 5, 7, 8, 10, 12 and 14.
XMVP	20		It moves the axes 1, 3, 5, 7, 8, 10, 12 and 14 to Position 20.

● XSON (Extension motion control board axis servo ON)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XSON	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It turns the servo ON for the pulse I/O board axis specified by XAXS Command.

> ⚠ Caution
> Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

[Example]	XAXS	01	100	Set an axis pattern that uses axes 2 and 3.
	XSON			Turn ON the servos of the specified axes.

● XSOF (Extension motion control board axis servo OFF)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XSOF	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It turns the servo OFF for the pulse I/O board axis specified by XAXS Command.

> ⚠ Caution
> Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 "Extension motion control board axis pattern not set error" would occur.

[Example]	XAXS	0	1100	Set an axis pattern that uses axes 2 and 3.
	XSOF			Turn OFF the servos of the specified axes.

● XHOM (Extension motion control board axis home return)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XHOM	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It turns the home return for the pulse I/O board axis specified by XAXS Command.
The servo of the axis to be returned home turns ON automatically.

⚠ Caution
- Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.
- The servo would turn OFF if the operation is either paused or stopped to cancel during the home-return operation of the pulse I/O board axis. When resuming the operation after a pause, confirm the servo is ON and then make sure to start with a home-return operation.

[Example]	XAXS	0	1100	Set an axis pattern that uses axes 2 and 3.
	XHOM			Home Return the servos of the specified axes.

● XMVP (Move extension motion control board axis to indicated position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMVP	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified by XAXS Command by PTP operation to the position number specified in Operand 1.

⚠ Caution
- Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.
- It is able to specify another axis that is connected to a different pulse I/O board at the same time. However, since the system is controlled by each board, the operation cannot be synchronized.

[Example 1]	XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
	XMVP	10		Move the specified axes to the positions corresponding to position No. 10.

[Example 2]	XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
	LET	1	10	Assign 10 to variable 1.
	XMVP	*1		Move the specified axes to the positions corresponding to position No. 10 according to the content of variable 1, or 10.

● XMPI (Perform extension motion control board axis position relative movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMPI	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified by XAXS Command by PTP operation with the position number in Operand 1 taken as the amount of movement.

⚠ Caution
- Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.
- It is able to specify another axis that is connected to a different pulse I/O board at the same time. However, since the system is controlled by each board, the operation cannot be synchronized.

[Example 1]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XMPI	10		Move by the travel corresponding to position No. 10.

[Example 2]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
LET	1	10	Assign 10 to variable 1.
XMPI	*1		Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10.

● XMVL (Move extension motion control board axis for position indicated interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMVL	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified by XAXS Command by direct interpolation movement to the position number specified in Operand 1.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

(Note 2) If another axis connected to a different pulse I/O board is specified, Error No. C30 “Axis pattern error” would occur.

(Note 3) It is necessary to specify the speed, acceleration and deceleration values by VEL, VLMX, ACC, and DCL Commands before executing this command. If not specified, an error would occur.

[Example 1]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XMVL	10		Move the specified axes to the positions corresponding to position No. 10.

[Example 2]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
LET	1	10	Assign 10 to variable 1.
XMVL	*1		Move the specified axes to the positions corresponding to position No. 10 according to the content of variable 1, or 10.

● XMLI (Move extension motion control board axis for position relative interpolation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMLI	Position number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified by XAXS Command by direct interpolation movement with the position number in Operand 1 taken as the amount of movement.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

(Note 2) If another axis connected to a different pulse I/O board is specified, Error No. C30 “Axis pattern error” would occur.

(Note 3) It is necessary to specify the speed, acceleration and deceleration values by VEL, VLMX, ACC, and DCL Commands before executing this command. If not specified, an error would occur.

[Example 1]	XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
	XMLI	10		Move by the travel corresponding to position No. 10.
[Example 2]	XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
	LET	1	10	Assign 10 to variable 1.
	XMLI	*1		Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10.

● XMVD (Move extension motion control board axis to directly indicated absolute position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMVD	Axis number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified in Operand 1 by absolute position movement to the values specified in the five variables in a row from Variable No. n in Operand 2.

Variable No.	Description of setting	Effective Digits
n	Target position [mm]	3 digits minimum
n+1	Speed [mm/s]	2 digits minimum
n+2	Acceleration [G]	2 digits minimum
n+3	Deceleration [G]	2 digits minimum
n+4	Positioning complete width [mm]	3 digits minimum

(Note 1) VLMX Command is invalid to this command.

[Example]

```
LET   300  100   Set the target position to 100mm.
LET   301  200   Set the speed to 200mm/s.
LET   302  0.3   Set the acceleration/deceleration to 0.3G.
LET   303  0.3   Set the deceleration/deceleration to 0.3G.
LET   304  0.1   Set the in-position width to 0.1mm.
XMVD  1    300   Move RC-axis 1 absolutely to the specified position.
```

● XMDI (Move extension motion control board axis to directly indicated relative position)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XMDI	Axis number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board axis specified in Operand 1 by relative position movement to the values specified in the five variables in a row from Variable No. n in Operand 2.

Variable No.	Description of setting	Effective Digits
n	Travel [mm]	3 digits minimum
n+1	Speed [mm/s]	2 digits minimum
n+2	Acceleration [G]	2 digits minimum
n+3	Deceleration [G]	2 digits minimum
n+4	Positioning complete width [mm]	3 digits minimum

(Note 1) VLMX Command is invalid to this command.

[Example]

```
LET   300  50    Set the travel to 50mm.
LET   301  200   Set the speed to 200mm/s.
LET   302  0.3   Set the acceleration/deceleration to 0.3G.
LET   303  0.3   Set the deceleration/deceleration to 0.3G.
LET   304  0.1   Set the in-position width to 0.1mm.
XMDI  1    300   Move RC-axis 1 absolutely to the specified position.
```

● XJ□□ (Perform extension motion control board axis jog operation)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XJ□□	Input, output, flag number	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It moves the pulse I/O board control shaft specified by XAXS Command in back and forth while the flag on the input port or output port specified in Operand 1 is turning ON and OFF.

XJFN ············It moves forward when the specified port is ON.
XJFF·············It moves forward when the specified port is OFF.
XJBN ············It moves backward when the specified port is ON.
XJBF ············It moves backward when the specified port is ON.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

(Note 2) It is effective also to the axis that the home-return operation is incomplete. However, the upper limit for the speed is that set in pulse I/O board command parameter No. 4 “Maximum JOG speed at home return incomplete”. In such a condition, exercise precaution not to crash into the work or stroke end since the coordinate values become meaningless.

(Note 3) This command is valid on Main CPU Application Section Ver.1.02 or later. And a PC software Ver.7.6.5.0 or later which is applicable for this command is also required. (This is for XSEL-P/Q. It is available from the first for XSEL-R/S.)

[Example 1]

VEL	100		Set the speed to 100mm/s.
XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XJBF	10		Move axis 5 backward while input 10 is OFF.

[Example 2]

VEL	100		Set the speed to 100mm/s.
LET	5	20	Assign 20 to variable 5.
XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XJFN	*5		Move axis 0 and 1 forward while the content of variable 5 (input 20), is ON.

● XPED (Waiting for extension motion control board axis to finish positioning operation of axis used by self-program)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XPED	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It waits for the positioning operation of the pulse I/O board axis used in the program its own. By this command, it is possible to wait for the completion of the positioning operation (XMVP, XMPI, XMVL, XMLI, XMVD and XMDI) when the positioning complete width is valid. The output becomes ON when the operation is completed in normal condition. The command would not react after an execution of any operation command other than positioning operation. (Output section is OFF.)
For an operation that the positioning complete band is valid, recovery from the operation command is performed once the actuator reaches in front of the positioning complete band at the current position (or current position command when the pulse input and output boards are mounted). (Output section is OFF.) It is possible to confirm the positioning is complete by executing this command after the command recovery. Also, the driven axis is usually occupied by the executed program after the operation command recovery. By executing this command, the axis gets released, thus the axis becomes available for a use by other programs.

[Example]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XMVP	10		Move the specified axes to the positions corresponding to position No. 10. For an operation that the positioning complete band is valid, recovery from the operation command is performed once the actuator reaches in front of the positioning complete band at the current position (or current position command when the pulse input and output boards are mounted).
BTON	308		It turns ON Output Port No. 308.
XPED			It waits for the positioning operation axis No. 0 and 1 of the program itself to complete.

● XSTP (Cancel operation of extension motion control board axis)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XSTP	Prohibited	Prohibited	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It decelerates and stops the expansion pulse I/O board axis specified by XAXS Command. It is valid for the pulse I/O board axis actuator control command other than XSOF command.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

[Example]

XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XSTP			Decelerate the specified axes to a stop

● XWIP (Waiting for extension motion control board axis positioning complete signal to be turned ON)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XWIP	Prohibited	Prohibited	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It waits till the positioning complete signal of the pulse I/O board control shaft specified by XAXS Command turns ON. This enables to wait for the completion of the positioning operation of the slave shaft (= slave shaft positioning complete signal ON) while in synchronizing process by executing this command to the synchronizing slave shaft after the synchronizing master shaft operation command is complete (*). The positioning complete signal for the slave shaft turns ON when position deviation ≤ positioning complete width. The status would not become waiting unless the pulse order is output from the master shaft side to the slave side.

* The pulse order of the slave shaft is completed by the completion of the master shaft operation command.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” would occur.

(Note 2) If the positioning complete signal does not turn ON even after the time set in the pulse I/O board output channel parameter No. 33 “Positioning complete confirmation time” of the specified axis is passed, Error No. 454 “pulse I/O board axis positioning complete timeout error” would occur.
Check if the positioning complete signal cable is broken.

(Note 3) This command is valid on Main CPU Application Section Ver.1.02 or later. And a PC software Ver.7.6.5.0 or later which is applicable for this command is also required. (This is for XSEL-P/Q. It is available from the first for XSEL-R/S.)

[Example]

XCAS	0	10	It starts to synchronize the electronic cam on axis 0. (*1)
XCAS	1	20	It starts to synchronize the electronic cam on axis 1. (*2)
MOVP	5		It moves the 1st axis of the main CPU control axes to Position No. 5.
MOVP	6		It moves the 1st axis of the main CPU control axes to Position No. 6.
XAXS	0	11	Set an axis pattern that uses axes 0 and 1.
XWIP			It waits till the positioning complete signal of axes 0 and 1 to turn ON. (Axes 0 and 1 continue the synchronizing process.)
MOVP	7		It moves the 1st axis of the main CPU control axe to Position No. 7.

* It is assumed the setting that the 1st axis of the main CPU control axes is set to Variables No. 10 to 19 as the electronic cam synchronizing process is established.

* It is assumed the setting that the 1st axis of the main CPU control axes is set to Variables No. 20 to 29 as the electronic cam synchronizing process is established.

● XCAS (Start synchronizing extension motion control board axis electronic cam (indicating main axis))

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XCAS	Slave shaft number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It starts the synchronizing process with the axis specified in Operand 1 as the slave shaft following the electronic cam table. The synchronizing electronic cam operation settings such as the master shaft on the electronic cam table are to be specified in ten variables in a row from Variable No. n in Operand 2. The output section turns ON when the synchronizing process is started.

■Operand 2 : Synchronizing Electronic Cam Operation Settings

<table>
<tr><th>Variable No.</th><th>Data name</th><th colspan="2">Description</th></tr>
<tr><td>n</td><td>Synchronizing type</td><td colspan="2">See below</td></tr>
<tr><td>n+1</td><td>Master shaft type</td><td colspan="2">0 : Main CPU control shaft
1 : Pulse I/O board control shaft
2 : Pulse input channel</td></tr>
<tr><td rowspan="4">n+2</td><td rowspan="4">Master shaft number</td><td>Master shaft type</td><td></td></tr>
<tr><td>0</td><td>1 to 6 axis (XSEL-P/Q), 1to 8 axis (XSEL-R/S)</td></tr>
<tr><td>1</td><td>0 to 15 axis</td></tr>
<tr><td>2</td><td>0 to 3 channel</td></tr>
<tr><td>n+3</td><td>Electronic cam table number</td><td colspan="2">*0 to</td></tr>
<tr><td>n+4</td><td>Stroke type</td><td colspan="2">0 : Master shaft stroke length indication
1 : Master shaft stroke end position indication</td></tr>
</table>

<table>
<tr><th>Variable No.</th><th>Data name</th><th colspan="3">Description</th></tr>
<tr><td rowspan="8">n+5</td><td rowspan="8">Master shaft stroke length / stroke end position (Storage position number)</td><td>Stroke type</td><td>Master shaft type</td><td></td></tr>
<tr><td rowspan="3">0</td><td>0</td><td>Master shaft stroke length storage position number
* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke length storage position number
* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td>2</td><td>Master shaft stroke length (pulse unit)</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>Master shaft stroke end position storage position number
* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke end position storage position number
* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td>2</td><td>Master shaft stroke end position indication (pulse unit)</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>n+6</td><td>Slave stroke length storage position number</td><td colspan="3">Indicate the pulse I/O board control slave shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td rowspan="4">n+7</td><td rowspan="4">Master shaft synchronizing start position (Storage position number)
* Effective only when "Master shaft reaches specified synchronizing start position" is selected for synchronizing type</td><td colspan="2">Master shaft type</td><td></td></tr>
<tr><td colspan="2">0</td><td>Master shaft synchronizing start position storage position number
* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">1</td><td>Master shaft synchronizing start position storage position number
* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">2</td><td>Master shaft synchronizing start position (pulse unit)</td></tr>
<tr><td>n+8</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
<tr><td>n+9</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
</table>

■Synchronizing Type (Variable No.n)

Set value	Description	
	Synchronizing start type	Synchronizing process repeat type
0	Immediately	Operate for 1 cycle
1	Immediately	Repeated operation
2	Master shaft reaches specified synchronizing start position	Operate for 1 cycle
3	Master shaft reaches specified synchronizing start position	Repeated operation

The synchronizing process continues until:
- XSYE Command (to cancel synchronizing process) is executed,
- an operation cancel is executed to the slave shaft (XSTP Command, CANC Command),
- Synchronizing Process Repeat Type is set to 1 cycle and the master shaft reaches to the stroke end, or the slave shaft operation program that XCAS Command is executed is over.

- If the master axis is a main CPU control axis or pulse I/O board control axis, set the master stroke length/stroke end position and master shaft synchronizing start position to the master shaft position data. If the master shaft is the pulse input channel, set it directly to the variable for operation settings. Set the slave shaft stroke length to the slave shaft position data.
- If Stroke Type = Indicate master stroke end position, the master stroke length (1 cycle) on the electronic cam table is [Master shaft stroke end position-Synchronizing start master shaft position]. The relation between the master shaft position and the electronic cam table phase is that the synchronizing start master shaft position is the phase 0, and the direction from the synchronizing start master shaft position to the master shaft stroke end position is the phase positive direction.
- If Stroke Type = Indicate master shaft stroke length, the relation between the master shaft position and the electronic cam table is that the synchronizing start master shaft position is the phase 0, and if the stroke length is a positive value, the positive direction on the master axis coordinate is the phase positive direction and, if the stroke length is a negative value, the positive direction on the master shaft coordinate is the phase negative direction.

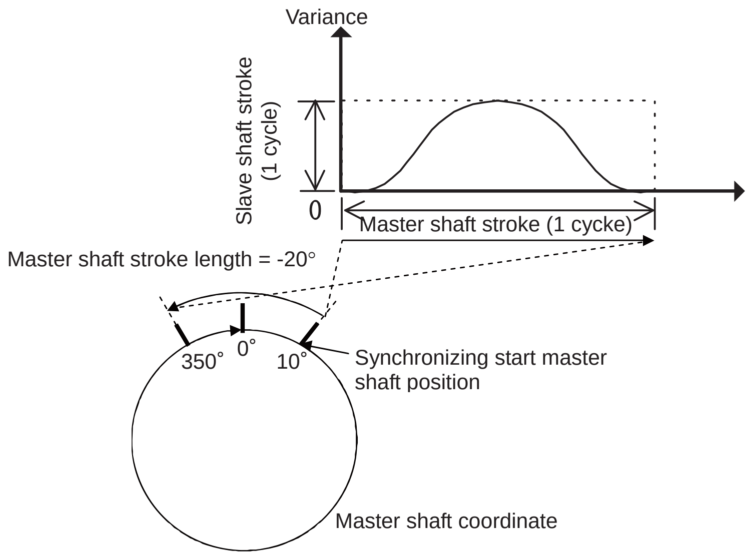
⚠ Caution

- If there is a mistake in the synchronizing electronic cam operation settings specified in the variables in Operand 2, Error No. 455 “Extension motion control board synchronizing electronic cam operation setting error” would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (in hexadecimal numbers).
  - Variable with an error occurred: Master axis number (Stored variable)
    - The specified master shaft number is inappropriate or invalid
    - The specified master shaft is a synchronized slave shaft or ZR unit shaft (if the master shaft is a main CPU control shaft)
    - The specified master shaft is the shaft specified as the slave (if the master shaft is a pulse I/O board control shaft)
  - The specified master shaft is on a different pulse I/O board or channel from that the slave shaft is on (if the master shaft is a pulse I/O board control shaft or a pulse input channel)
  - Variable with an error occurred: stroke length/stroke end position storage position number
    - The specified position number is inappropriate or position data is invalid
- If the master shaft type for the synchronizing electronic cam operation settings is the main CPU control shaft, BASE Command settings would be effective to the master shaft number. Also, GRP Command settings are invalid to the position data to store the stroke length and stroke end position.
- If the robot is equipped with multiple pulse I/O board, the electronic cam table which is stored to the board that the slave shaft is connected to would be used.
- If the slave shaft starts to move for a synchronizing process during the master shaft is in move, the speed and acceleration/deceleration may get too high and may cause an error. Lower the settings for the speed and acceleration of the master shaft during the slave synchronizing movement starts.
- During a movement following the electronic cam table, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the speed, acceleration/deceleration and electronic cam table so they are set to the allowable speed and acceleration/deceleration for the shaft.

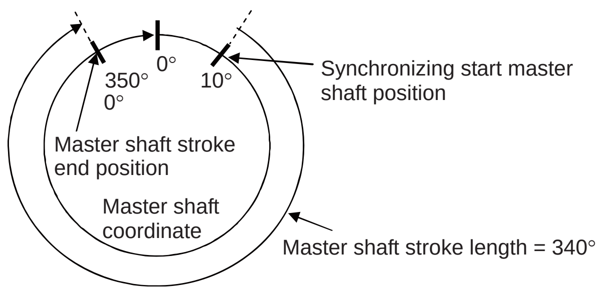
⚠ Caution

- Once the slave shaft starts synchronizing, it will be occupied by the program until XSYE Command (to cancel synchronizing) is executed or the slave shaft operation program that XCAS Command is executed is finished.
  Therefore Error No. 449 “Extension motion control board shaft duplication use error” would occur if the shaft is used by another program. Also, even in the same program, Error No. 449 would occur if an operation command is executed to the shaft that the synchronizing process is already completed. Execute XSYE Command if a next operation is required after the synchronizing process is finished.
  For XSEL-R/S, the number of the main CPU control axes that can be the master axis for the synchronizing operation of such as the electronic cam is six axes at the maximum.
  The axes that can be the master axis can be selected in I/O Parameter No. 529 “Extension Motion Control Board Synchronizing Main CPU Control Master Select Axis Pattern” (dynamic change not available). Please refer to the parameter list in XSEL-R/S Instruction Manual for the details of the parameter.
  When indicating an axis that is not selected as the main axis select in I/O Parameter No. 529 as the main axis in XCAS Command, Error No. 455 “Extension Motion Control Board Synchronizing Electronic Cam Operation Setting Error” will occur.
- If the master shaft is the main CPU control axis with the rotation axis close control, set the master shaft stroke type in the synchronizing electronic cam operation settings to the stroke length setting. If the setting is specified to the stroke end position, it may not perform a synchronizing process that is expected.

[Example] For the synchronizing process in range of master shaft position = 10° to 350°



If the stroke end position = 350° assuming the master shaft stroke type = stroke end position, the figure will be as shown below:



Program Example

LET	200	1	It sets the synchronizing type = 1 (Immediate start, Repeat operation) to Variable No. 200.
LET	201	0	It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 201.
LET	202	1	It sets the master shaft No. = 1 to Variable No. 202.
LET	203	0	It sets the electronic cam table No. = 1 to Variable No. 203.
LET	204	1	It sets the stroke type = 1 (Master stroke end position specification) to Variable No. 204.
LET	205	2	It sets the master shaft stroke end position storage position No. = 2 to Variable No. 205.
LET	206	0	It sets the slave stroke length storage position No. = 0 to Variable No. 206.
LET	207	0	It sets 0 to Variable No. 207 (Unused data)
LET	208	0	It sets 0 to Variable No. 208 (Reserved area)
LET	209	0	It sets 0 to Variable No. 209 (Reserved area)
XAXS	0	1	It specifies the pulse I/O board 0^{th} axis.
XSON			It turns the pulse I/O board 0^{th} axis servo ON.
XHOM			It returns the pulse I/O board 0^{th} axis to home return.
XCAS	0	200	It starts the synchronizing electronic cam operation for the pulse I/O board 0^{th} axis with the synchronizing electronic cam operation settings specified in Variables No. 200 to 209.
TAG	1		
MOVP	2		It moves the XSEL control master axis to Position No. 2.
MOVP	1		It moves the XSEL control master axis to Position No. 1
GOTO	1		

● XCTM (Extension motion control board Single Electronic Cam (Time Specification) Movement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XCTM	Slave shaft number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It performs a movement following the electronic cam table having the time axis of the shaft specified in Operand 1 as the master shaft. Single electronic cam operation settings such as the movement time and the electronic cam table number, etc., are to be specified in five variables in a row from Variable No. n in Operand 2. The output turns OFF at the same time the command is started and turns ON when the movement is complete.

■Operand 2 : Single Electronic Cam Operation Settings

Variable No.	Data	Description
n	Electronic cam table number	
n+1	Moving time	Unit in 0.001sec
n+2	Slave stroke length storage position number	* Indicate the pulse I/O board control slave shaft position number (from 0 to Max. position No.)
n+3	Reserved	Set 0
n+4	Reserved	Set 0

Set the slave shaft stroke length to the slave shaft position data.
When the slave shaft stroke length is a positive value, the movement is made to the coordinate positive direction. When the slave shaft stroke length is a negative value, the movement is made to the coordinate positive direction.

> ⚠ Caution
>
> - If there is a mistake in the single electronic cam operation settings specified in the variables in Operand 2, Error No. 456 “Extension motion control board single electronic cam operation settings error” would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (in hexadecimal numbers).
>
>   - Variable with an error occurred: Slave stroke length storage position number (Stored variable)
>   - Specified position number is inappropriate or the position data is inefficient.
>
> - If the robot is equipped with multiple pulse I/O board, the electronic cam table which is stored to the board that the slave shaft is connected to would be used.
>
> - During a movement following the electronic cam table, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the moving time and electronic cam table so they are set to the allowable speed and acceleration/deceleration for the shaft.

Program Example

LET	200	0	It sets the electronic cam table No. = 0 to Variable No. 200.
LET	201	1000	It sets the movement time 1000ms to Variable 201.
LET	202	0	It sets the slave stroke length storage position No. = 0 to Variable No. 202.
LET	203	0	It sets 0 to Variable No. 203 (Reserved area)
LET	204	0	It sets 0 to Variable No. 204 (Reserved area)
XAXS	0	1	It specifies the pulse I/O board 0th axis.
XSON			It turns the pulse I/O board 0th axis servo ON.
XHOM			It returns the pulse I/O board 0th axis to home position.
XCTM	0	200	It starts the single electronic cam operation for the pulse I/O board 0th axis with the single electronic cam operation settings specified in Variables No. 200 to 204.

● XSFS (Extension Motion Montrol Board Electronic Shaft Synchronizing Start)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XSFS	Slave shaft number	Variable number	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It starts the synchronizing operation of the shaft specified in Operand 1 as the slave shaft following the master shaft. The electronic shaft operation settings such as the master shaft are to be specified in the four variables in a row from Variable No. n in Operand 2. The output section turns ON with the synchronizing start.

■Operand 2 : Single Electronic Shaft Operation Settings

Variable number	Data	Description
n	Master shaft type	0 : Main CPU control shaft 1 : Pulse I/O board control shaft 2 : Pulse input channel
n+1	Master shaft number	* Axes 1 to 6 (XSEL-P/Q), 1 to 8 axis (XSEL-R/S) when main CPU control shaft, axes 0 to 15 when pulse I/O board control shaft, channels 0 to 3 when pulse input channel
n+2	Gear ratio numerator	-10000 to -1, 1 to 10000
n+3	Gear ratio denominator	1 to 10000

The synchronizing process continues until:
- XSYE Command (to cancel synchronizing process) is executed,
- an operation cancel is executed to the slave shaft (XSTP Command, CANC Command),
- Synchronizing Process Repeat Type is set to 1 cycle and the master shaft reaches to the stroke end, or the slave shaft operation program that XSFS Command is executed is over.

If the gear ratio is set to a negative value, the slave shaft operates in a reverse direction against the master shaft.

⚠ Caution

If there is a mistake in the single electronic cam operation settings specified in the variables in Operand 2, Error No. 456 “Extension motion control board single electronic cam operation settings error” would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (in hexadecimal numbers).

- Variable with an error occurred : Master shaft number
  - The specified master shaft number is inappropriate or invalid
  - The specified master shaft is a synchronized slave shaft or ZR unit shaft (if the master shaft is a XSEL control shaft)
  - The specified master shaft is the shaft specified as the slave (if the master shaft is a pulse I/O board control shaft)
  - The specified master shaft is on a different pulse I/O board or channel from that the slave shaft is on (if the master shaft is a pulse I/O board control shaft or a pulse input channel)

⚠ Caution

- If the master shaft type in the electronic shaft operation settings is the main CPU control shaft, BASE Command would be effective to the master shaft number.
- If the master shaft is the pulse input channel, the slave shaft operates with a condition taking 1 input pulse from the pulse input channel as 0.001mm.
- If the slave shaft starts to move for a synchronizing process during the master shaft is in move, the speed and acceleration/deceleration may get too high and may cause an error. Lower the settings for the speed and acceleration of the master shaft during the slave synchronizing movement starts.
- During a master shaft, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the speed, acceleration/deceleration and gear ratio so they are set to the allowable speed and acceleration/deceleration for the shaft.
- Once the slave shaft starts synchronizing, it will be occupied by the program until XSYE Command (to cancel synchronizing) is executed or the slave shaft operation program that XSFS Command is executed is finished.
  Therefore Error No. 449 “Extension motion control board shaft duplication use error” would occur if the shaft is used by another program. Also, even in the same program, Error No. 449 would occur if an operation command is executed to the shaft that the synchronizing process is already completed. Execute XSYE Command if a next operation is required after the synchronizing process is finished.
  For XSEL-R/S, the number of the main CPU control axes that can be the master axis for the synchronizing operation of such as the electronic cam is six axes at the maximum.
  The axes that can be the master axis can be selected in I/O Parameter No. 529 “Extension Motion Control Board Synchronizing Main CPU Control Master Select Axis Pattern” (dynamic change not available). Please refer to the parameter list in XSEL-R/S Instruction Manual for the details of the parameter.
  When indicating an axis that is not selected as the main axis select in I/O Parameter No. 529 as the main axis in XCAS Command, Error No. 457 “Extension Motion Control Board Electronic Cam Operation Setting Error” will occur.

[Example]	LET	200	0	It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 200.
	LET	201	1	It sets the master shaft No. = 1 to Variable No. 201.
	LET	202	1	It sets the gear ratio numerator = 1 to Variable No. 202.
	LET	203	50	It sets the gear ratio denominator = 1 to Variable No. 203.
	XSFS	0	200	It starts electronic shaft synchronizing process of the pulse I/O board 0th shaft with the electronic shaft operation settings specified in Variables No. 200 to 203.
	TAG	1		
	MOVP	2		It moves the main CPU control master axis to Position No. 2.
	MOVP	1		It moves the main CPU control master axis to Position No. 1.
	GOTO	1		

● XSYE (Extension motion control board synchronizing process complete)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XSYE	Slave shaft number	(Complete type)	PE

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It finishes the synchronizing process of the slave shaft specified in Operand 1. This command is effective to the slave shaft that is in the synchronizing process with the synchronizing electronic cam (master shaft specified) started by the same program or that in electronic shaft synchronizing process (XSFS Command). If another shaft is specified, Error No. 444 “Extension motion control board axis number error” would occur.

It is able to specify the complete type in Operand 2. If 0 is select or no selection is done, the synchronizing process would be cancelled. If 1 is selected, it waits for the synchronizing process to finish. It is applicable in a case to wait for the slave shaft to finish its synchronizing process with “Synchronizing type = operate for 1 cycle” in XCAS Command or to wait till the synchronizing process to be cancelled by XSTP Command from another program.

The output section turns ON when 1 is selected for the complete type and the slave shaft finishes 1 cycle of operation with the synchronizing type = “Operate for 1 cycle”.

■Operand 2 : Complete Type

= 0 or no selection: Synchronizing process to be cancelled

= 1: Wait for synchronizing process to finish

⚠ Caution

- In the case “Synchronizing process to be cancelled” is chosen for the complete type, the command would only finishes the synchronizing process and would not confirm the slave shaft positioning completion (Slave Driver positioning complete signal). If waiting for the positioning to complete is desired, wait till the positioning complete signal input port turns ON directly by WTON Command after XSYE Command execution.
- By executing this command the occupation of the shaft is released. Thus, the slave shaft being in the synchronizing process can be used for other programs.

Program Example

LET	200	1	It sets the synchronizing type = 1 (Immediate start, operate for 1 cycle) to Variable No. 200.
LET	201	0	It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 201.
LET	202	1	It sets the master shaft No. = 1 to Variable No. 202.
LET	203	0	It sets the electronic cam table No. = 1 to Variable No. 203.
LET	204	1	It sets the stroke type = 1 (Master stroke end position specification) to Variable No. 24.
LET	205	2	It sets the master shaft stroke end position storage position No. = 2 to Variable No. 205.
LET	206	0	It sets the slave stroke length storage position No. = 0 to Variable No. 206.
LET	207	0	It sets 0 to Variable No. 207 (Unused data)
LET	208	0	It sets 0 to Variable No. 208 (Reserved area)
LET	209	0	It sets 0 to Variable No. 209 (Reserved area)
XCAS	0	200	It starts the synchronizing electronic cam operation for the pulse I/O board 0^{th} axis with the synchronizing electronic cam operation settings specified in Variables No. 200 to 209.
XSYE	0	1	It waits till the pulse I/O board 0^{th} shaft to operate 1 cycle for the synchronizing electronic cam operation. (Assuming the master shaft is operated by another program)
XAXS	0	1	It set the pulse I/O board 0th shaft to the axis pattern.
XMVP	10		It moves the specified axis to Position No. 10.

● XAST (Acquire extension motion control board axis status)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Output operation type (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	XAST	Variable number	Axis number	CP

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

[Function] It reads the axis status specified in Operand 2 into the variable in Operand 1.

Bit Construction of Axis Status

Bit	Information
27-31	Reserved
26	Axis Alarm (error detected by XSEL) * “Axis Alarm” includes “Continuous Impossible Alarm Occurrence (error detected in slave controller)” as well as the errors related to the extension motion control board of XSEL.
25	Axis in use
24	Reserved
14-23	Reserved
13	Reserved
12	Reserved
11	Reserved
10	Reserved
9	Safety speed valid status (safety speed is valid for XSEL)
8	Reserved
7	Reserved
6	Reserved
5	Reserved
4	Servo ON status.
3	Reserved
2	Home return completion * The bit rises when the home-return operation of XSEL controller is finished and the home-return complete status on the slave driver turns on.
1	Point position completion * The bit rises when an operation command of XSEL controller is finished and the positioning complete status on the slave driver turns on.
0	Continuity disabled alarm is generated (an error the slave driver generated)

[Example] XAST 200 10 Variable

[28] Conveyor Tracking Related Commands

● TRMD (Tracking mode setting)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Outpu (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TRMD	0 (Mode OFF) or 1 (Mode ON)	Operand 1 = 0 Prohibited Operand 1 = 1 (TRAC Command timeout time)	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	×	×	○	○	×	×	×

[Function] Set the Tracking Mode ON/OFF in Operand 1.
Only when Operand 1 = 1 (Tracking Mode ON), TRAC Command (explained later) timeout time (timeout time until the datum point in the working range exceeds the “minimum work position available for the tracking operation start” after TRAC Command is executed) can be selected in Operand 2. The settable range for the timeout time is settable from 0.00 to 99.00sec. When a selection of no timeout time setting (Operand 2 = not set) is made, TRAC Command defines there is no timeout setting and waits with no time limitation.
Work detection process becomes valid only when Tracking Mode is ON.

• Return Code in TRMD Command (Variable 99 (Local Space))
* When Operand 1 = 0 (Tracking Mode OFF), the return code cannot be returned. (Variable 99 not operated)
* When the return code is the numbers except for 0, Tracking Mode is turned OFF.

0 : Tracking Mode ON (In normal condition)
1 : Vision System initializing incomplete
2 : Ethernet connection incomplete

⚠ Caution : TRMD and TRAC Commands are available only when they are in the same program (task).
Dedicated application software is required when using the conveyor tracking function. Please contact us for the details.

● TRAC (Tracking operation setting & datum point position information obtainment in work)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Outpu (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	TRAC	0 (Operation OFF) or 1 (Operation (Standby) ON)	Operand 1 = 0 Prohibited Operand 1 = 1 Position number to save the work position information	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	×	×	○	○	×	×	×

[Function] Set the Tracking Mode ON/OFF in Operand 1.
It is necessary to specify the work position information storage position number in Operand 2 if Operand 1 = 1 (Tracking operation on setting) is specified.
After the tracking operation command is executed, the datum position information for the identified top work is stored to the position number specified in Operand 2 if the datum point within the work detected and confirmed during the work detection process exceeds “Minimum tracking operation start work position” (if it is already exceeded, at the same time TRAC Command is executed). If the work position information is acquired, move the actuator to the position above the work immediately with MOVL Command with a care to Z-axis (height).

Datum Point Position Information in Work Saved in Position Data
• X, Y, (rotation) R-axis

If Tracking Operation ON Command is executed while already in the conveyor tracking operation, the tracking operation will continue and only the datum point position information obtainment in the next work is performed.
When the tracking operation OFF command is executed, the tracking operation is cancelled and it decelerates and stops tracking. If the tracking operation is cancelled by the tracking operation OFF command, etc., the data such as the acquired work datum position information would be invalid (meaningless).

- Return Code in TRAC Command (Variable 99 (Local Space))
  * When Operand 1 = 0 (Tracking Operation OFF), the return code cannot be returned. (Variable 99 not operated)
  0: Tracking operation start & datum point position information obtaining succeeded

1. Datum point in work position information obtaining timeout
   Timeout value should be indicated in Operand 2 of TRMD Command as described previously.
2. Datum point in work position information obtaining timer cancel (Timer cancel by TIMC Command)
3. Reached the maximum work position for tracking operation start (Work reached a position that cannot be tracked)
   Even thought the datum point position information in work can be obtained, the data is no more meaningful, thus ensure not to do the positioning with that position information. The work attribute (it is for the future expansion, currently fixed value = no attribute identification) isaved to the local variable indicated in "All-Axes Parameter No. 93 Tracking Work Attribute Saved Local Variable Number".
4. Tracking operation stop
   When the work reaches the position to finish the tracking operation, tracking reversed operation workposition, or an error is occurred, the tracking operation gets cancelled (stopped).
5. Tracking Mode Cancelled
   Work detection is set to invalid by Tracking Mode OFF Command or an error and all the existed work information is deleted.

⚠ Caution

1) TRMD and TRAC Commands are available only when they are in the same program (task).
2) Execute the Tracking Operation ON Command on the position where there is no interference to the surroundings with the tracking operation, movement to the point above the datum point in the working range after the tracking, or the combination of both operations.
3) Ensure to use MOVL Command for the movement on the axis during the conveyor tracking operation. If the position data (Datum point position information in Work) is "obtained in normal condition", perform a positioning quickly with "MOVL" to the point around the position (around the point above the datum point in the work) considering "the target values for the axes that the data is not obtained for, such as Z-axis (height)". The obtained position data is effective only in "the tracking operation that time" and becomes ineffective after "the tracking operation that time" is complete.
   For SCARA Robot, the robot arm getting close to the area around the peculiar point as the result of conveyor tracking may cause an abnormal acceleration and it is very risky. In the case this abnormal acceleration around the peculiar point is occurred, the arm deceleration angle after the error detection also becomes larger. Do not locate interfering object in the surroundings. The following errors will be detected if the abnormal acceleration around the peculiar point:
   - Error No. B74 CP Operation Limited Area Invasion Error
   - Error No. B91 Main Excess Speed Necessity Error
   - Error No. D09 Driver Excess Speed Error

   To avoid this phenomenon, the work tracking limit can be set in "All-Axes No.75 Tracking Operation Complete Work Position", however, since it all depends on the work position, there is still a risk that the robot arm reaches the peculiar position unless it is quickly moved to the position (Datum Point in Work) obtained in normal condition by Tracking Operation ON Command.
   - It is also an effective way for the debug test operation when turning the system on to detect in the simple interference check zone before invasion to the peculiar point on the exist side if the operation is under a comparatively low conveyor speed.
   - If reaching to the axis soft limit or interference range due to the positional correlation of "minimum work position for tracking operation start" and "point above datum point for work", have an appropriate treatment on the sequence to avoid it such as by setting the different start time for the positioning to the point above the work datum point with a timer, etc.
4) Since the right for the servo use is occupied by the TRAC Command execution task during Tracking Operation ON Command, the tracking related servo axis cannot be used from other tasks. (For SCARA Robot, 4 axes are occupied for the purposes of the posture control and other related.)
5) For SCARA Robot, it is operated on the work coordinate system of when the conveyor tracking operation is started during the conveyor tracking operation.
6) Conveyor tracking operation does not stop at the break points of SEL program.
   - The break point only pauses the next program step execution.
7) PUSH Command cannot be used during the conveyor tracking operation.

[29] Vision System I/F Related Command

● SLVS (Select vision system I/F)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Outpu (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	SLVS	Select vision system I/F	(Timeout time)	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	TT:×, TTA:○	○ (PC/PG only)

[Function] Select whether using Vision System I/F in this command (GTVD Command).

Operand 1 : Select Vision System I/F
0 : Vision System I/F not selected
1 : Vision System I/F selected for use

Operand 2 : Operand 1 = Invalid when set to “0”.······Prohibited
Operand 1 = Except for “0” ···· Timeout time (sec) when GTVD Command is executed
The setting range for the timeout time is from 0.00 to 99.00sec.
When no indication (Operand 2 = blank) is defined, the timeout setting is not established and is set to no limitation.

- Return Code in SLVS Command (Variable 99 (Local Space))

The result in SLVS execution is stored in Variable 99 as a return code.
* No return code will be obtained (Variable 99 not executed) when Operand 1 = 0.
* The return codes not listed below are in common with OPEN Command (for Ethernet connection). Refer to “OPEN Command” in Ethernet Instruction Manual provided separately.

0 : Completed in normal condition
1 : Timeout
(Related Parameters: I/O Parameter No. 127, Network Attribute 8, Bits 0 to 7)
2 : Timer cancelled (condition that the waiting status is cancelled by TIMC Command)
6 : Task Complete (Program complete request, etc.)
(Unable to identify from SEL Command)
23 : Vision System Initializing Incomplete Error

(Note 1) SLVS and GTVD Commands can be executed only on the same program (task).

(Note 2) Executing SLVS Command with Operand 1 = 1 is indicated opens the communication channel that is specified in I/O Parameter No. 351, Bits 4 to 7. And also, executing SLVS Command with Operand 1 = 0 is indicated closes the communication channel that is specified in I/O Parameter No. 351, Bits 4 to 7.

(Note 3) When the Vision System I/F is used with Ethernet, message communication attribute is fixed to client.

[Example 1]

SLVS	1		Select Vision System I/F Usage (GTVD Command Timeout Value = None)
•			
•			
SLVS	0		Cancel Vision System I/F Selection

[Example 2]

SLVS	1	60	Select Vision System I/F Usage (GTVD Command Timeout Value = 60sec is indicated)
•			
•			
SLVS	0		Cancel Vision System I/F Selection

● GTVD (Vision system I/F image-capture data acquirement)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Outpu (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	GTVD	Capturing trigger classification	Variable number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	TT:×, TTA:○	○ (PC/PG only)

[Function] This outputs the image-capture command to the Vision System I/F selected by SLVS Command and stores the received image data to the variables and position data. With one time of execution of this command, one image data can be obtained.

Operand 1 : Capturing Trigger Classification
- 0 : Reserved
- 1 : Immediate Image-Capture Command Output
- 2 : Image-Capture Command Output when Image-Capture Trigger Port (I/O Port and Flag) is ON
- 3 to 6 : Reserved

Operand 2 : Variable number

Assuming the variable number selected in Operand 2 is n, the contents are stored in the variables of quantity 8 in a row starting from n.

Variable No.n : Top position number for image data work coordinates storage
No. 1 to 12 … 1 to 12 positions of centers of work piece gravity
(Note) Make sure the continuous 12 positions after the top position number are not in use.

Variable No.n+1 : Variable number for image data work attribute storage
(Note) Make sure the continuous 12 variables after the top variable number are not in use.

Variable No.n+2 : Variable number for image data work quantity storage

Variable No.n+3 : Image-capture trigger port number
(Valid only when Operand 1 = 2 is input)

Variable No.n+4 : Reserved (to be fixed to 0)

Variable No.n+5 : Reserved (to be fixed to 0)

Variable No.n+6 : Reserved (to be fixed to 0)

Variable No.n+7 : Reserved (to be fixed to 0)

- Return Code in GTVD Command (Variable 99 (Local Space))
  The result in GTVD execution is stored in Variable 99 as a return code.
  0 : Completed in normal condition
  1 : Work Information Acquirement WAIT Timeout
  2 : GTVD Timer cancelled (condition that the waiting status is cancelled by TIMC Command)
  3 : Vision System Unset Detection (SLVS Command not executed, etc.))
  4 : Work Detection Cancel Status Detection (errors, etc.)

(Note 1) SLVS and GTVD Commands can be executed only on the same program (task).

(Note 2) Receivable communication formats can be switched in I/O Parameter No. 352, Bits 0 to 7.

(Note 3) The system is capable to obtain the work data (coordinates and attributes) of 12 work pieces in 1 shot of image capturing.
Error No. 417 is issued when 13 or more work pieces are detected in 1 shot, and "4: Work Detection Cancel Status Detection (error, etc.)" is set to the return code.

(Note 4) Error No. 416 (Received Message Error) is issued when there is an error in the received message during SLVS Command execution.
Check the communication format selection parameter (I/O Parameter No. 352 or Bits 0 to 7) settings and the output communication format on Vision System side.

(Note 5) There will be no change in the variables for work attribute storage and the position data when the quantity of detected work piece in the received image data is 0.

(Note 6) It is prohibited to capture an image during the movement of the robot if the camera is mounted on the robot.
Make sure to capture an image in the stop condition.
An accurate work data cannot be acquired if a capturing is conducted during the robot movement.

[30] Anti-Vibration Control Related Command

● NTCH (Anti-Vibration Control Parameter Set Select)

Extension condition (LD, A, O, AB, OB)	Input condition (I/O, flag)	Command, declaration			Outpu (Output, flag)
		Command, declaration	Operand 1	Operand 2	
E	N, Cnd	Cmnd	Operand 1	Operand 2	Pst
Optional	Optional	NTCH	Axis pattern	Parameter set number	CC

Applicable models								
XSEL -J/K	XSEL -P/Q/ PCT/QCT	XSEL -R/S	XSEL -JX/KX	XSEL -PX/QX	XSEL -RX/SX/ RXD/SXD	ASEL PSEL SSEL	TT/TTA	MSEL
×	○	○	×	×	×	×	×	×

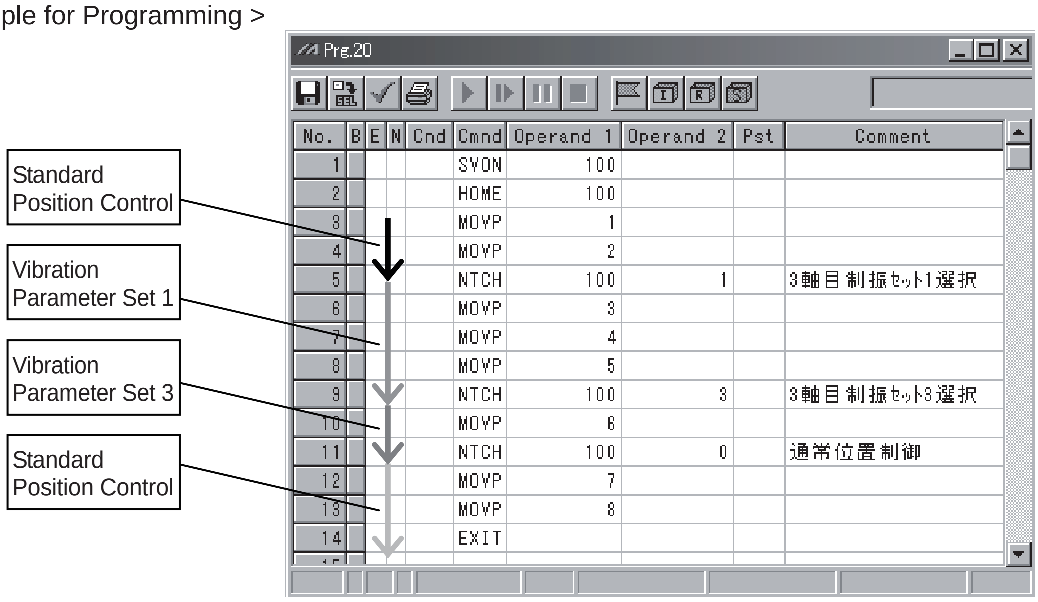
[Function] It declares what, in the specific frequency patterns registered to the parameters, is to be used for the anti-vibration control for the axis pattern set in Operand 1.

Operand 1 : Axis pattern selection
Indicate the axis that the anti-vibration control parameter set selection is conducted as “1” and the one not to be conducted as “0”.

Operand 2 : Parameter set number
Select if use/not to use the anti-vibration control and which specific frequency pattern (parameter set) is to be used.
0 : Standard Position Control (Anti-vibration control is not done)
1 : Vibration Control Parameter Set 1 (Each Axis Parameter No.151 to 154)
2 : Vibration Control Parameter Set 2 (Each Axis Parameter No.156 to 159)
3 : Vibration Control Parameter Set 3 (Each Axis Parameter No.161 to 164)
Except for 0 to 3 : Standard Position Control (Anti-vibration control is not done)

[Example] NTCH 110 2 Setting the anti-vibration control parameter set 2 to the 2nd and 3rd axes

<Example for Programming >

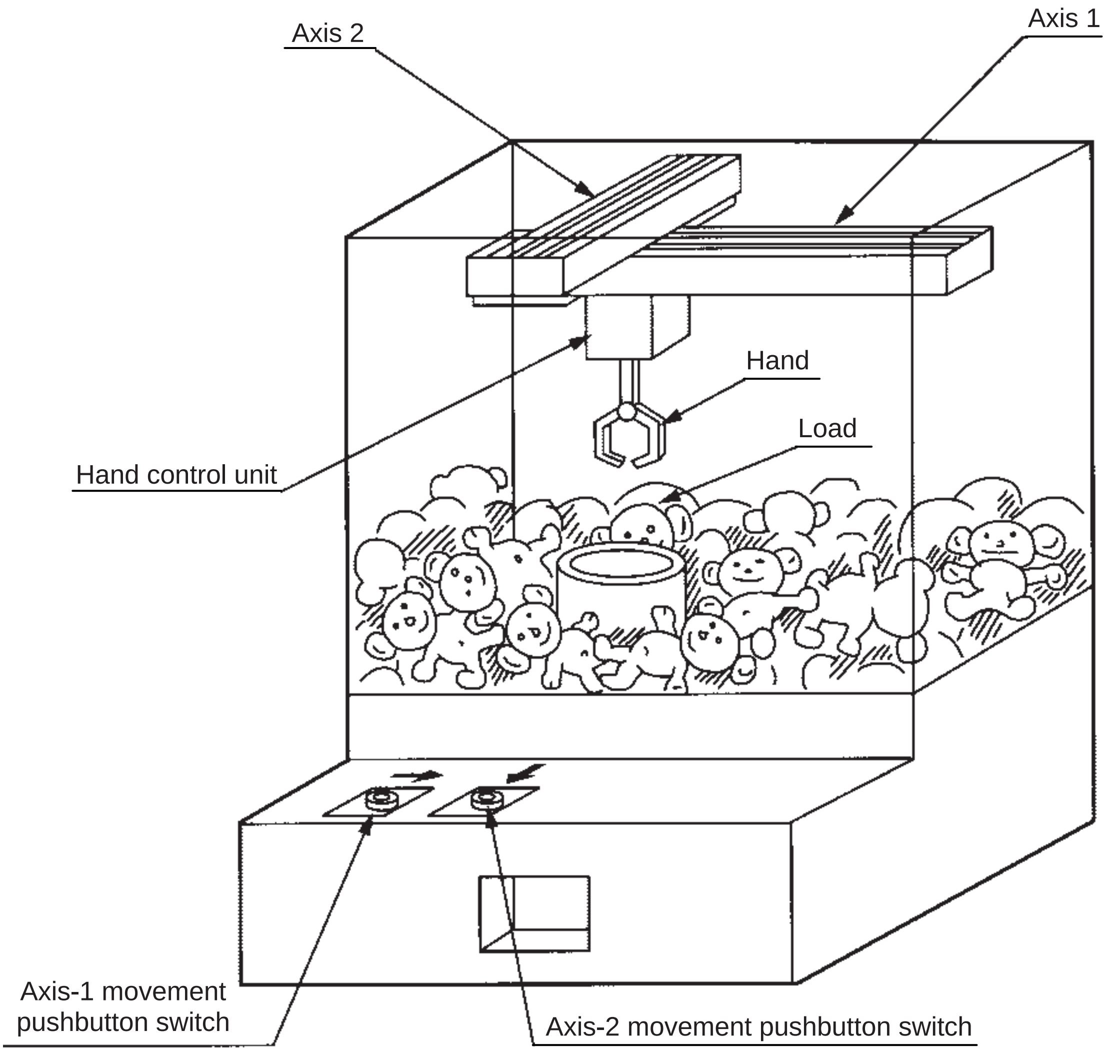


6. Program Examples

6.1 Operation by Jog Command Doll-Picking Game Machine

(1) Overview of the system

This system is a doll-picking game machine consisting of axis-1 and axis-2 actuators. Pushbutton switches corresponding to the two axes are provided on an external operation switch box, and these switches are used to move the actuators to a desired position to grab and pick up dolls inside the case.



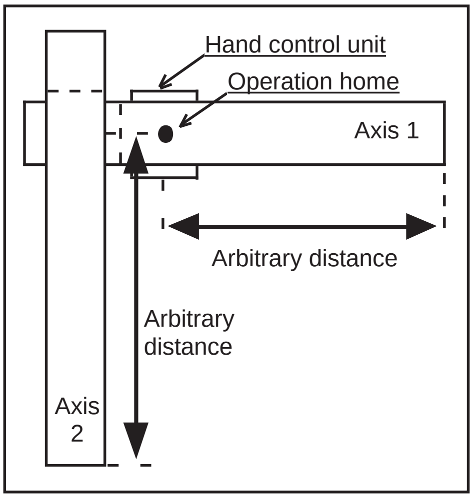
(2) Explanation of the operation

How this system operates is explained.

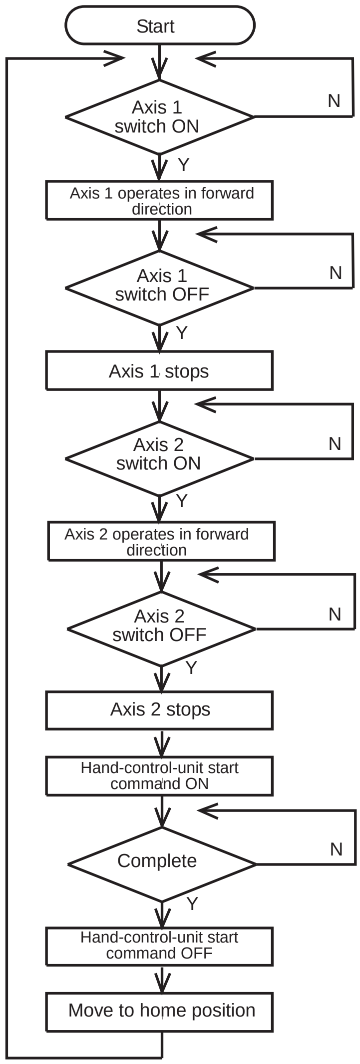
1) Wait for the axis-1 movement pushbutton switch to turn ON.
2) The X-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
3) Wait for the axis-2 movement pushbutton switch to turn ON.
4) The Y-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
5) Output a start command to the hand control unit.
6) Wait for an operation completion input from the hand control unit.
7) Move to the home after the input is received.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position



Operation Flow Chart



I/O Assignments

Category		I/O No.	Signal name	Specification
XSEL	Input	16	Axis-1 movement command	Pushbutton switch
		17	Axis-2 movement command	Pushbutton switch
		18	Hand operation completion	External control unit
	Output	309	Hand start command	24V DC
	* Flag is not used.			

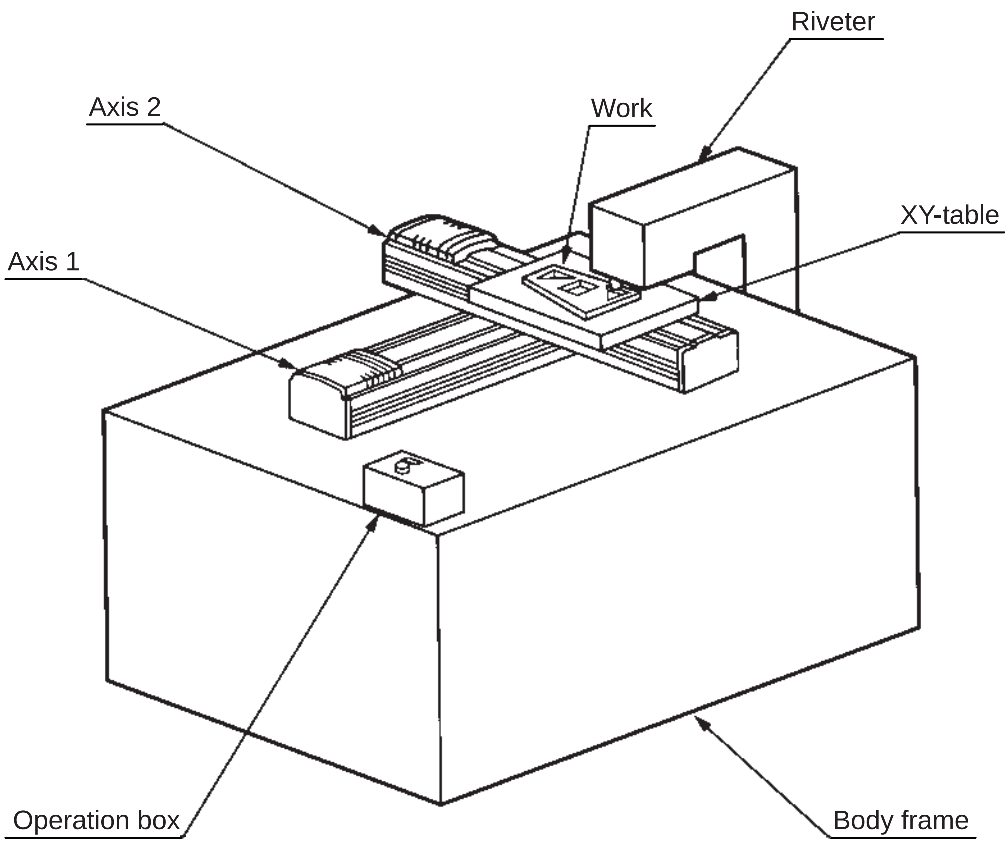
(3) XSEL Controller application program

Step	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1				HOME	11			Axes 1 and 2 return to home (servo ON).
2				VEL	400			Set speed to 400mm/s.
3				TAG	1			
4				WTON	16			Wait for input from axis-1 movement switch.
5				JFWN	1	16		Move forward while axis-1 movement switch is ON.
6				WTON	17			Wait for input from axis-2 movement switch.
7				JFWN	10	17		Move forward while axis-2 movement switch is ON.
8				BTON	309			Start command for external control unit turns ON.
9				WTON	18			Wait for external control unit to complete operation.
10				BTOF	309			Start command for external control unit turns OFF.
11				JBWF	11	18		Axes 1 and 2 move backward while 18 is ON.
12				GOTO	1			Jump to TAG1.
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								

6.2 Operation by Point Movement Command Riveting System

(1) Overview of the system

This system is a riveting system consisting of an XY-table operated by axis-1 and axis-2 actuators and a riveter. By setting a work on the XY-table at the operation home and turning ON the start switch, rivets will be driven at the three points specified on the work.

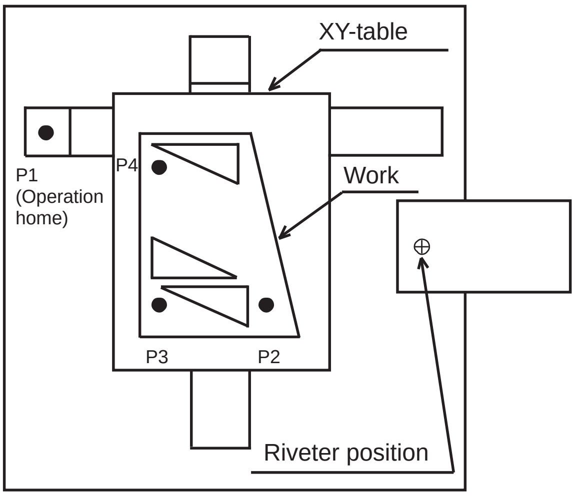


(2) Explanation of the operation
How this system operates is explained.

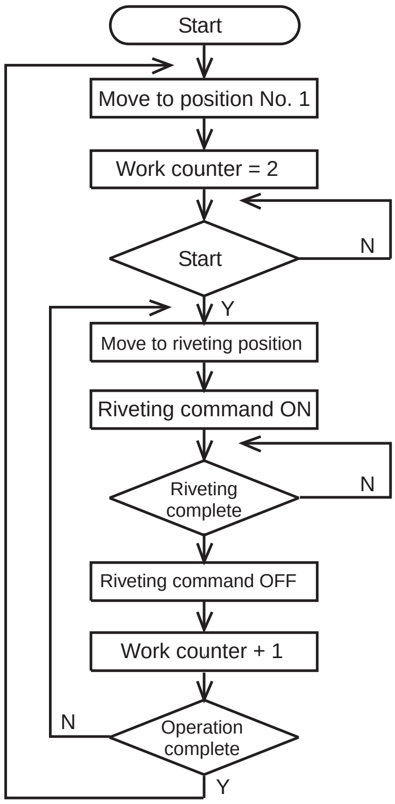
1) The XY-table moves to the operation home (P1) and waits.
2) The operator sets a work on the XY-table and turns ON the start switch.
3) The XY-table moves to riveting position No. 1 (P2) on the work and a riveting command is output to the riveter.
4) When the riveter completes the riveting operation and a completion signal is input, the table will move to riveting position No. 2 (P3) and then No. 3 (P4), in the same manner.
5) When all three points have been riveted, the table will return to the operation home (P1).

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position



Operation Flow Chart



I/O Assignments

Category		I/O No.	Signal name	Specification
XSEL	Input	16	Start command	Pushbutton switch
		17	Riveting completion	Contact signal
	Output	309	Riveting command	24V DC
	* Flag is used from 600.			

(3) XSEL Controller application program

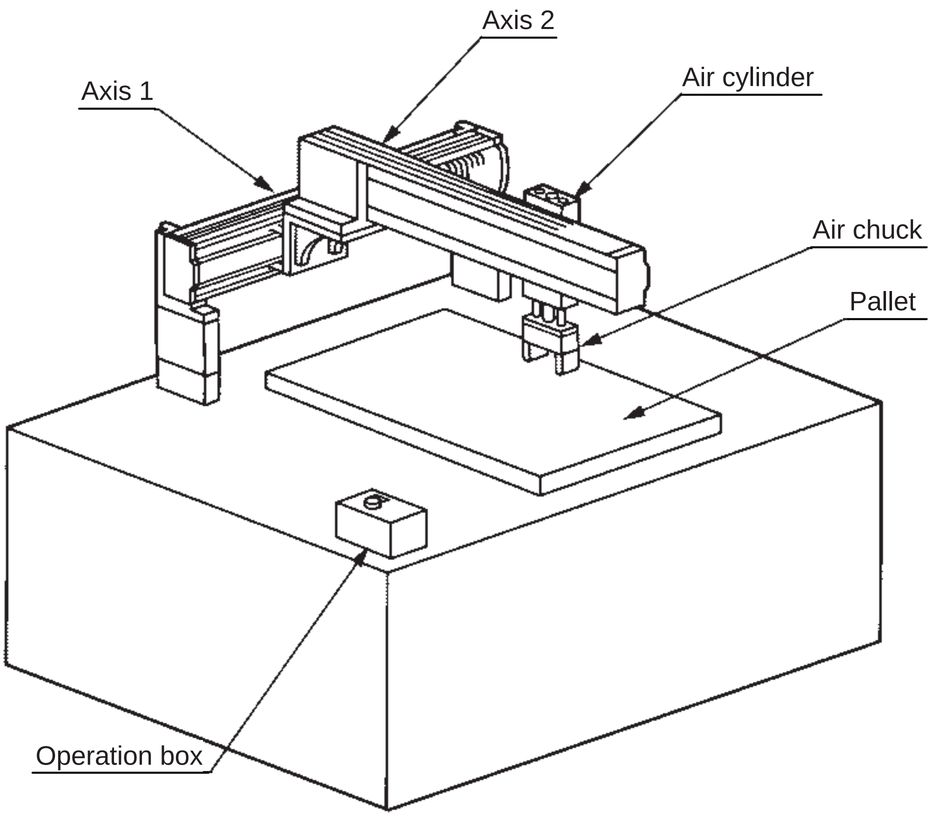
Step	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1				HOME	11			XY-table returns to home (servo ON).
2				VEL	400			Set speed to 400mm/s.
3				TAG	1			
4				MOVL	1			Move to position No. 1 (home of work).
5				LET	1	2		Set 2 in work counter.
6				BTOF	600			Clear completion flag.
7				WTON	16			Wait for start command.
8				TAG	2			
9				MOVL	*1			Move to work counter position.
10				BTON	309			Riveting command turns ON.
11				WTON	17			Wait for riveting to complete.
12				BTOF	309			Riveting command turns OFF.
13				ADD	1	1		Increment work counter by 1.
14				CPEQ	1	5	600	Turns ON flag if operation is complete.
15		N	600	GOTO	2			Jump to TAG2 if not complete.
16				GOTO	1			Jump to TAG1 if complete.
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								

6.3 Palletizing Operation Palletizing System

(1) Overview of the system

This system is a palletizing system consisting of axis-1 and axis-2 actuators and a Z-axis air cylinder. It clamps a work at the work feed point and transfers it onto a pallet, and repeats this operation in a sequence.

(Operation is implemented by an offset command without using a palletizing function.)

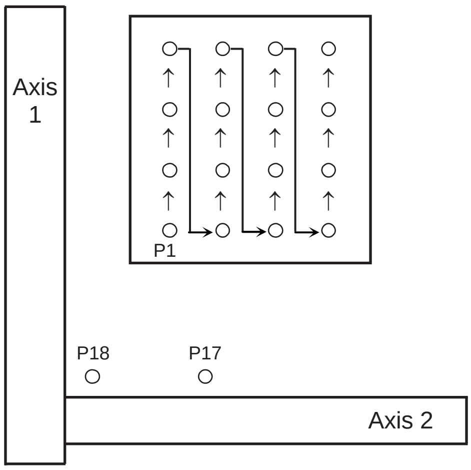


(2) Explanation of the operation
How this system operates is explained.

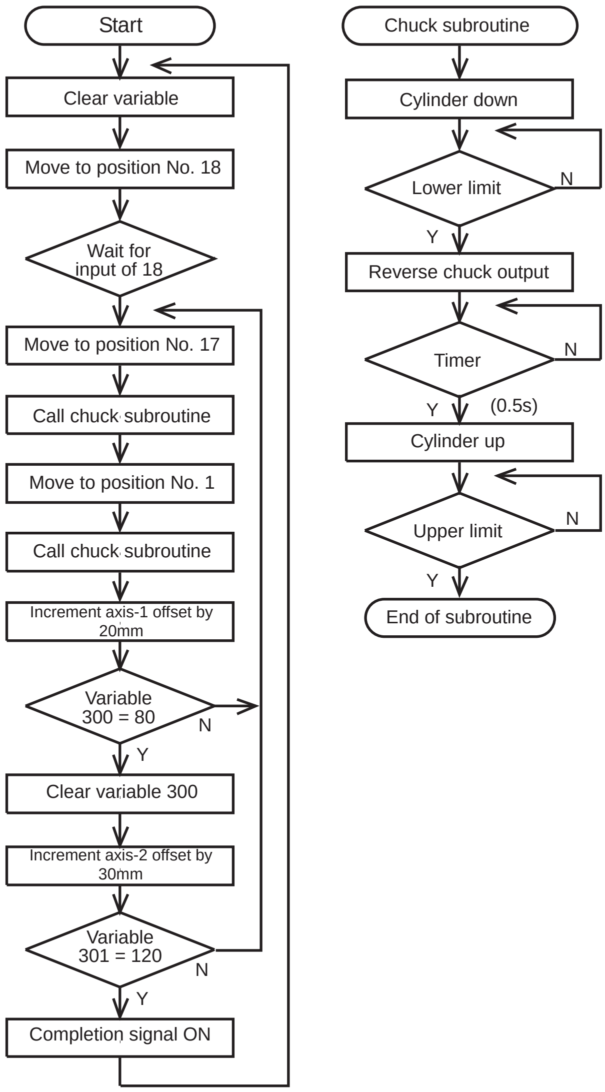
1) Move to the standby point and wait for a start input.
2) Move to the work feed point after a start input is received.
3) The Z-axis comes down and the air chuck clamps the work.
4) The Z-axis rises and moves to above the pallet.
5) The Z-axis comes down and releases the work.
6) The Z-axis rises and moves to above the work feed point.
7) When the pallet becomes full, a pallet-completion indicator signal is output. The axes move to P18 and then wait for restart.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position



Operation Flow Chart



I/O Assignments

Category		I/O No.	Signal name	Specification
XSEL	Input	16	Z-axis cylinder upper limit	Proximity SW
		17	Z-axis cylinder lower limit	Proximity SW
		18	Start	Pushbutton switch
	Output	309	Z-axis cylinder SV	24V DC
		310	Z-axis chuck SV	24V DC
		311	Pallet-completion indicator	24V DC
	* Flag is used from 600.			

Pallet specifications
Axis-1 direction: 20mm pitch
Axis-2 direction: 30mm pitch

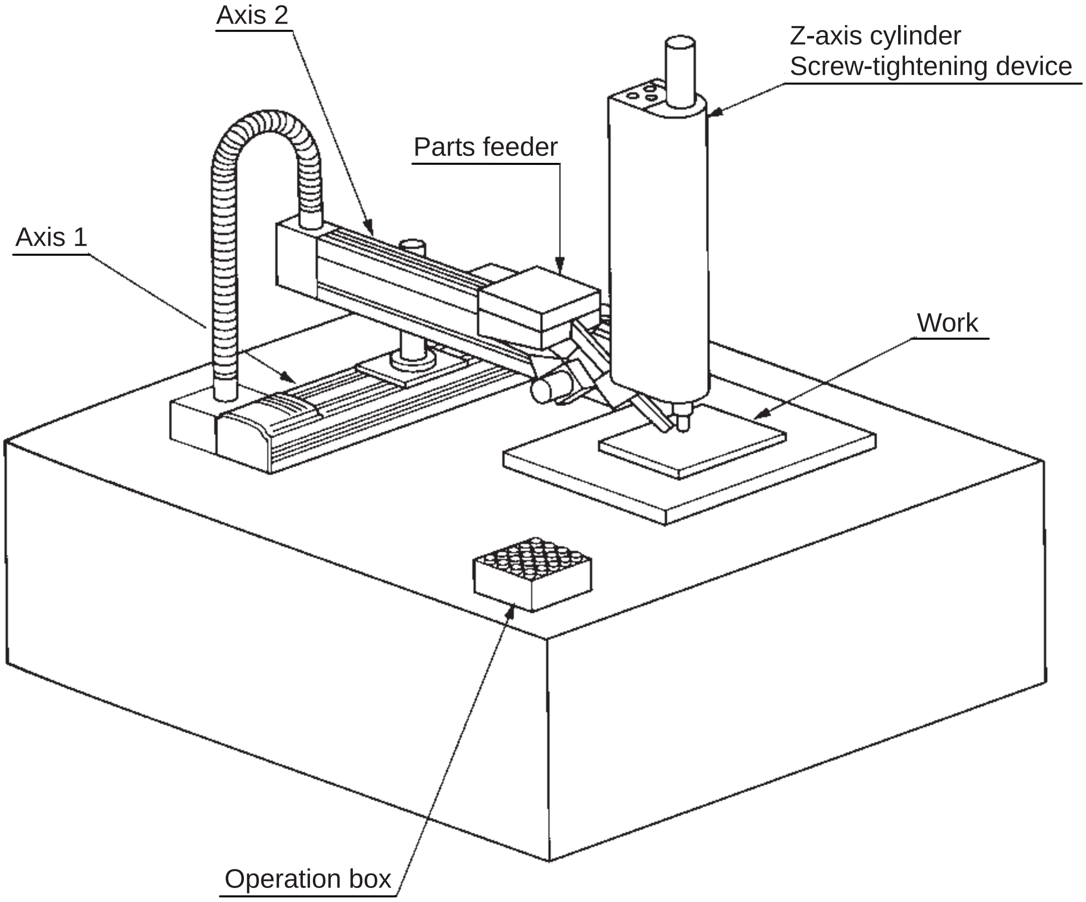
(3) XSEL Controller application program

Step	E	N	Cnd	Cmnd	Operand 1	Operand 2	Pst	Comment
1				HOME	11			Axes 1 and 2 return to home.
2				VEL	100			Set speed to 100mm/s.
3				ACC	0.2			Acceleration/deceleration: 0.2G
4				TAG	1			
5				LET	300	0		Clear variable.
6				LET	301	0		Clear variable.
7				OFST	11	0		Clear offset value.
8				MOVL	18			Move to position No. 18.
9				WTON	18			Wait for start input.
10				BTOF	311			Output 311 turns OFF.
11				TAG	2			
12				OFST	11	0		Clear offset value.
13				MOVL	17			Move to position No. 17.
14				EXSR	1			Call chuck subroutine (chuck).
15				OFST	1	*300		Offset axis 1 by value in variable 300.
16				OFST	10	*301		Offset axis 2 by value in variable 301.
17				MOVL	1			Move to position No. 1 + offset value.
18				EXSR	1			Call chuck subroutine (unchuck).
19				ADD	300	20		Add 20 to variable 300.
20				CPEQ	300	80	600	Turn ON flag 600 if variable 300 = 80.
21		N	600	GOTO	2			Jump to TAG2 if flag 600 is OFF.
22				LET	300	0		Clear variable 300.
23				ADD	301	30		Add 30 to variable 301.
24				CPEQ	301	120	601	Turn ON flag 601 if variable 301 = 120.
25		N	601	GOTO	2			Jump to TAG2 if flag 601 is OFF.
26				BTON	311			Output 311 turns ON.
27				GOTO	1			Jump to TAG1.
28				BGSR	1			Start chuck subroutine.
29				BTON	309			Z-axis cylinder down
30				WTON	17			Wait for lower-limit input.
31				BTNT	310			Reverse air-chuck output.
32				TIMW	0.5			Timer: 0.5 second
33				BTOF	309			Z-axis cylinder up
34				WTON	16			Wait for upper-limit input.
35				EDSR				End of chuck subroutine
36								
37								
38								
39								

6.4 Screw-Tightening Machine

(1) Overview of the system

This system consists of axis-1 and axis-2 actuators, Z-axis cylinder, screw-tightening device and parts feeder, and tightens the screws fed by the parts feeder at the specified positions on the work.



(2) Equipment

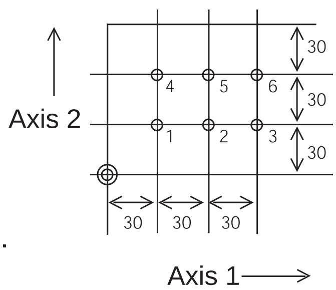
Screw-tightening machine (for Z-axis)	
Actuators (for axes 1 and 2)	IAI's 60W servo motor/actuator with 300mm stroke × 2
Controller	IAI's XSEL controller

(3) Explanation of the operation

(1) Tighten six screws at 30mm pitches on axes 1 and 2.
1) The actuators move to a screw-tightening position.
2) The Z-axis air cylinder of the screw-tightening machine comes down.
3) The screw-tightening machine starts operating.
4) When the screw tightening is complete, the Z-axis air cylinder rises.
5) The actuators move to the next position.

(2) The parts feeder operates in parallel with the above operation.
1) The parts feeder starts when screws are short.
2) The parts feeder stops when the screws are fully loaded.

Coordinates



[Hardware]

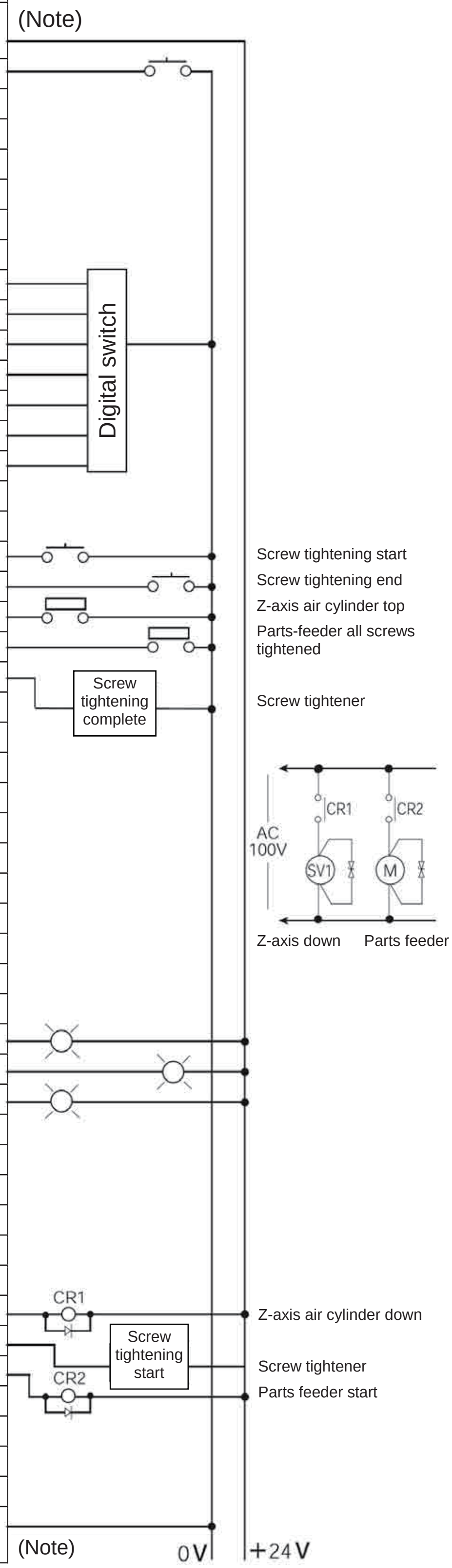
1) I/O assignment

I/O connector (50 pins)

Pin No.	Category	Port No.	Function	Cable color
1	Input	–	General-purpose: NC / Compact: +24V input	Brown-1
2		000	Program start	Red-1
3		001	General-purpose input	Orange-1
4		002	General-purpose input	Yellow-1
5		003	General-purpose input	Green-1
6		004	General-purpose input	Blue-1
7		005	General-purpose input	Purple-1
8		006	General-purpose input	Gray-1
9		007	Program specification (PRG No.1)	White-1
10		008	Program specification (PRG No.2)	Black-1
11		009	Program specification (PRG No.4)	Brown-2
12		010	Program specification (PRG No.8)	Red-2
13		011	Program specification (PRG No.10)	Orange-2
14		012	Program specification (PRG No.20)	Yellow-2
15		013	Program specification (PRG No.40)	Green-2
16		014	General-purpose input	Blue-2
17		015	General-purpose input	Purple-2
18		016	Screw tightening start	Gray-2
19		017	Screw tightening end	White-2
20		018	Z-axis air cylinder top	Black-2
21		019	Parts-feeder all screws tightened	Brown-3
22		020	Screw tightening complete	Red-3
23		021	General-purpose input	Orange-3
24		022	General-purpose input	Yellow-3
25		023	General-purpose input	Green-3
26		024	General-purpose input	Blue-3
27		025	General-purpose input	Purple-3
28		026	General-purpose input	Gray-3
29		027	General-purpose input	White-3
30		028	General-purpose input	Black-3
31		029	General-purpose input	Brown-4
32		030	General-purpose input	Red-4
33		031	General-purpose input	Orange-4
34	Output	300	Alarm output	Yellow-4
35		301	Ready output	Green-4
36		302	Emergency stop output	Blue-4
37		303	General-purpose output	Purple-4
38		304	General-purpose output	Gray-4
39		305	General-purpose output	White-4
40		306	General-purpose output	Black-4
41		307	General-purpose output	Brown-5
42		308	General-purpose output	Red-5
43		309	Z-axis air cylinder down	Orange-5
44		310	Screw tightening start	Yellow-5
45		311	Parts feeder start	Green-5
46		312	General-purpose output	Blue-5
47		313	General-purpose output	Purple-5
48		314	General-purpose output	Gray-5
49		315	General-purpose output	White-5
50		–	General-purpose: NC / Compact: 0V	Black-5

2) Layout drawing

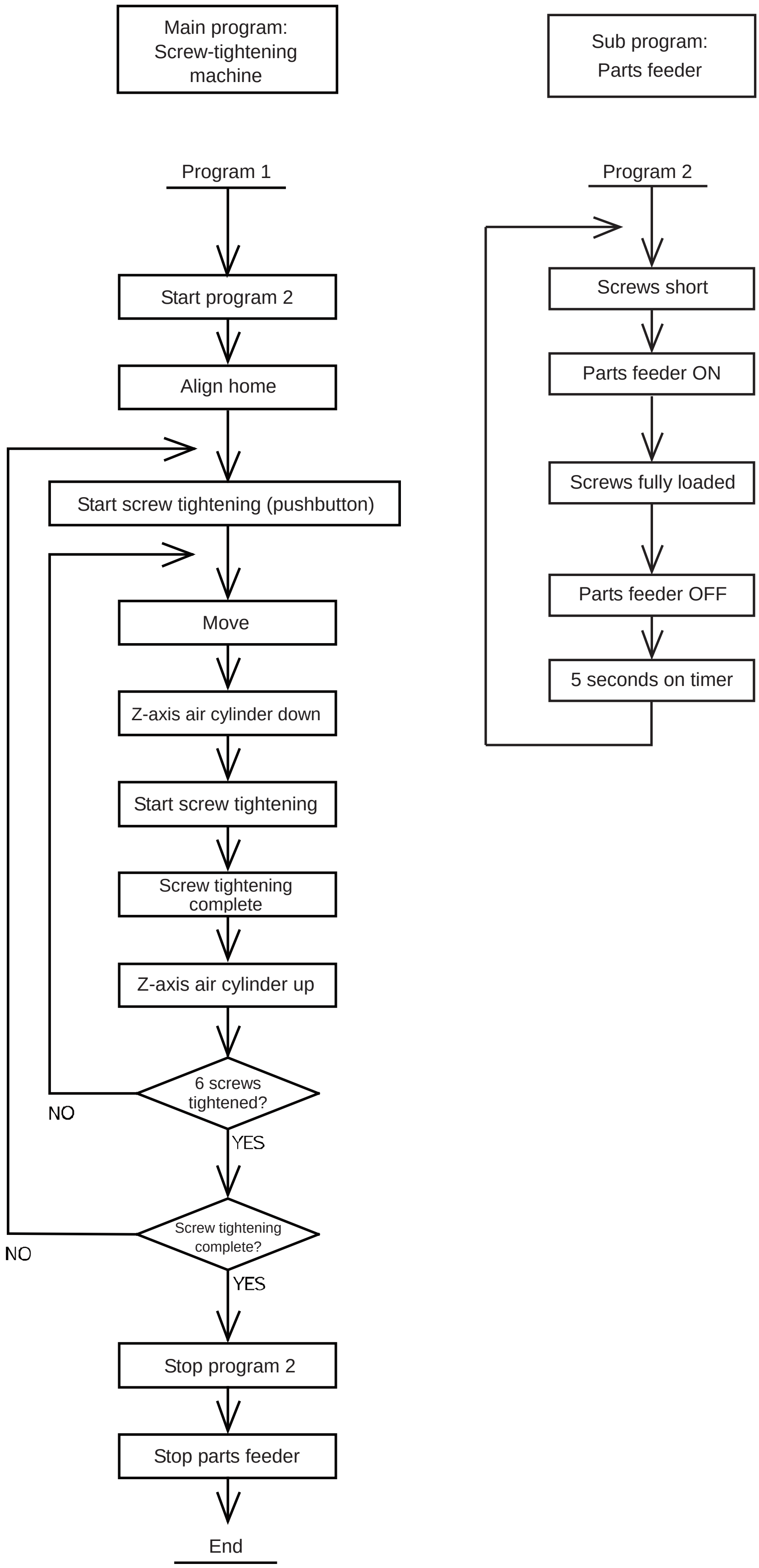
Pin No.	Category	Port No.	Function
1	Input	–	General-purpose: NC / Compact: +24V input
2		000	Program start
3		001	General-purpose input
4		002	General-purpose input
5		003	General-purpose input
6		004	General-purpose input
7		005	General-purpose input
8		006	General-purpose input
9		007	Program specification (PRG No.1)
10		008	Program specification (PRG No.2)
11		009	Program specification (PRG No.4)
12		010	Program specification (PRG No.8)
13		011	Program specification (PRG No.10)
14		012	Program specification (PRG No.20)
15		013	Program specification (PRG No.40)
16		014	General-purpose input
17		015	General-purpose input
18		016	Screw tightening start
19		017	Screw tightening end
20		018	Z-axis air cylinder top
21		019	Parts-feeder all screws tightened
22		020	Screw tightening complete
23		021	General-purpose input
24		022	General-purpose input
25		023	General-purpose input
26		024	General-purpose input
27		025	General-purpose input
28		026	General-purpose input
29		027	General-purpose input
30		028	General-purpose input
31		029	General-purpose input
32		030	General-purpose input
33		031	General-purpose input
34	Output	300	Alarm output
35		301	Ready output
36		302	Emergency stop output
37		303	General-purpose output
38		304	General-purpose output
39		305	General-purpose output
40		306	General-purpose output
41		307	General-purpose output
42		308	General-purpose output
43		309	Z-axis air cylinder down
44		310	Screw tightening start
45		311	Parts feeder start
46		312	General-purpose output
47		313	General-purpose output
48		314	General-purpose output
49		315	General-purpose output
50		–	General-purpose: NC / Compact: 0V



Pin No. 1 and 50 are not connected for general-purpose types.
Pin No. 1 is connected to +24V, while Pin No. 50 is connected to 0V, for compact types.

[Software]

1) Control flow chart



2) Main program
Screw-tightening program No. 1

Application program

Comment	Extension condition	Input condition	Command			Output condition	Comment
	AND, OR	I/O, flag	Command	Operand 1	Operand 2	Output port, flag	
1			EXPG	2			Start program 2.
2			HOME	11			Align home.
3			VEL	100			Speed: 100mm/sec
4			ACC	0.3			Acceleration: 0.3G
5			TAG	1			Jump destination at restart
6			WTON	16			Screw-tightening start pushbutton
7			LET	1	1		Set screw counter.
8			TAG	2			Jump destination after tightening one screw
9			MOVL	*1			Move.
10			BTON	309			Z-axis air cylinder down
11			BTON	310			Start screw tightening.
12			WTON	20			Screw tightening complete.
13			BTOF	309	310		Cylinder up, screw tightening stopped.
14			WTON	18			Check Z-axis air cylinder top position.
15			ADD	1	1		Increment screw counter by 1.
16			CPEQ	1	7	900	Compare after tightening six screws.
17		N900	GOTO	2			Go to next screw-tightening cycle after tightening one screw.
18		N17	GOTO	1			Restart screw tightening.
19			ABPG	2			Stop program 2.
20			BTOF	311			Stop parts feeder.
21			EXIT				End of program 1

Position program

No.	X	Y
1	30	30
2	60	30
3	90	30
4	30	60
5	60	60
6	90	60

3) Sub program
Parts feeder program No. 2

Application program

Comment	Extension condition	Input condition	Command			Output condition	Comment
	AND, OR	I/O, flag	Command	Operand 1	Operand 2	Output port, flag	
1			TAG	1			Jump destination for repeating
2			WTOF	19			Screws short.
3			BTON	311			Start parts feeder.
4			WTON	19			Screws fully loaded.
5			BTOF	311			Stop parts feeder.
6			TIMW	5			5 seconds on restart timer
7			GOTO	1			Repeat.

7. Appendix

ASCII Code Table

Upper 3 bits → / ↓ Lower 4 bits	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF/NL	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Change History

Revision Date	Description of Revision
2010.11	First edition 1A → 1B (note corrected) Correction of page number for each command in pages 1 to 19, 254 to 260 Pages 44 Correction in reference for (3) Pages 103 Correction in right-hand rule Pages 248 Addition of note to state to refer to TT Instruction Manual Pages 295 Correction to the note of continuous operation command reference Pages 489 and 491 Correction to the note of palletizing reference
2011.11	Second edition Contents changed in Safety Guide Caution notes added for when working with two or more persons Contents deleted regarding Vertical Articulated and Rectangular 6-Axes Robots
2011.12	Edition 2B Note corrected etc.
2012.04	Third edition Note added for PCT/QCT Types for CT4 Actuator and explanation changed for related command (SCRV)
2012.09	Fourth edition Note added for XSEL-R/S/RX/SX/RXD/SXD types and related commands added and changed
2012.10	Fifth edition Note revised
2013.10	Sixth edition TTA added
2014.06	Seventh edition MSEL added
2014.08	Eighth edition Notes added for MSEL Cartesian Type application

Manual No.: ME0224-8A (August 2014)

IAI
Quality and Innovation

IAI Corporation

Head Office: 577-1 Obane Shimizu-KU Shizuoka City Shizuoka 424-0103, Japan
TEL +81-54-364-5105 FAX +81-54-364-2589
website: www.iai-robot.co.jp/

Technical Support available in USA, Europe and China

IAI America, Inc.

Head Office: 2690 W. 237th Street, Torrance, CA 90505
TEL (310) 891-6015 FAX (310) 891-0815
Chicago Office: 1261 Hamilton Parkway, Itasca, IL 60143
TEL (630) 467-9900 FAX (630) 467-9912
Atlanta Office: 1220 Kennestone Circle, Suite 108, Marietta, GA 30066
TEL (678) 354-9470 FAX (678) 354-9471
website: www.intelligentactuator.com

IAI Industrieroboter GmbH

Ober der Röth 4, D-65824 Schwalbach am Taunus, Germany
TEL 06196-88950 FAX 06196-889524

IAI (Shanghai) Co., Ltd.

SHANGHAI JIAHUA BUSINESS CENTER A8-303, 808, Hongqiao Rd. Shanghai 200030, China
TEL 021-6448-4753 FAX 021-6448-3992
website: www.iai-robot.com

IAI Robot (Thailand) Co., Ltd.

825, PhairojKijja Tower 12th Floor, Bangna-Trad RD., Bangna, Bangna, Bangkok 10260, Thailand
TEL +66-2-361-4458 FAX +66-2-361-4456

14.08.000