MITSUBISHI ELECTRIC

Mitsubishi Electric Industrial Robot

CR800-R/CR800-Q series controller

iQ Platform Supporting Extended Function Instruction Manual

MELFA

BFP-A3528-C

⚠ Safety Precautions

Always read the following precautions and the separate "Safety Manual" before starting use of the robot to learn the required measures to be taken.

⚠ CAUTION

All teaching work must be carried out by an operator who has received special training. (This also applies to maintenance work with the power source turned ON.)
Enforcement of safety training

⚠ CAUTION

For teaching work, prepare a work plan related to the methods and procedures of operating the robot, and to the measures to be taken when an error occurs or when restarting. Carry out work following this plan. (This also applies to maintenance work with the power source turned ON.)
Preparation of work plan

⚠ WARNING

Prepare a device that allows operation to be stopped immediately during teaching work. (This also applies to maintenance work with the power source turned ON.)
Setting of emergency stop switch

⚠ CAUTION

During teaching work, place a sign indicating that teaching work is in progress on the start switch, etc. (This also applies to maintenance work with the power source turned ON.)
Indication of teaching work in progress

⚠ DANGER

Provide a fence or enclosure during operation to prevent contact of the operator and robot.
Installation of safety fence

⚠ CAUTION

Establish a set signaling method to the related operators for starting work, and follow this method.
Signaling of operation start

⚠ CAUTION

As a principle turn the power OFF during maintenance work. Place a sign indicating that maintenance work is in progress on the start switch, etc.
Indication of maintenance work in progress

⚠ CAUTION

Before starting work, inspect the robot, emergency stop switch and other related devices, etc., and confirm that there are no errors.
Inspection before starting work

The points of the precautions given in the separate "Safety Manual" are given below.
Refer to the actual "Safety Manual" for details.

⚠ DANGER	When automatic operation of the robot is performed using multiple control devices (GOT, programmable controller, push-button switch), the interlocking of operation rights of the devices, etc. must be designed by the customer.
⚠ CAUTION	Use the robot within the environment given in the specifications. Failure to do so could lead to a drop or reliability or faults. (Temperature, humidity, atmosphere, noise environment, etc.)
⚠ CAUTION	Transport the robot with the designated transportation posture. Transporting the robot in a non-designated posture could lead to personal injuries or faults from dropping.
⚠ CAUTION	Always use the robot installed on a secure table. Use in an instable posture could lead to positional deviation and vibration.
⚠ CAUTION	Wire the cable as far away from noise sources as possible. If placed near a noise source, positional deviation or malfunction could occur.
⚠ CAUTION	Do not apply excessive force on the connector or excessively bend the cable. Failure to observe this could lead to contact defects or wire breakage.
⚠ CAUTION	Make sure that the workpiece weight, including the hand, does not exceed the rated load or tolerable torque. Exceeding these values could lead to alarms or faults.
⚠ WARNING	Securely install the hand and tool, and securely grasp the workpiece. Failure to observe this could lead to personal injuries or damage if the object comes off or flies off during operation.
⚠ WARNING	Securely ground the robot and controller. Failure to observe this could lead to malfunctioning by noise or to electric shock accidents.
⚠ CAUTION	Indicate the operation state during robot operation. Failure to indicate the state could lead to operators approaching the robot or to incorrect operation.
⚠ WARNING	When carrying out teaching work in the robot's movement range, always secure the priority right for the robot control. Failure to observe this could lead to personal injuries or damage if the robot is started with external commands.
⚠ CAUTION	Keep the jog speed as low as possible, and always watch the robot. Failure to do so could lead to interference with the workpiece or peripheral devices.
⚠ CAUTION	After editing the program, always confirm the operation with step operation before starting automatic operation. Failure to do so could lead to interference with peripheral devices because of programming mistakes, etc.
⚠ CAUTION	Make sure that if the safety fence entrance door is opened during automatic operation, the door is locked or that the robot will automatically stop. Failure to do so could lead to personal injuries.
⚠ CAUTION	Never carry out modifications based on personal judgments, or use non-designated maintenance parts. Failure to observe this could lead to faults or failures.

⚠ WARNING

When the robot arm has to be moved by hand from an external area, do not place hands or fingers in the openings. Failure to observe this could lead to hands or fingers catching depending on the posture.

⚠ CAUTION

Do not stop the robot or apply emergency stop by turning the robot controller's main power OFF. If the robot controller main power is turned OFF during automatic operation, the robot accuracy could be adversely affected. Moreover, it may interfere with the peripheral device by drop or move by inertia of the arm.

⚠ CAUTION

Do not turn off the main power to the robot controller while rewriting the internal information of the robot controller such as the program or parameters. If the main power to the robot controller is turned off while in automatic operation or rewriting the program or parameters, the internal information of the robot controller may be damaged.

⚠ DANGER

Do not connect the Handy GOT when using the GOT direct connection function of this product. Failure to observe this may result in property damage or bodily injury because the Handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not.

⚠ DANGER

Do not connect the Handy GOT to a programmable controller when using an iQ Platform compatible product with the CR800-R/CR800-Q controller. Failure to observe this may result in property damage or bodily injury because the Handy GOT can automatically operate the robot regardless of whether the operation rights are enabled or not.

⚠ DANGER

Do not remove the SSCNET III cable while power is supplied to the multiple CPU system or the servo amplifier. Do not look directly at light emitted from the tip of SSCNET III connectors or SSCNET III cables of the Motion CPU or the servo amplifier. Eye discomfort may be felt if exposed to the light. (Reference: SSCNET III employs a Class 1 or equivalent light source as specified in JIS C 6802 and IEC60825-1 (domestic standards in Japan).)

⚠ DANGER

Do not remove the SSCNET III cable while power is supplied to the controller. Do not look directly at light emitted from the tip of SSCNET III connectors or SSCNET III cables. Eye discomfort may be felt if exposed to the light. (Reference: SSCNET III employs a Class 1 or equivalent light source as specified in JIS C 6802 and IEC60825-1 (domestic standards in Japan).)

⚠ DANGER

Attach the cap to the SSCNET III connector after disconnecting the SSCNET III cable. If the cap is not attached, dirt or dust may adhere to the connector pins, resulting in deterioration connector properties, and leading to malfunction.

⚠ CAUTION

Make sure there are no mistakes in the wiring. Connecting differently to the way specified in the manual can result in errors, such as the emergency stop not being released. In order to prevent errors occurring, please be sure to check that all functions (such as the teaching box emergency stop, customer emergency stop, and door switch) are working properly after the wiring setup is completed.

⚠ CAUTION

Use the network equipments (personal computer, USB hub, LAN hub, etc) confirmed by manufacturer. The thing unsuitable for the FA environment (related with conformity, temperature or noise) exists in the equipments connected to USB. When using network equipment, measures against the noise, such as measures against EMI and the addition of the ferrite core, may be necessary. Please fully confirm the operation by customer. Guarantee and maintenance of the equipment on the market (usual office automation equipment) cannot be performed.

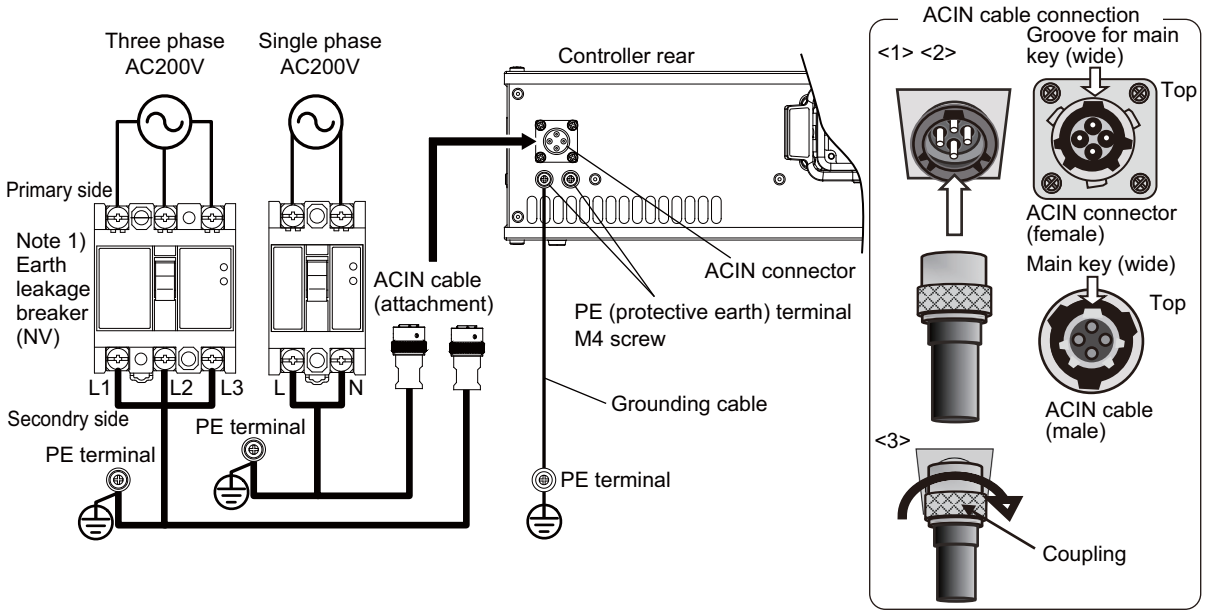
⚠ CAUTION

To maintain the safety of the robot system against unauthorized access from external devices via the network, take appropriate measures.
To maintain the safety against unauthorized access via the Internet, take measures such as installing a firewall.

*CR800 controller

Notes of the basic component are shown.

⚠ CAUTION Please install the earth leakage breaker in the primary side power supply of the controller because of leakage protection.



Note 1) Always use the terminal cover for the earth leakage breaker.

1) Prepare the following items.

Part name	Specifications	Remarks
Earth leakage breaker	The following is recommended product. Single phase: NV30FAU-2P-10A-AC100-240V-30mA (Terminal cover: TCS-05FA2) Three phase: NV30FAU-3P-10A-AC100-240V-30mA (Terminal cover: TCS-05FA3)	Prepared by customer.
Cable for primary power supply	AWG14 ($2mm^2$) or above	Prepared by customer. Tightening torque for terminal fixing screw is 2 ～ 3Nm.
Grounding cable	AWG14 ($2mm^2$) or above	Prepared by customer. Tightening torque for terminal fixing screw is 2 ～ 3Nm.
ACIN cable	Terminal: M5, cable length: 3m	Supplied with the product.

2) Confirm that the primary power matches the specifications.
3) Confirm that the primary power is OFF and that the earth leakage breaker power switch is OFF.
4) Connect the ACIN cable to the breaker.
   Connect the power terminals of the ACIN cable to the secondary side terminals of the earth leakage breaker. Also, ground the FG terminal of the cable.
5) Connect the ACIN cable to the ACIN connector on the rear of the controller.
   <1> Face the main key on the ACIN cable plug upwards. (Refer to the "ACIN cable connection" illustration.)
   <2> Align the main key of the ACIN cable plug with the grooves on the ACIN connector. Push the plug into the connector as far as it will go.
   The plug may be damaged if it is not correctly aligned with the connector.
   <3> Tighten the coupling on the ACIN cable, turning it to the right until it locks.
6) Connect one end of the grounding cable to the PE (protective earth) terminal on the controller and ground the other end (2-point grounding) in order to comply with the requirements of EN 61800-5-1 for the touch current of 3.5 mA AC or more.
7) Connect the primary power cable to the primary side terminal of the earth leakage breaker.

Revision history

Date of print	Specifications No.	Details of revisions
2017-05-17	BFP-A3528	• First edition created.
2017-07-03	BFP-A3528-A	• “2.1.1 Set up Sequencer's Multiple CPUs” was modified.
2018-03-01	BFP-A3528-B	• Descriptions of CR800-Q controller were added.
2018-12-25	BFP-A3528-C	• Added further explanation of the ACIN cable.

*Introduction

Thank you for buying the industrial robot MELFA manufactured by Mitsubishi Electric.
This document provides the instructions for iQ Platform supporting extended functions in CR800-R/CR800-Q series robot controller. Our extended functions allows the sequencer easily to monitor the robot through CPU buffer memory between sequencer and robot, set up data, and operate the robot without a program (sequencer direct performance). This document provides a detailed description of the data configuration of the CPU buffer memory*, monitoring, and operation procedures. (*shared memory if using the CR800-Q)

Please carefully read and fully understand this document before making use of the extended functions.

Target controller of this document
This document supports the robot controller below:

- CR800-R/CR800-Q series controller

<Precautions>
Read "CPU buffer memory" as "CPU shared memory" if using the CR800-Q.

Contents

Contents

Contents

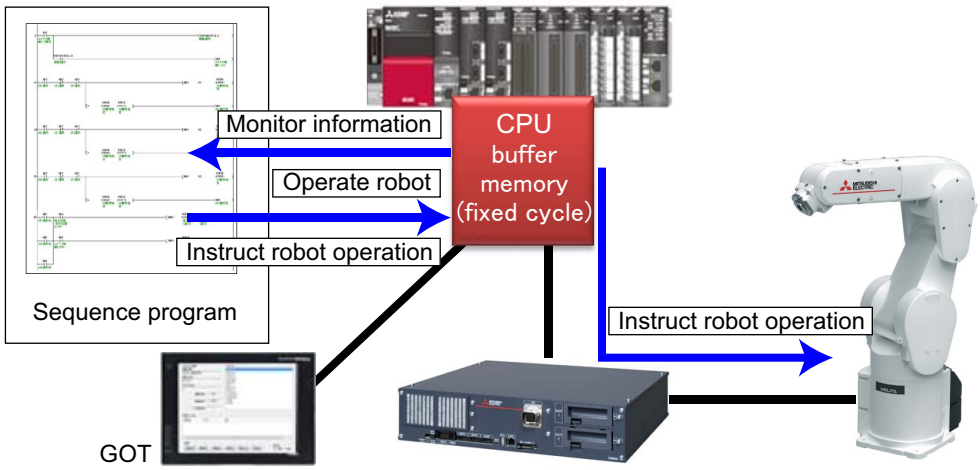
1 Overview

These specifications describe the functions (sequencer direct performances) which extend the CPU buffer memory in CR800-R series robot controller, exchange various robot information between sequencer and robot through the extended CPU buffer memory, and operate the robot without a robot program.

Note: These CPU buffer memory extended functions only support MELFA-BASIC V and VI or later. They do not support MELFA-BASIC Ⅳ.

(For more information, refer to Page 13, "2.1.4 Check Robot Language Setting")

Sequencer direct performance does not support mecha 2 and 3 for multiple mecha. It supports additional axis.



1.1 Function List

These CPU buffer memory extended functions are largely classified into monitoring and operation functions. Monitoring function periodically updates and outputs the data in CPU buffer memory on the robot. Operation function outputs a request from the sequencer to the robot as needed and exchanges the data. CPU buffer memory extended functions also provide a direct performance function to directly operate the robot.

No	Item		Description	I/F btwn Robots	Update Cycle
1	Monitoring function	Monitor operation control setting values	Monitors the setting values relating to operation control command and operation control.	Motoring output (Robot side periodically updates the data in CPU buffer memory)	3.5ms(CR800-R) 7.1ms(CR800-Q)
2		Monitor activities	Monitors the robot's activities (current speed, arrival factor to the aimed position, etc.)		3.5ms(CR800-R) 7.1ms(CR800-Q)
3		Monitor current and aimed positions	Monitors current and aimed positions of robot.		3.5ms(CR800-R) 7.1ms(CR800-Q)
4		Monitor general position and joint information	Monitors various position type data (orientation at collision, etc.) and joint type data (current value, load factor, etc.)		It may differ according to each item. Refer to Page 37, "3.2.4 Monitor Position and Joint Information".
5		Monitor maintenance information	Monitors the maintenance information (battery and grease remaining times).		Depending on the parameter MFINTVL
6	Operation function	Read/write variables	Reads/ writes variables used in the robot's program.	Request reply method (The robot side answers by the output request of the sequencer, and delivers the data on the CPU buffer memory)	Responds within 1s (It may vary according to the load status of robot control)
7		Read program's current line	Reads currently performing line of the robot program on a per line basis (up to 128 characters).		
8		Set up maintenance	Resets the servomotor information.		
9		Read error information	Reads detailed error information (program name, occurred line, etc.)		
10		Read product information	Reads the robot's product information (model name, version, and serial number).		

No	Item		Description	I/F btwn Robots	Update Cycle
11		Perform sequencer direct	Operates the robot from the sequencer through CPU buffer memory		

1.2 Features

(1) Fulfilling functions to monitor and operate robot from GOT. Advances T/B and PC-less solution.
→ Various functions can be performed by reading/ writing the data in CPU buffer memory from GOT.

- Allows you to check activities, position information, and setting values of operation control command and thereby analyze the operation in case of debugging or problem. (Monitoring current and aimed positions, activities, and operation control setting values)
- Allows you to read and write the contents of program and variables and thereby change the robot's operation in case of debugging or problem.
- Allows you to check and set up maintenance status.
- Allows you to check error's detailed content. (Reading error information)
- Allows you to display and check various information in the robot (product, servo information, etc.)

(2) Controls peripheral devices and system according to the robot activities with the sequencer
The sequencer allows you to monitor the data in CPU buffer memory and responsively control the peripheral device connected to the sequencer according to the monitored value.

- Allows you to control the peripheral devices by monitoring the robot's activities (current speed, arrival factor to the aimed position, etc.)
- Allows you to generate an alarm to the system and report to the upper side by monitoring the maintenance and servomotor information (load factor, etc.)

(3) Analyzes the data and performs the quality control by logging the robot information through sequencer
Allows you to analyze the system data and perform the products' quality control by sending the logged robot information in CPU buffer memory to the sequencer and upper device connected to the sequencer.

- Allows you to control the system's operating situation by logging error information.
- Allows you to perform the quality control of product assembly by logging servo monitor information (current value, etc.)

(4) Allows to operate the robot without learning robot language (sequencer direct performance)

- Allows to operate the robot without knowing robot language.
  Allows you to operate the robot by writing predetermined setting value into the specified address in CPU buffer memory. Therefore, this function can be fulfilled regardless of sequencer language (ladder, ST language, SFC, etc.)
- Allows you to select either joint or linear interpolation. Also, allows you to adequately specify the robot operations such as override, acceleration and deceleration, tool setting.

Command		Action
Operation control	Mov	Move for joint interpolation
	Mvs	Move for linear interpolation
	Ovrd	Specify the overall speed
	Spd	Specify the linear interpolation speed
	Accel	Specify the acceleration and deceleration speed
Definition command	Tool	Specify the tool data
Hand command	Hopen/Hclose	Open/close a hand

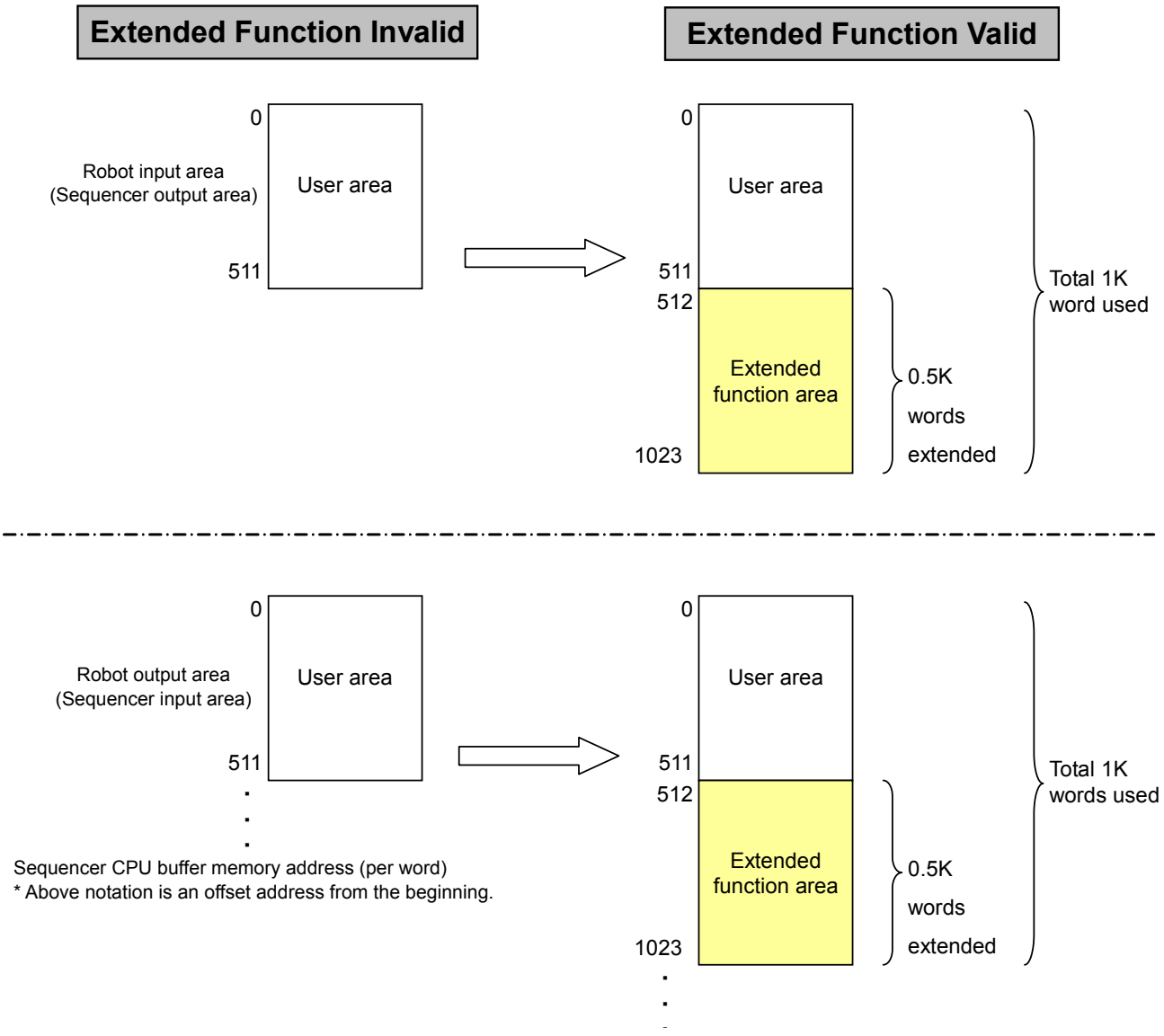
- Allows you to operate the robot with a sense, which is familiar to the sequencer programmer, to move a positioning unit.
- Allows you to control the system operations only with sequencer.
  Makes the program management easy so that a sequencer programmer can support for the change of system specification and the problem.
- Allows you to control the system settings only with the sequencer in the GOT screen.
  A sequencer programmer can support for the change of system specification and the problem so that the program management gets so easy.

1.3 CPU buffer Memory Configuration

Here, describes the CPU buffer memory configuration among multiple CPUs.

1.3.1 Memory Configuration for Valid/Invalid Extended Function

To use the CPU buffer memory extended functions, enable the CPU buffer memory extended functions with the parameter "IQMEM".
After enabling the CPU buffer memory extended functions, the CPU buffer memory is used by extending the robot I/O area by 0.5 K word.



Note) Only the user area can be referred to by robot program, signal monitor, and dedicated I/O signal allocation. They cannot refer to the extended function area.

1.3.2 Memory Map of Extended Function Area

The table below lists the memory map of extended function area in the CPU buffer memory among multiple CPUs.

* As the sequencer address may differ according to each CPU device, the sequencer address is described in the offset address from start address.
* When not otherwise specified, the values are stored in binary format.

(1) Robot input (sequencer output) area

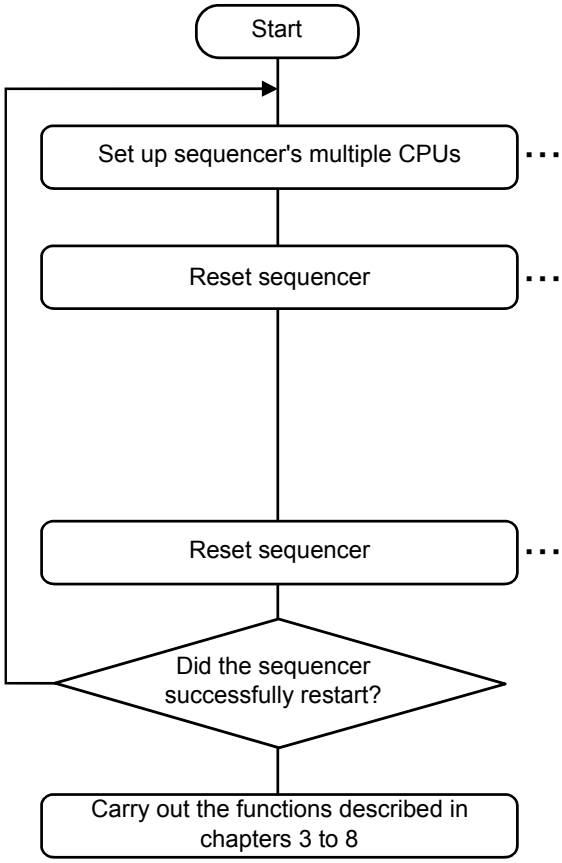
CPU buffer Memory Addr / Sequencer Addr	Description
512	Common setting area of extended function
	Sequencer direct performance area
600	
700	Common area of operation function
	Reading/ writing/ teaching area of variables
	Reading area of program's current line
800	
	Reset area of servo monitor information
	Reading area of error and product information
	Common area of monitoring function
	Monitoring area of general position and joint information
	(Reserved: Future extended area)
900	
1000	
1023	
1024	

(2) Robot output (sequencer input) area

CPU buffer Memory Addr / Sequencer Addr	Description
512	Common setting area of extended function
	Sequencer direct performance area
	Common area of operation function
	Read/write variables
	Reading area of program's current line
600	
	Reset area of servo monitor information
	Reading area of information
700	
	Common area of monitoring function
	Monitoring area of operation control setting values
800	
	Monitoring area of activities
	Monitoring area of current and aimed positions
900	
	Monitoring area of general position and joint information
	Monitoring area of maintenance information
1000	
	(Reserved)
1023	
1024	

2 Preparation for Using Extended Function

2.1 Operation flow



- Sets up sequencer's parameter (multiple CPU setting).
  For robot CPU, allocate 1K word of free user area in high-speed communication area of multiple CPUs to robot's I/O area.
  (Refer to Page 7, "2.1.1 Set up Sequencer's Multiple CPUs")

- Sets up multiple CPUs' parameters.
  (Refer to Page 11, "2.1.2 Set up Robot's Multiple CPUs")
- Sets up parameter IQMEM.
  1) To enable the extended function, set bit 0 to one.
  2) To enable the sequencer direct performance, set bit 1 to one.
  (Refer to Page 12, "2.1.3 Set up Parameter for Selecting CPU buffer Memory Extended Function")
- Set robot language to "MELFA BASIC V" or "MELFA BASIC VI."
  (Refer to Page 13, "2.1.4 Check Robot Language Setting")

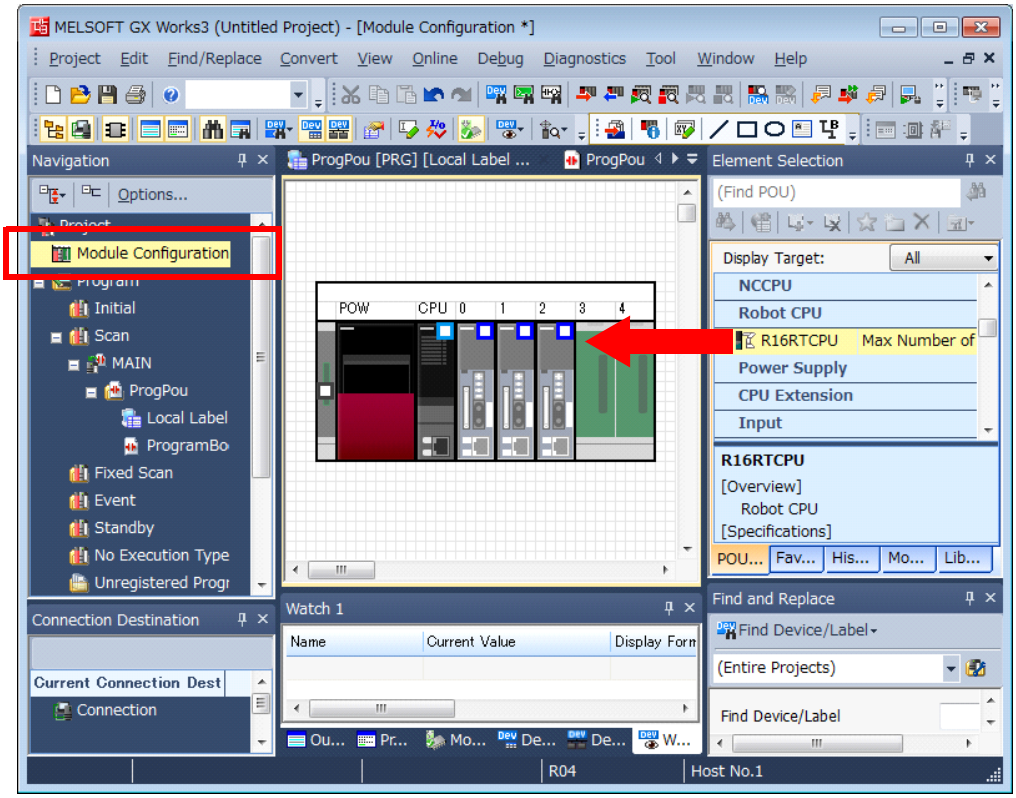
- Carry out the reset operation of sequencer or reset the sequencer by turning ON the power.

"3Monitor Robot Information"
"4Reads/Writes Robot's Variables"
"5Read Current Line of Robot Program"
"6Set up Robot's Maintenance"
"7Read Robot Information"
"8Perform Sequencer Direct"

2.1.1 Set up Sequencer's Multiple CPUs

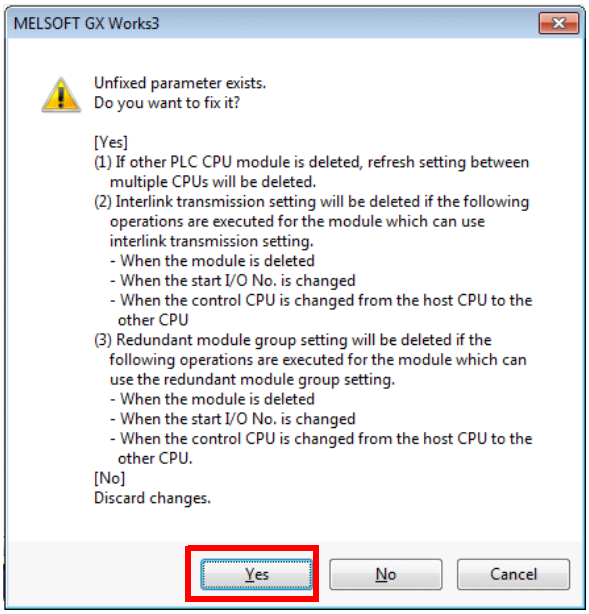
(1) CR800-R series

Set up a module configuration before set up sequencer's multiple CPUs. Based on the actual module configuration, arrange the modules as shown below.

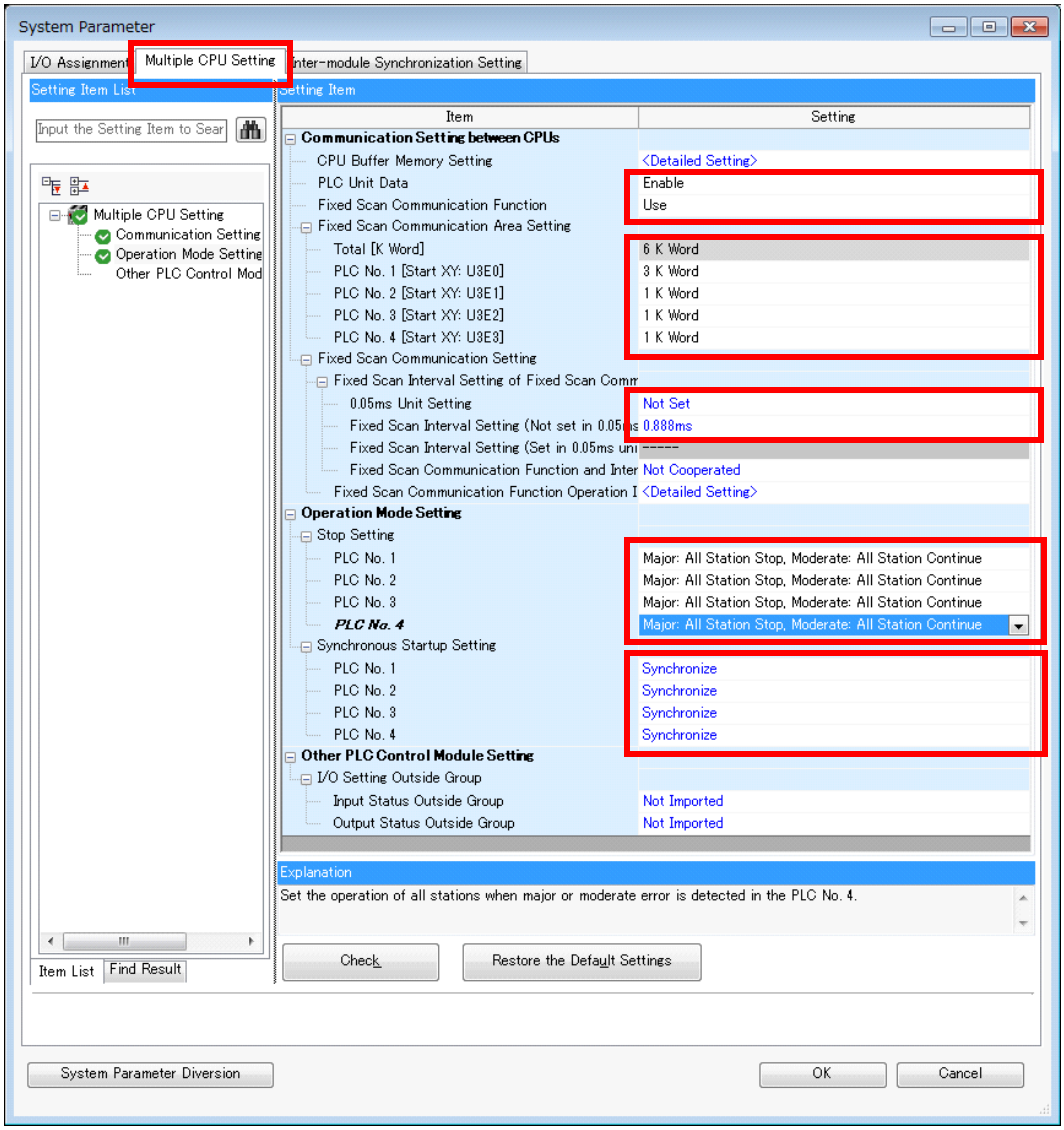


GX Works3 module configuration screen (three robots)

When the module configuration screen is closed after arranging the modules, the following dialogue will appear. Click [Yes] button.



Open the system parameter screen to set up the multiple CPUs.



Setting Item		Description	Setting Value
Communication Setting between CPUs	PLC Unit Data	Set up this to avoid data separation at data communications by refresh. When data of more than 64 bit data is handled, data croaking may occur.	Enable Disable The update cycle of the Sequencer link I/O function of the robot CPU is as follows. < PLC Unit Data - Disable > It will be the specified periodic communication interval. < PLC Unit Data - Enable > Even when setting the periodic communication interval setting to "0.444 ms" or less, it will be 0.888 ms. * Other data update cycle is as described in each chapter.
	Fixed Scan Communication Function	Select whether the fixed scan communication function is used or not. Always select "Use" when using together with a robot CPU.	Use
	Fixed Scan Communication Area Setting	Set up the range of transmission area of each device in the fixed scan communication area. The necessary area Note1) for robot is as follows: **CPU buffer memory extended functions are valid:** • **Robot input area: 1.0K** • **Robot output area: 1.0K** **CPU buffer memory extended functions are invalid:** • **Robot input area: 0.5K** • **Robot output area: 0.5K**	<CPU buffer memory extended functions are valid:> • Device #1: Sum of the size (1K) of the data to be sent to the robot and the size of the data to be sent to other devices • Robot device: Set 1K for it • Other devices: Set its own transmission size <CPU buffer memory extended functions are invalid:> • Device #1: Sum of the size (0.5K) of the data to be sent to the robot and the size of the data to be sent to other devices • Robot device: Set 1K for it Note2) • Other devices: Set its own transmission size
	0.05ms Unit Setting	Specify whether to set the fixed scan communication cycle in increments of 0.05ms. Please be sure to select "Not Set" when combining with robot CPU.	Not Set
	Fixed Scan Interval Setting (Not set in 0.05ms unit)	Select the fixed scan communication cycle to be set from the list items. The same option should be specified only for CPU modules which will use the fixed scan communication function. Please select "0.888ms" or less regardless of the PLC Unit Data. When using cooperative control function or interference avoidance function of the robot CPU.	0.222ms 0.444ms 0.888ms 1.777ms 3.555ms 7.111ms
Operation Mode Setting	Stop Setting	If a moderate or major error occurs at any of the CPUs, set whether to stop or continue operation for all CPUs.	Major: All Station Stop, Moderate: All Station Continue. * Set for all CPUs.
	Synchronous Startup Setting	Set up this to synchronize startup time of CPU modules in multiple CPUs system. * Always select "Synchronize" due to the robot CPU takes as long as 18 seconds to startup.	Synchronize

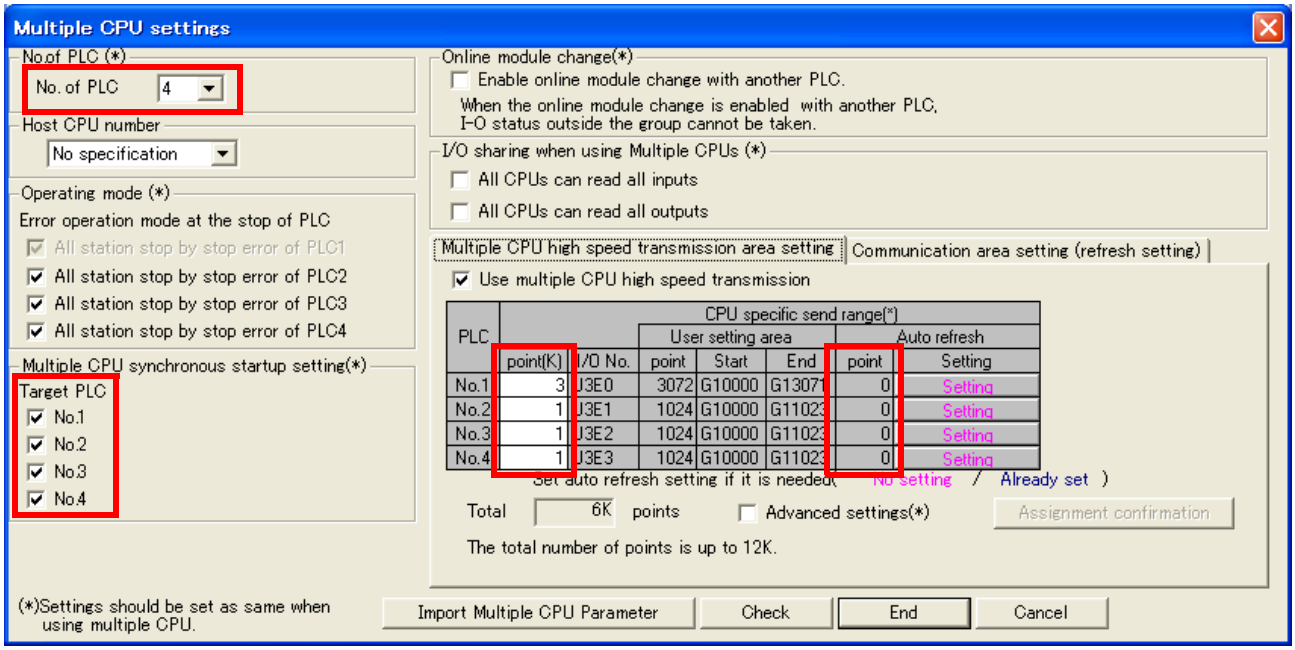
Note1) For information about multiple CPUs and fixed scan communication area, refer to the RCPU manual (MELSEC iQ-R CPU Module User's Manual (Application)).

Note2) Because the area is set up in 1K unit, allocate 1K even in case of 0.5K.

(2) CR800-Q series

Here, sets up the multiple CPU setting as a sequencer's PC parameter. Also refer to the description of sequencer link I/O functions described in Supplement volume "Instruction Manual, Detailed Description of Functions and Operations."

Below is an example of configuring multiple CPUs in GX-Developer. It is also possible to configure multiple CPUs in GX Works2 and GX Works3.



GX-Developer multiple CPU setting screen (three robots. The shared memory extended functions are valid in all robots)

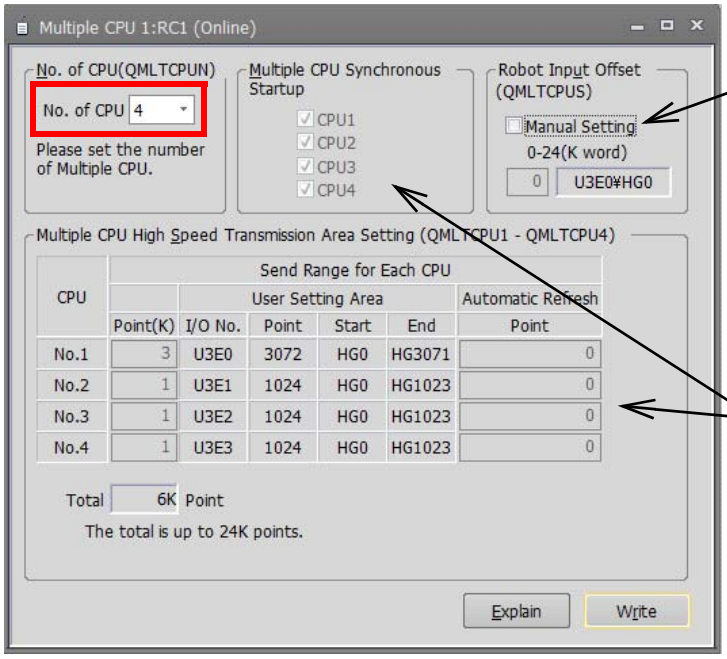
Setting Item	Description	Setting Value
CPU quantity	Set up the quantity of CPU units used in multiple CPU system.	2 - 4
Synchronous startup among multiple CPUs	Set up this to synchronize the startup times of CPU units in multiple CPU system. * Because the robot CPU takes a dozen second for startup, select the synchronize startup	Required for check
High-speed communication area setting among multiple CPUs	Set up this when the data is transferred by using the high-speed communication area Note1) among multiple CPUs. The necessary area for robot is as follows: **Shared memory extended functions are valid:** **• Robot input area: 1.0K** **• Robot output area: 1.0K** **Shared memory extended functions are invalid:** **• Robot input area: 0.5K** **• Robot output area: 0.5K**	<Shared memory extended functions are valid:> • Device #1: Sum of the size (1K) of the data to be sent to the robot and the size of the data to be sent to other devices • Robot device: Set 1K for it • Other devices: Set its own transmission size <Shared memory extended functions are invalid:> • Device #1: Sum of the size (0.5K) of the data to be sent to the robot and the size of the data to be sent to other devices • Robot device: Set 1K for it Note2) • Other devices: Set its own transmission size Note 2: Because the area is set up in 1K unit, allocate 1K even in case of 0.5K.
Automatic refresh setting	Set up this when the device data is automatically refreshed by using the high-speed communication area among multiple CPUs. * Robot CPU is not supported. Always set this to zero.	Robot device: Set zero point for it Other devices: To use automatic refresh function, set its score and target device

Note1) For information about multiple CPUs and high-speed communication area among multiple CPUs, refer to the QCPU manual (QCPU User Manual, Multiple CPU System).

Note2) Because the area is set up in 1K unit, allocate 1K even in case of 0.5K.

2.1.2 Set up Robot's Multiple CPUs

Here, sets up the multiple CPUs as a robot's parameter. In the description below, parameter setting screen of RT ToolBox3 illustrates this setting. This can also be set up by specifying the parameter name in the teaching box's parameter setting screen.



• The multiple CPU input offset parameter "QMLTCPUS" can also be set.

• In the multiple CPU Synchronous Startup area and Multiple CPU High Speed Transmission Area Setting, automatically get the value of device #1 settings.

RT ToolBox3 multiple CPU setting screen
(three robots. CPU buffer memory extended functions are valid in all robots)

[Start address of robot input offset]
The Table 2-1 lists the start addresses of robot input area in the robot's initial setting (multiple CPU input offset parameter "QMLTCPUS" is set to "-1") (The start address changes according to whether the CPU buffer memory extended functions are enabled or not).

Table 2-1:Start address of robot input area when the multiple CPU input offset parameter is initial value

Device No	CPU buffer Memory Extended Functions	
	Invalid	Valid
Device #2 (robot 1)	0K	0K
Device #3 (robot 2)	0.5K	1.0K
Device #4 (robot 3)	1.0K	2.0K

The start address of robot's input area may differ, when the valid/invalid setting for CPU buffer memory extended function may differ in other devices or when a unit other than robot is installed. In these cases, set up the multiple CPU input offset parameter (QMLTCPUS).
For setting example, refer to Page 14, "2.1.5 Allocation Example of CPU buffer Memory".
About input offset parameter (QMLTCPUS)
Sets up the offset of robot's input signals in multiple CPUs in 1K word unit.
For example, when QMLTCPUS is set to one, the start address of robot's input area is an address (U3E0\HG1024) offset by 1K word from the start address of transmission area of device #1 (sequencer).
When QMLTCPUS is set to -1 (initial value), the start address of robot's input area is as listed in the table above.

2.1.3 Set up Parameter for Selecting CPU buffer Memory Extended Function

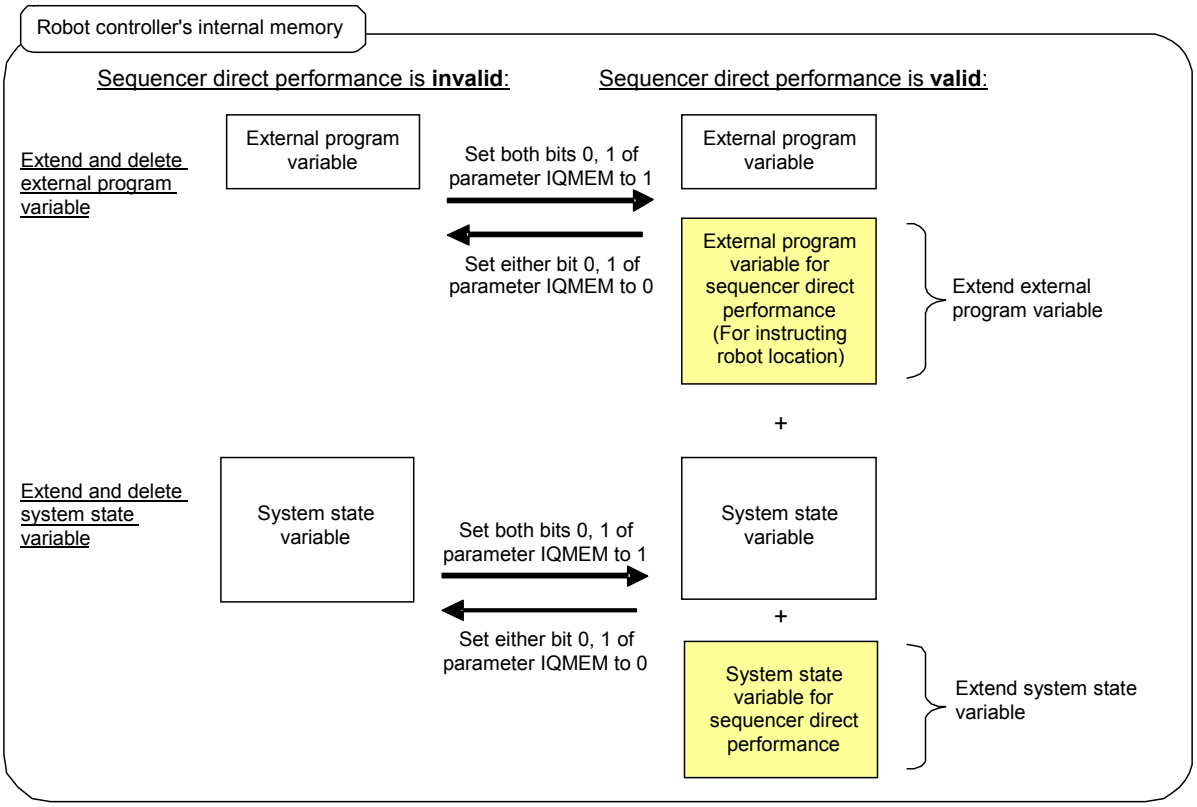
The parameter "IQMEM" for selecting the CPU buffer memory extended function is 16bit data. Set the bit 0 to one to use the extended functions (monitoring, operation functions). Set the bit 1 to one to use the sequencer direct performance function. Both bits can be set to one.
For information on how to set up a parameter, refer to Supplement volume "Instruction Manual, Detailed Description of Functions and Operations."

Parameter	Parameter Name	Array Qty Character Qty	Description	Factory Default
Select CPU buffer memory extended function	IQMEM	1 digit integer	Set validity (1)/ invalidity (0) for the function. Sets each bit by allocating a function to each bit. 0000000000000000 bit2-15: Not used |+- bit0: Use the CPU buffer memory extended function +-- bit1: Sequencer direct performance function	0000000000000000

To use the CPU buffer memory extended functions and sequencer direct performance functions, set each bit as follows:

	Bit 1	Bit 0
Use the CPU buffer memory extended function	0	1
Use the sequencer direct performance function	1	1

When using the sequencer direct performance function, robot's internal memory is extended as follows:



⚠ CAUTION When the sequencer direct performance function is valid, external program variable and system state variable areas are extended in the robot controller (extended variables). When the function gets invalid, the extended variable area is cleared. Consequently, after the sequencer direct performance function was enabled once, the robot location was taught, and the data was set, when the parameter is turned back, be aware that the previous teaching and setting data will disappear.

2.1.4 Check Robot Language Setting

The CPU buffer memory extended functions can be carried out only when the robot language is set to MELFA-BASIC V or MELFA-BASIC VI.

Check the value of robot language setting parameter "RLNG".

To use the CPU buffer memory extended function, set the parameter "RLNG" to 2 or 3.

For information on how to set up a parameter, refer to Supplement volume "Instruction Manual, Detailed Description of Functions and Operations."

Parameter	Parameter Name	Array Qty Character Qty	Description	Factory Default
Robot language	RLNG	1 digit inte-ger	Select the robot language to be used: 3: MELFA-BASIC VI (RT ToolBox3) 2: MELFA-BASIC V 1: MELFA-BASIC IV	3 (RT ToolBox3)

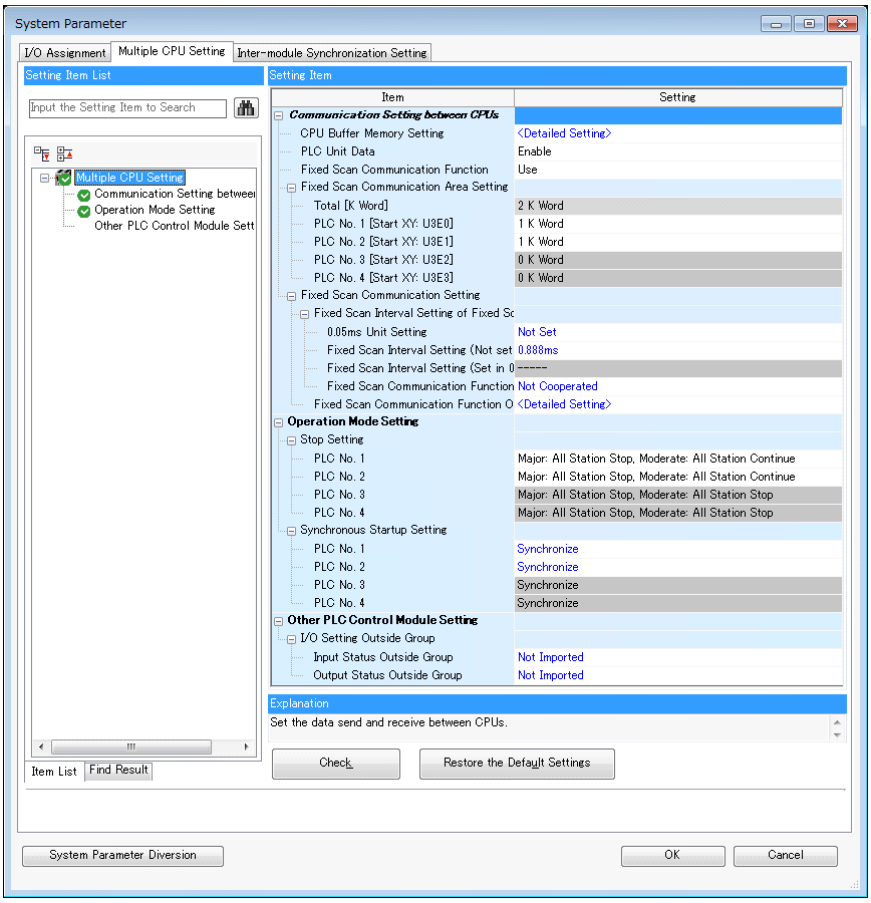
2.1.5 Allocation Example of CPU buffer Memory

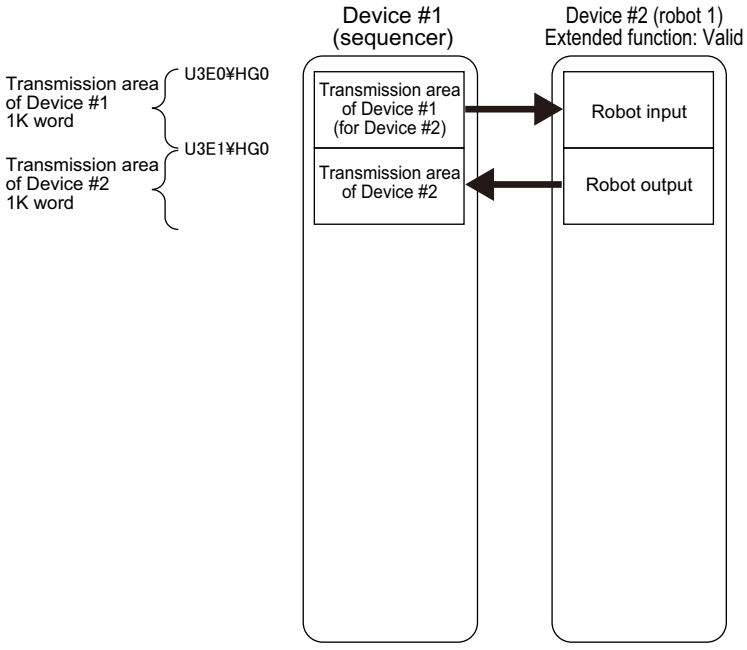
(1) Multiple CPU Configuration with One Sequencer plus One Robot

1) Case 1: Robot: Extended function is enabled, input offset parameter is initial value

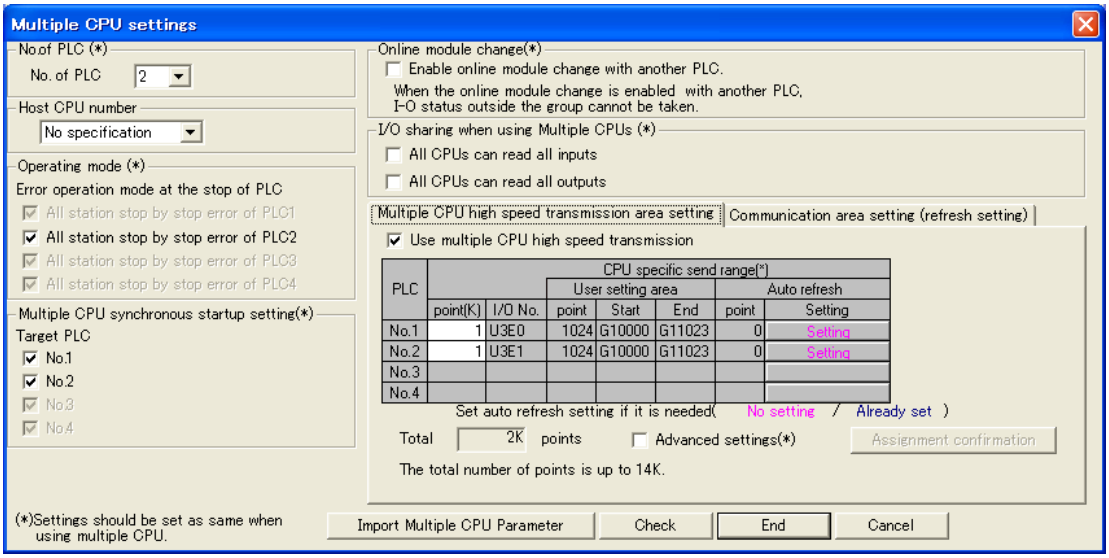
The robot uses each 1K word for I/O.

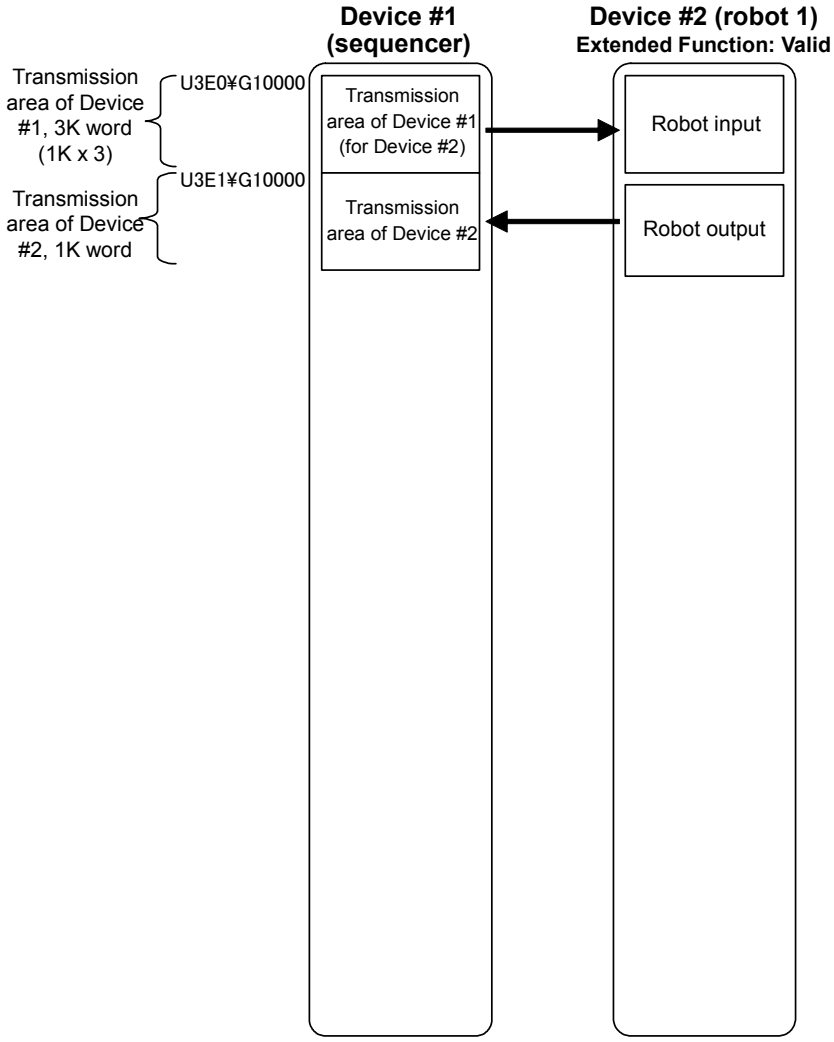
<CR800-R series>





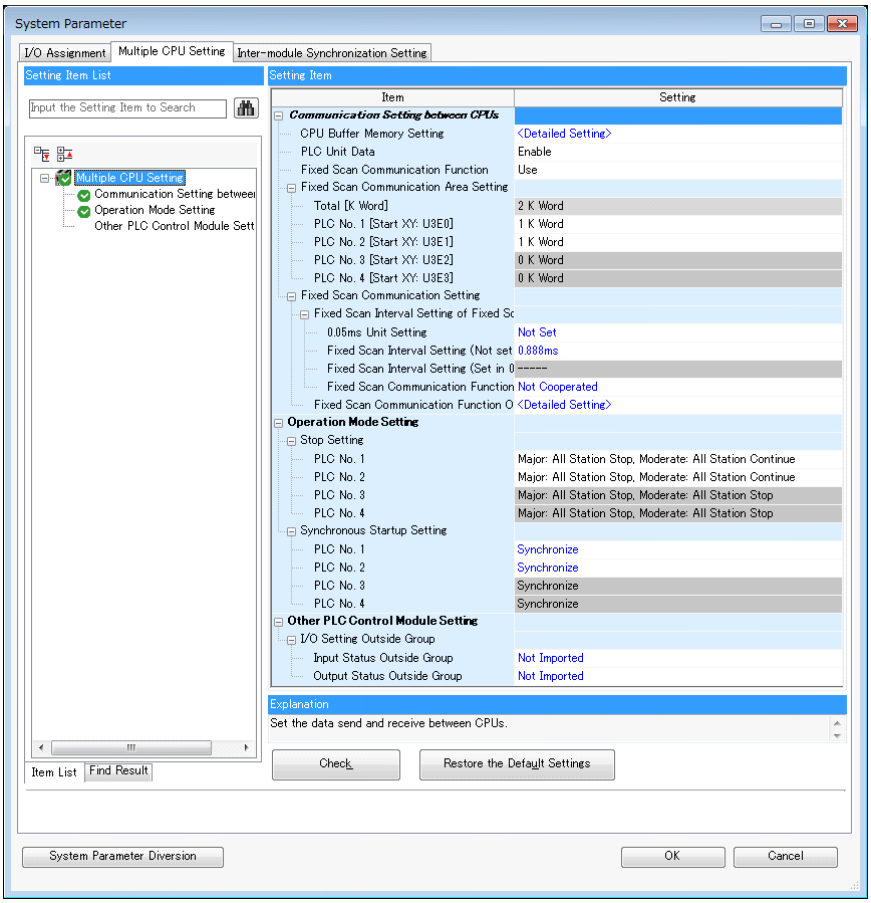
<CR800-Q series>

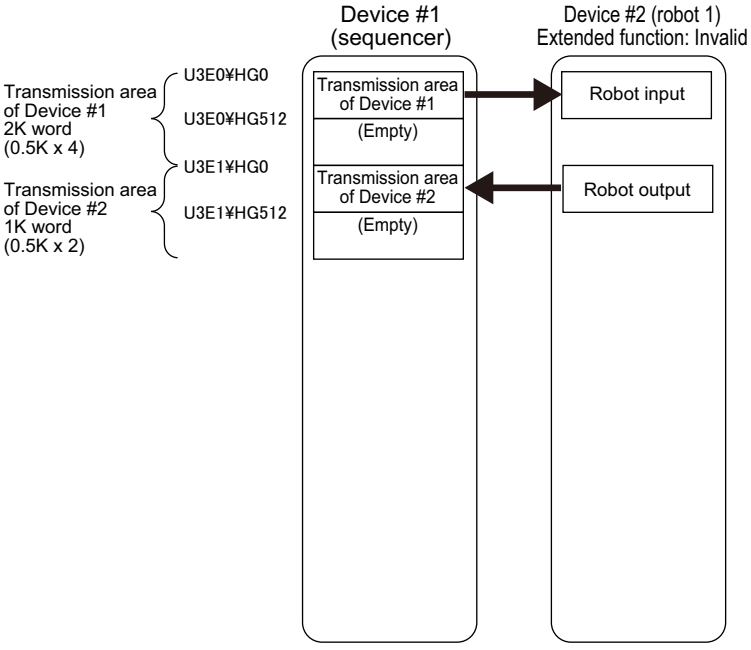




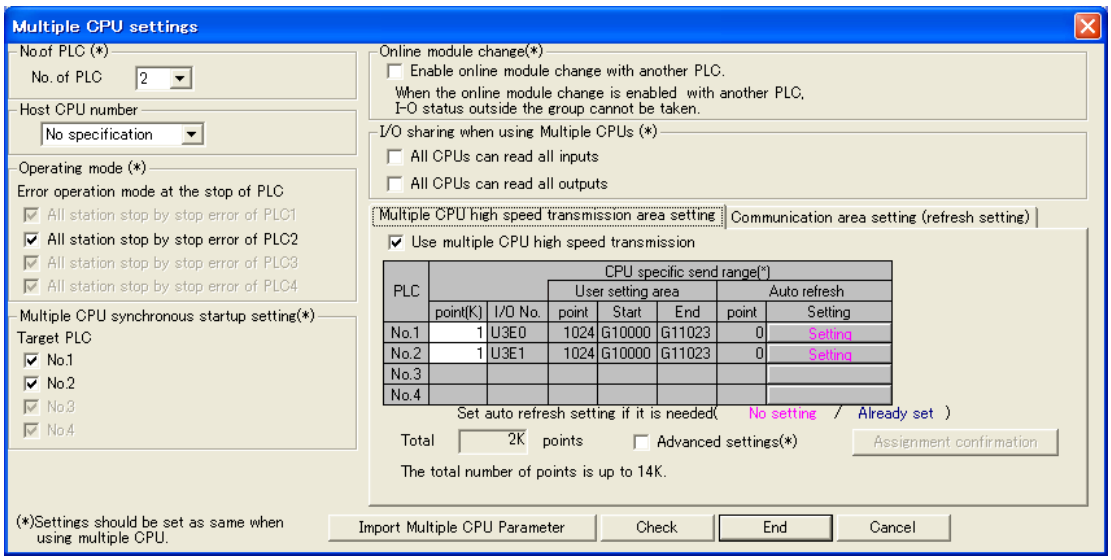
2) Case 2: Robot: Extended function is disabled, input offset parameter is initial value
The robot uses each 0.5K word for I/O.
As the transmission score is set yet in 1K word unit, the transmission score setting is as follows:

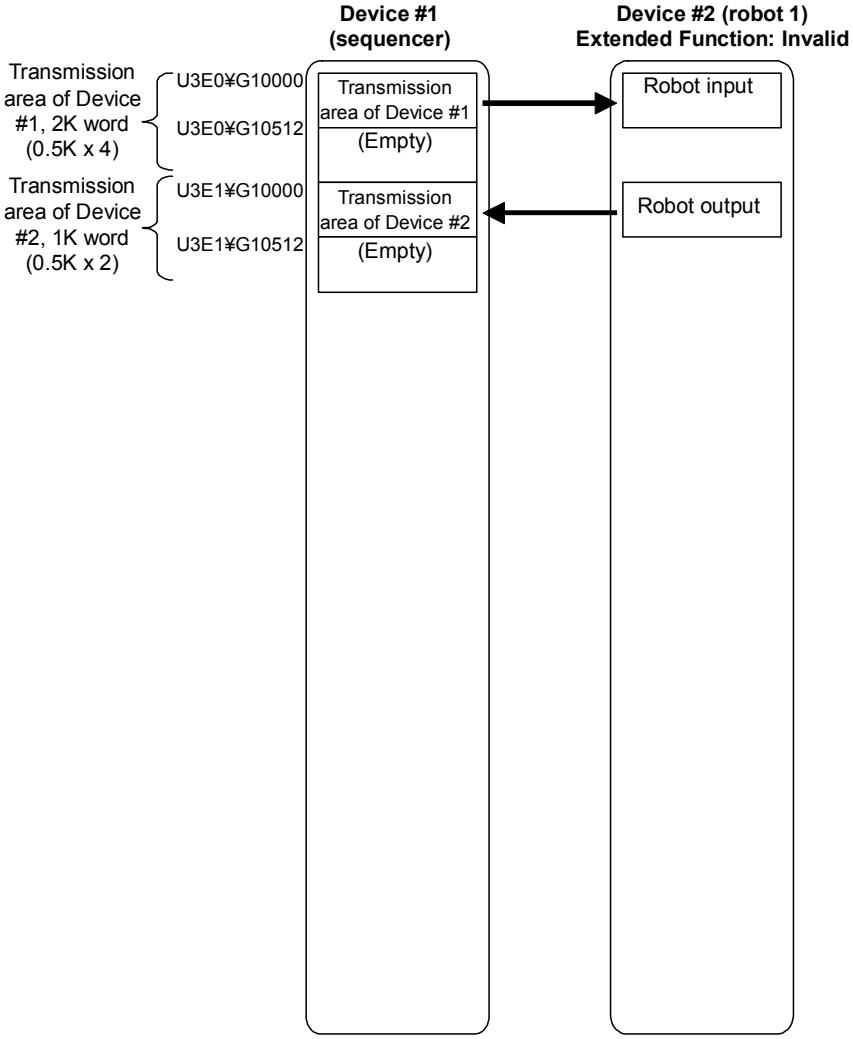
<CR800-R series>





<CR800-Q series>





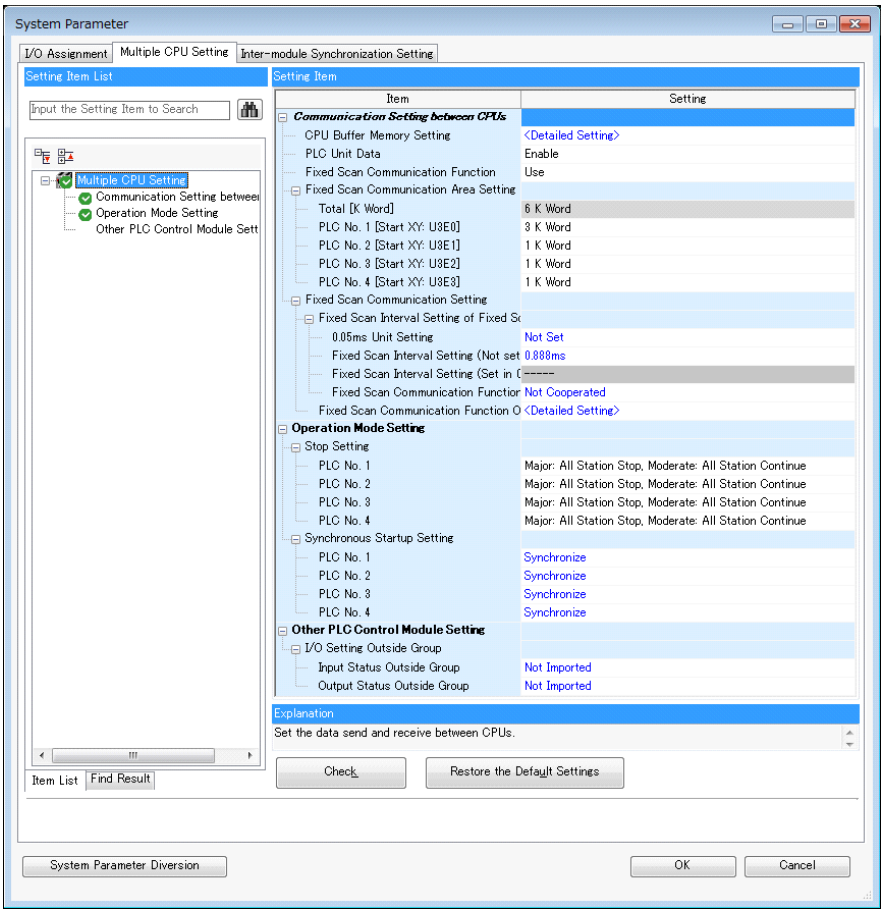
(2) Multiple CPU Configuration with One Sequencer plus Three Robots

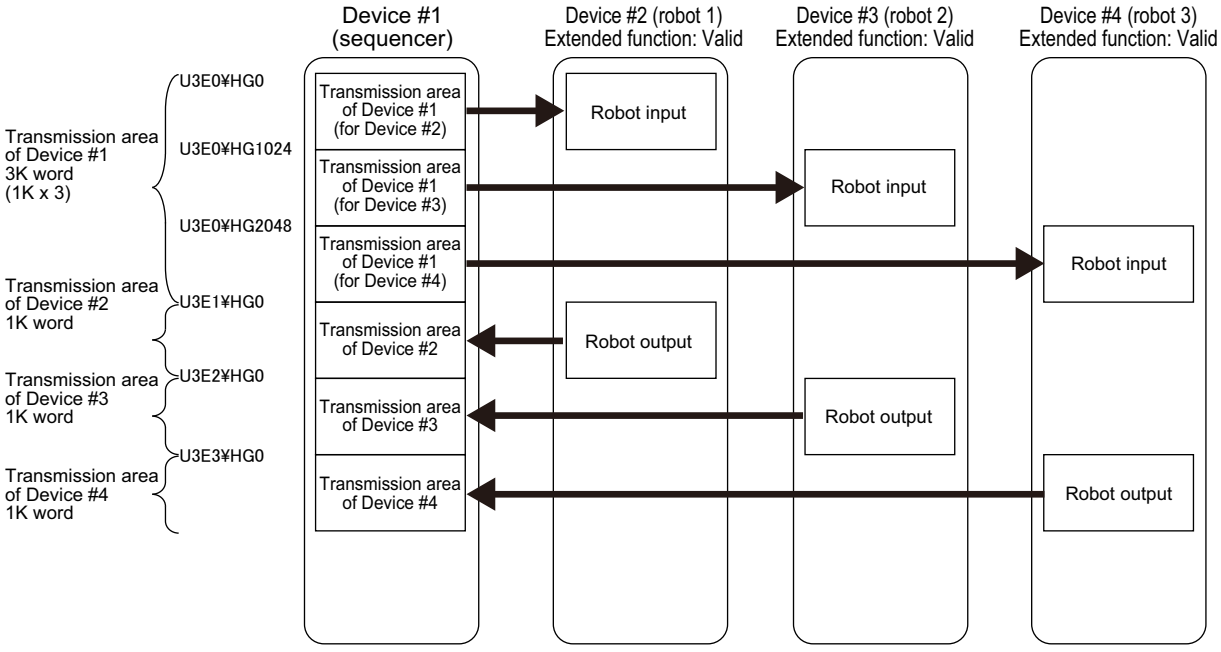
1) Case 1: All robots: Extended function is enabled, input offset parameter is initial value

All robots use each 1K word for I/O.

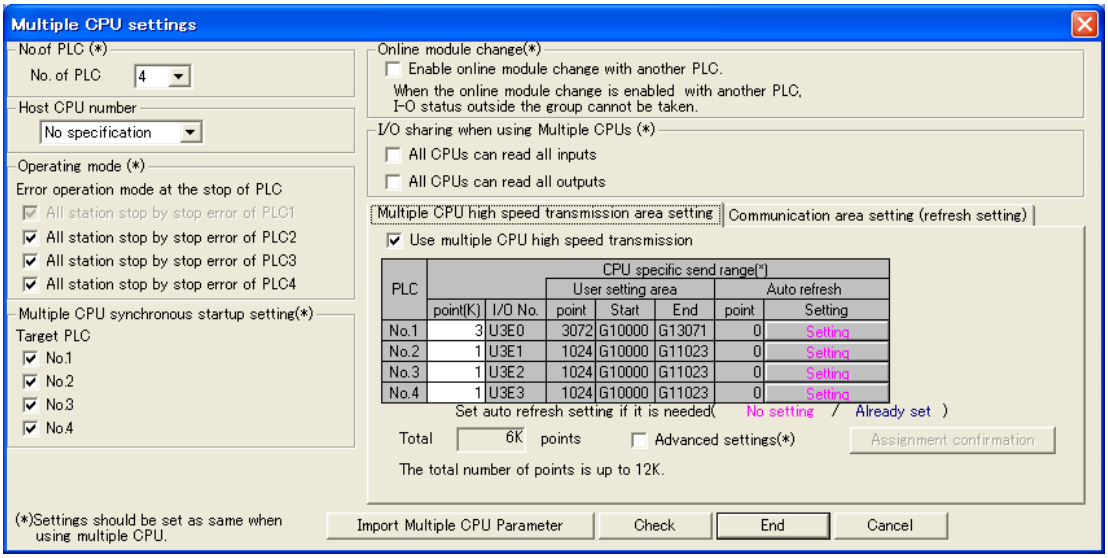
The beginning of robot 2 input area starts at 1.0K offset from the beginning of CPU buffer memory address, and the beginning of robot 3 input area starts at 2.0K offset from the beginning of CPU buffer memory address.

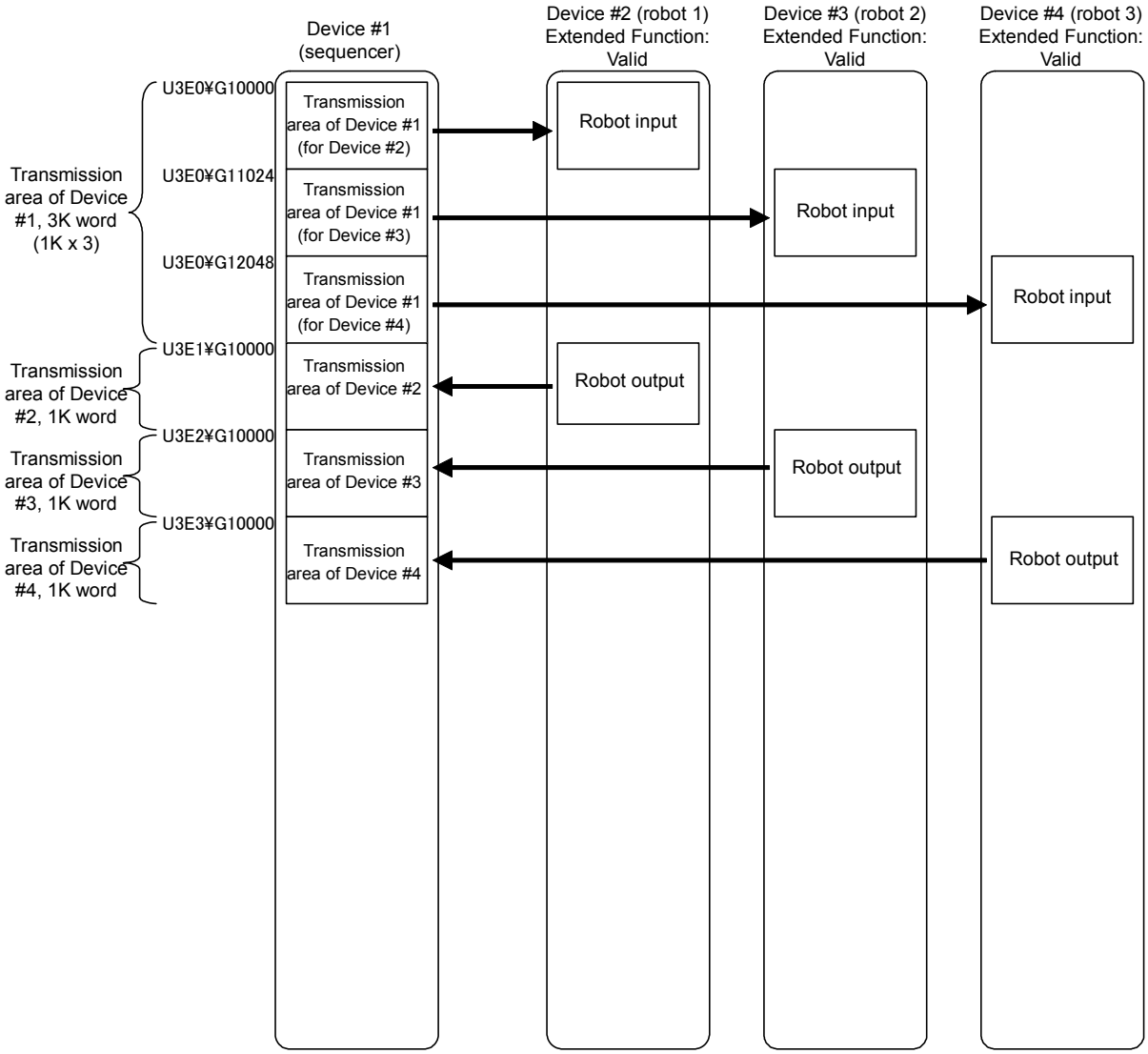
<CR800-R series>





<CR800-Q series>





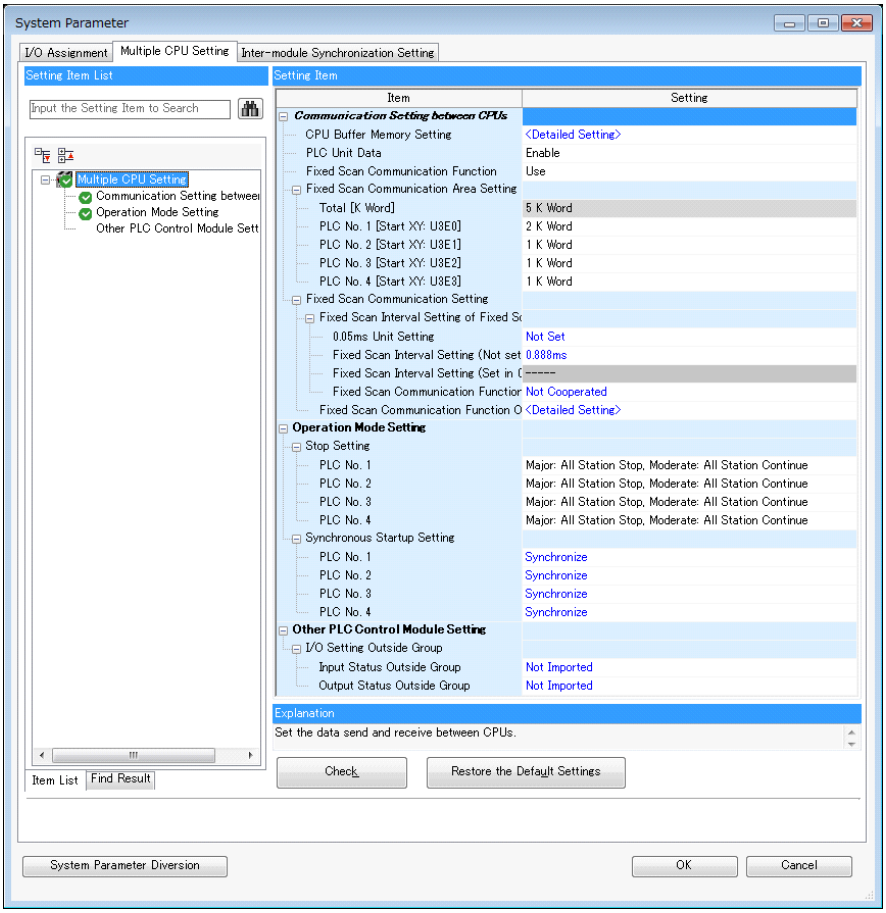
2) Case 2: All robots: Extended function is disabled, input offset parameter is initial value

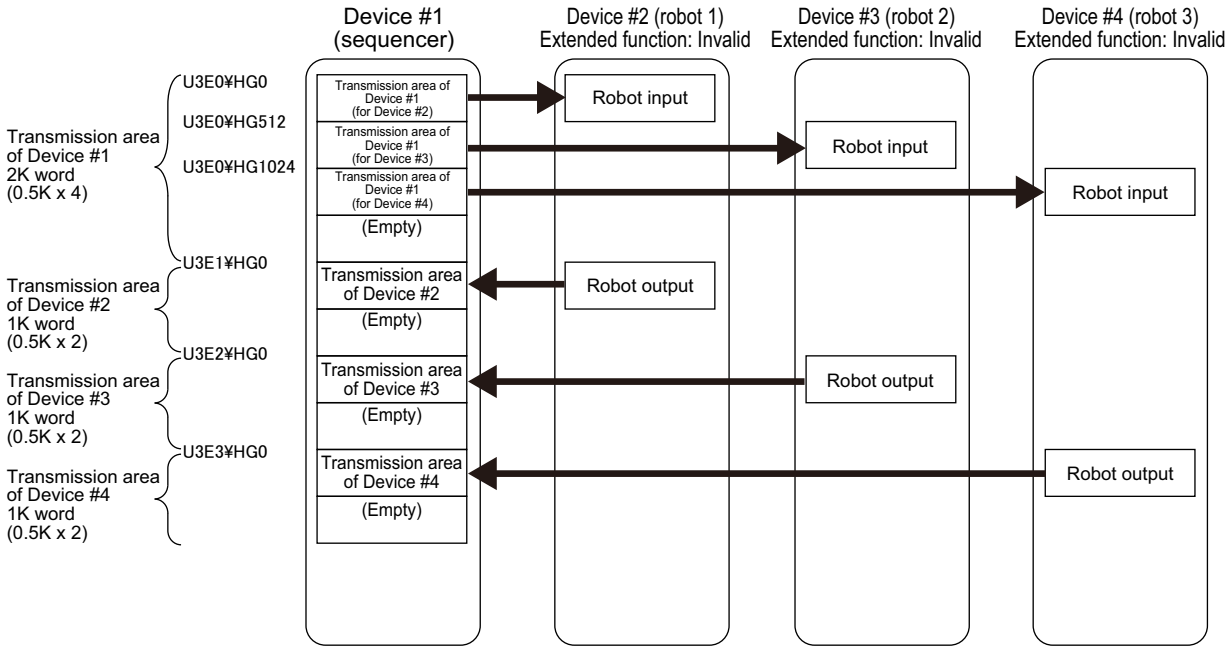
All robots use each 0.5K word for I/O.

The beginning of robot 2 input area starts at 0.5K offset from the beginning of CPU buffer memory address, and the beginning of robot 3 input area starts at 1.0K offset from the beginning of CPU buffer memory address.

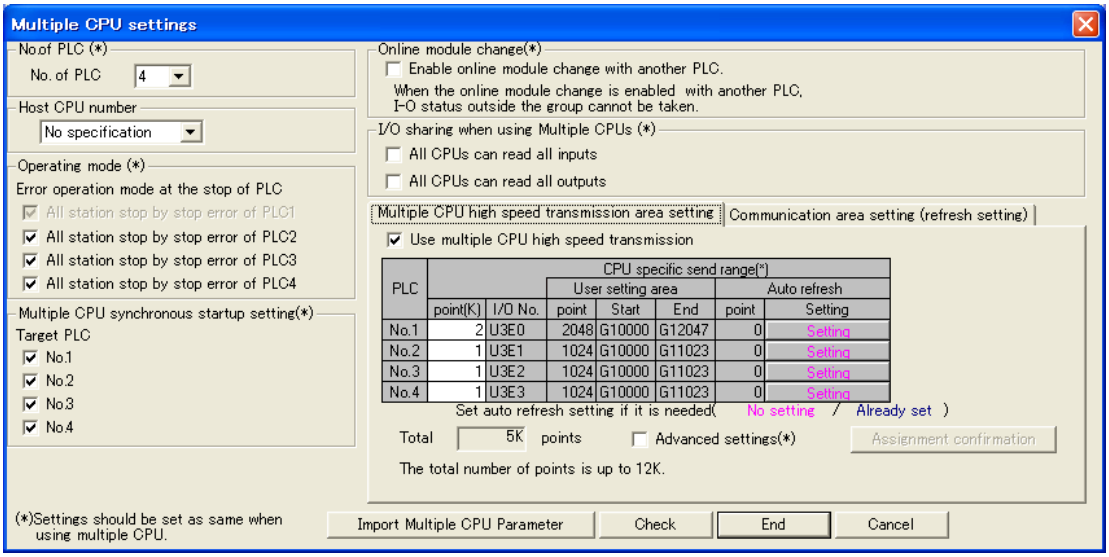
The setting is in 1K word unit as follows:

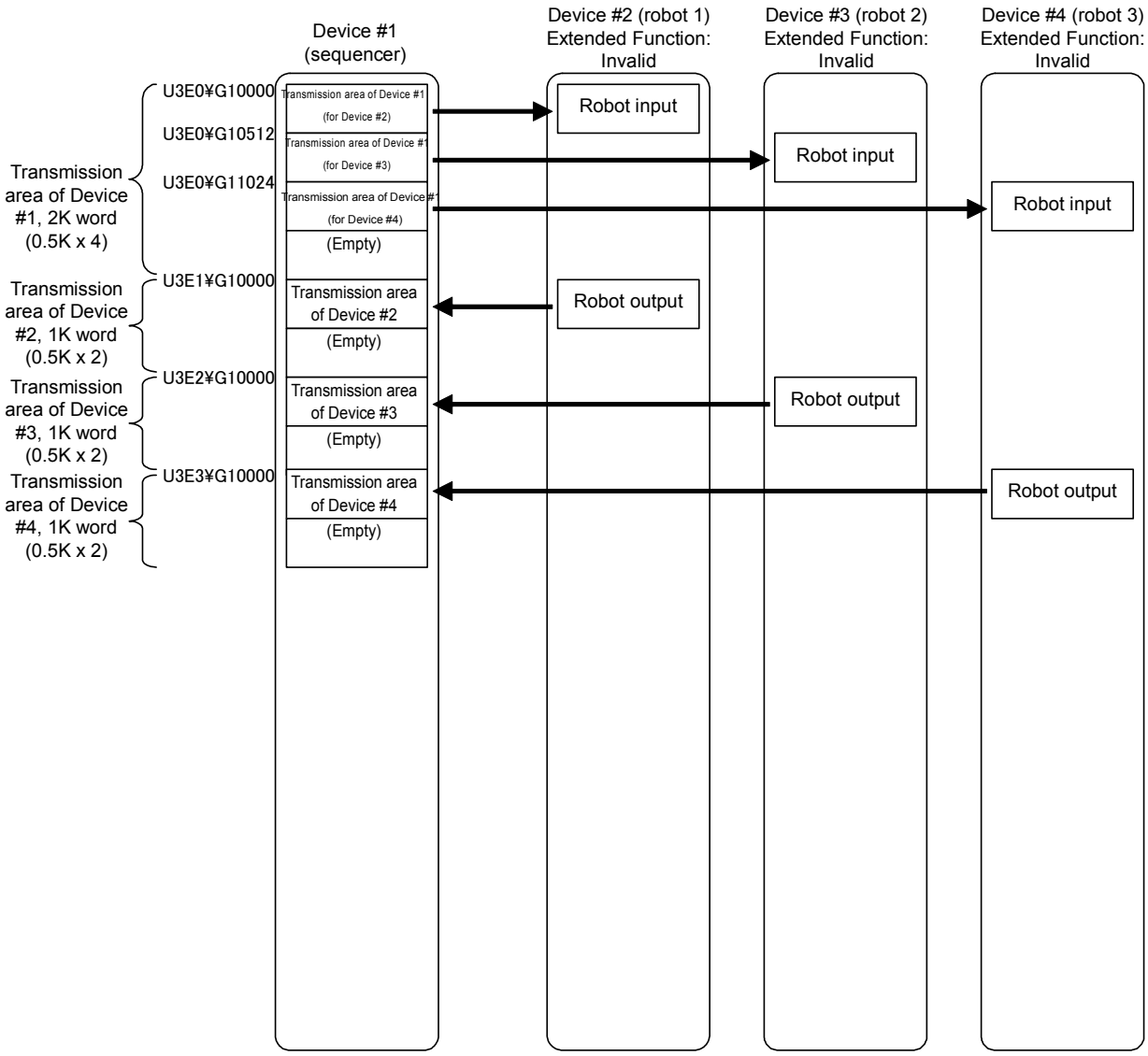
<CR800-R series>





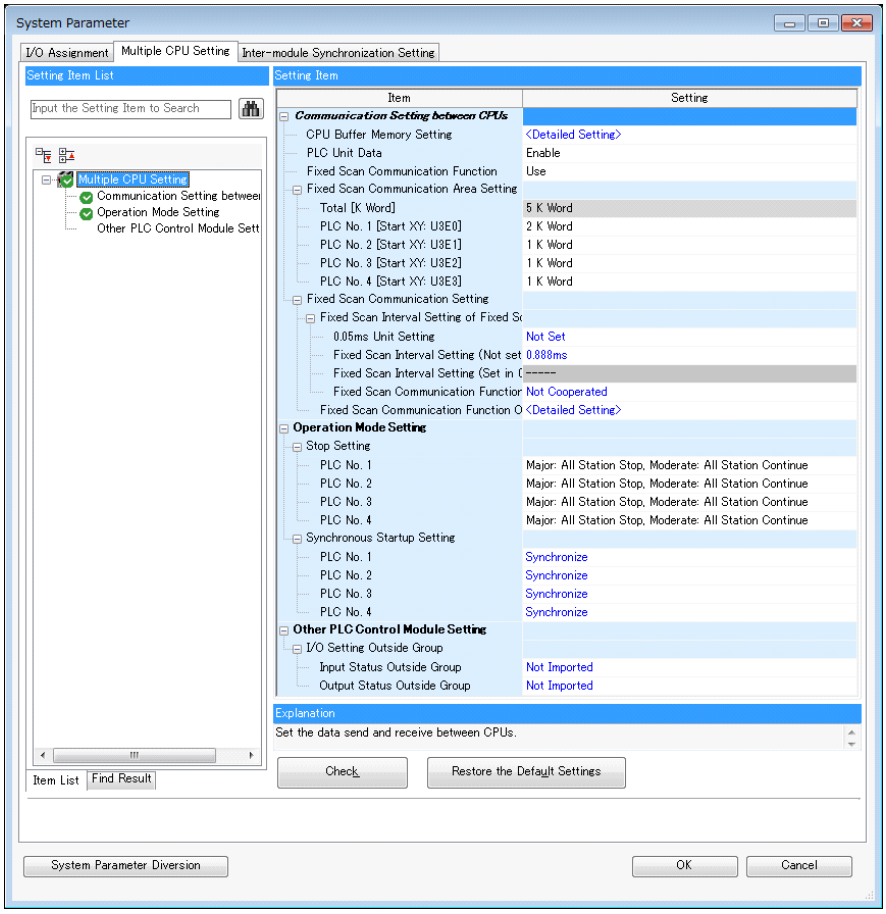
<CR800-Q series>

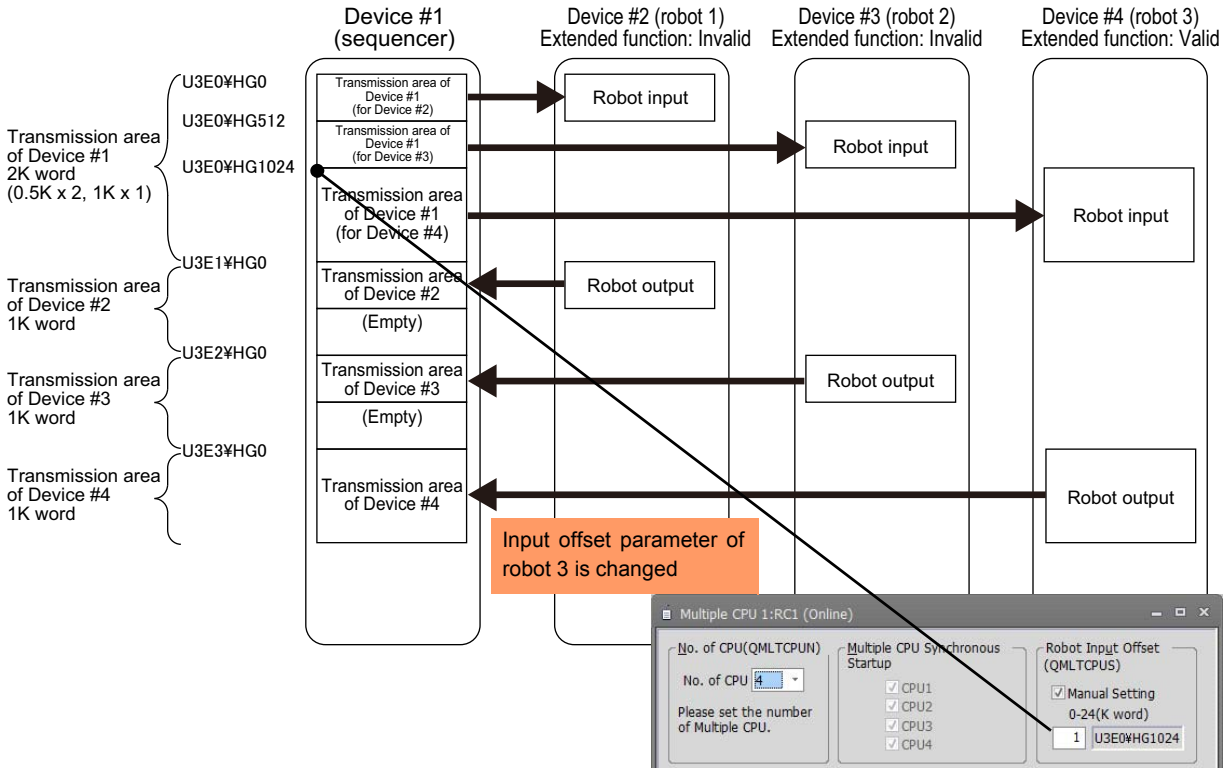




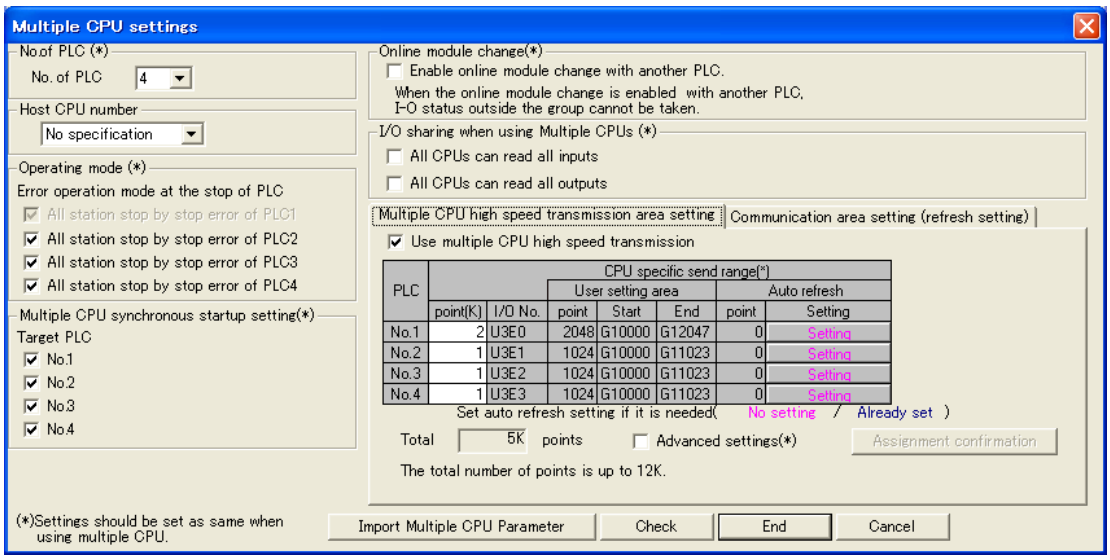
3) Case 3: Robots 1, 2: Extended function is disabled, Robot 3: Extended function is enabled (#1)
By default, the robot 3 input area starts at 2.0K offset from the beginning of CPU buffer memory (By default, the extended function of robots 1, 2 is also assumed to be enabled, similar to robot 3). Therefore, the multiple CPU input offset parameter (QMLTCPUS) of robot 3 should be set to "1."

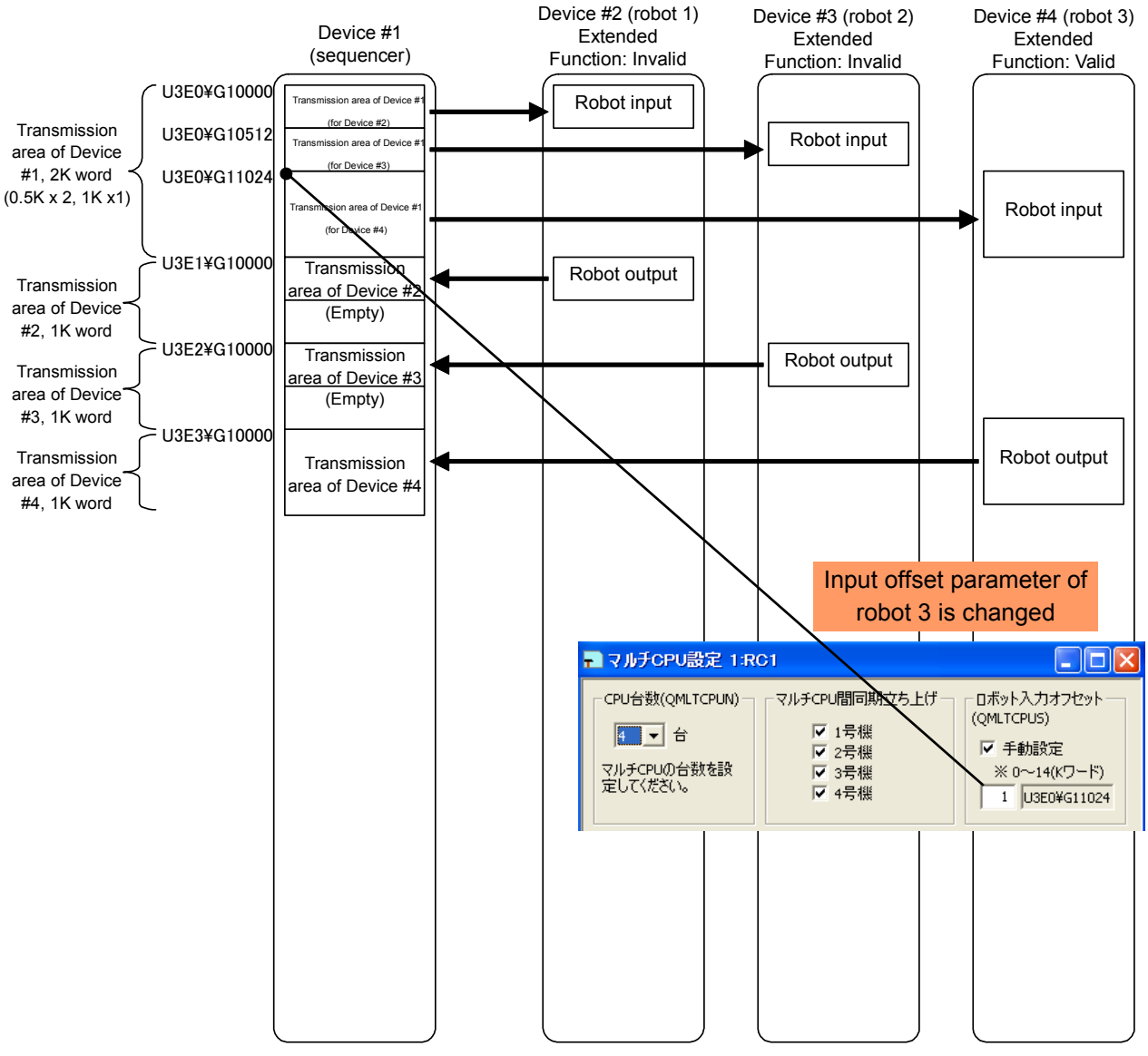
<CR800-R series>





<CR800-Q series>



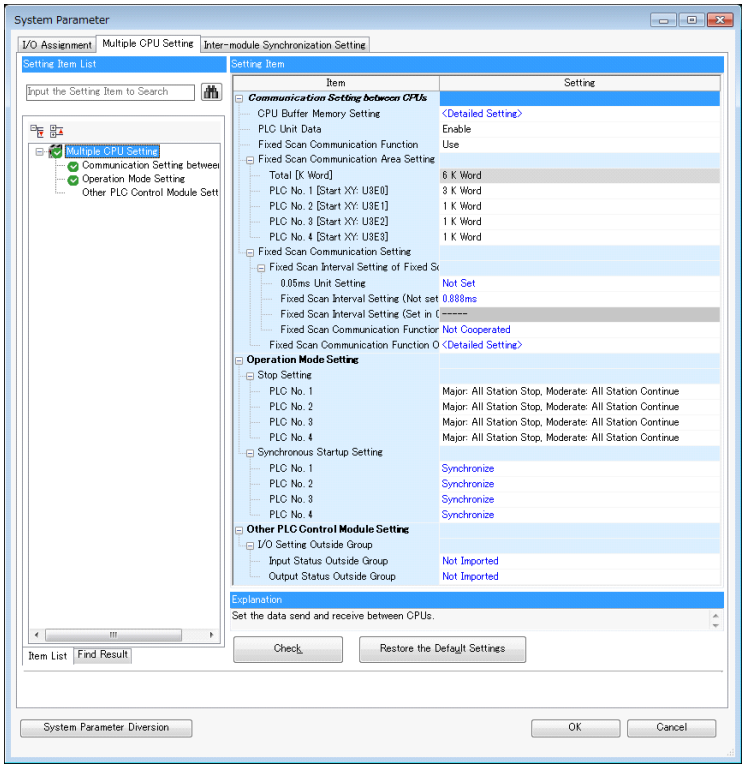


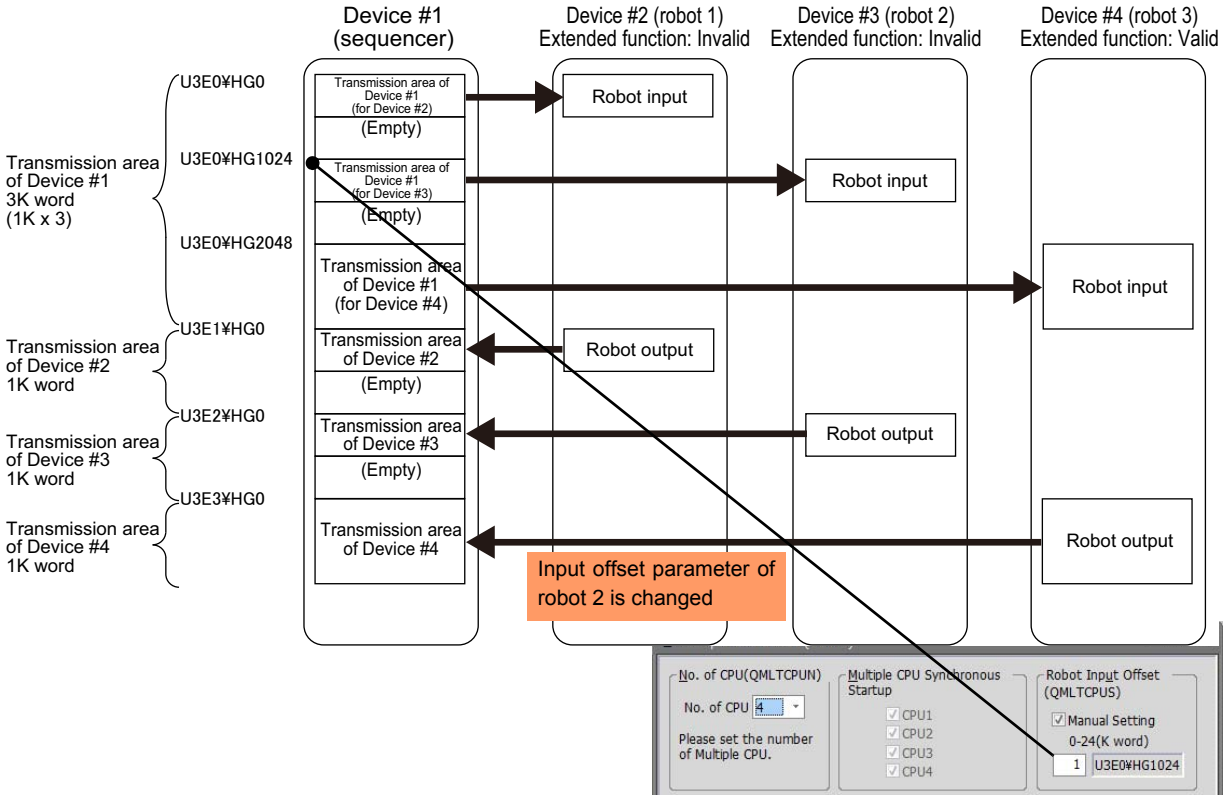
4) Case 4: Robots 1, 2: Extended function is disabled, Robot 3: Extended function is enabled (#2)

This example allocates 1K area in advance so that the allocation is not changed even when the extended function is enabled in the future, while the extended function was disabled and the extended area was not allocated.

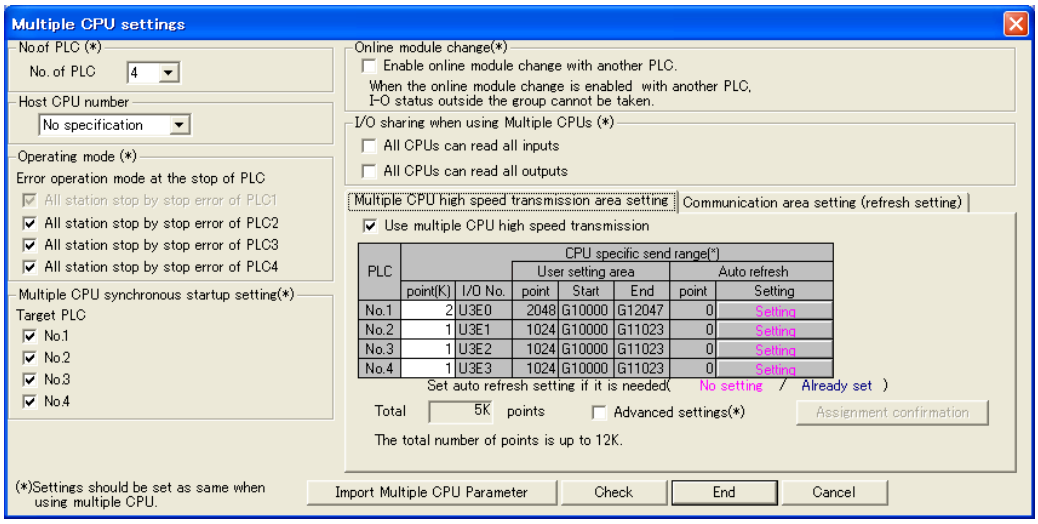
Empty area of 0.5K is kept at the back of each transmission area of robot 1 (for robots 2, 3). By default, the robot 2 input area starts at 0.5K offset from the beginning of CPU buffer memory (By default, the extended function of robots 1 is also assumed to be disabled, similar to robot 2). Therefore, the multiple CPU input offset parameter (QMLTCPUS) of robot 2 should be set to "1".

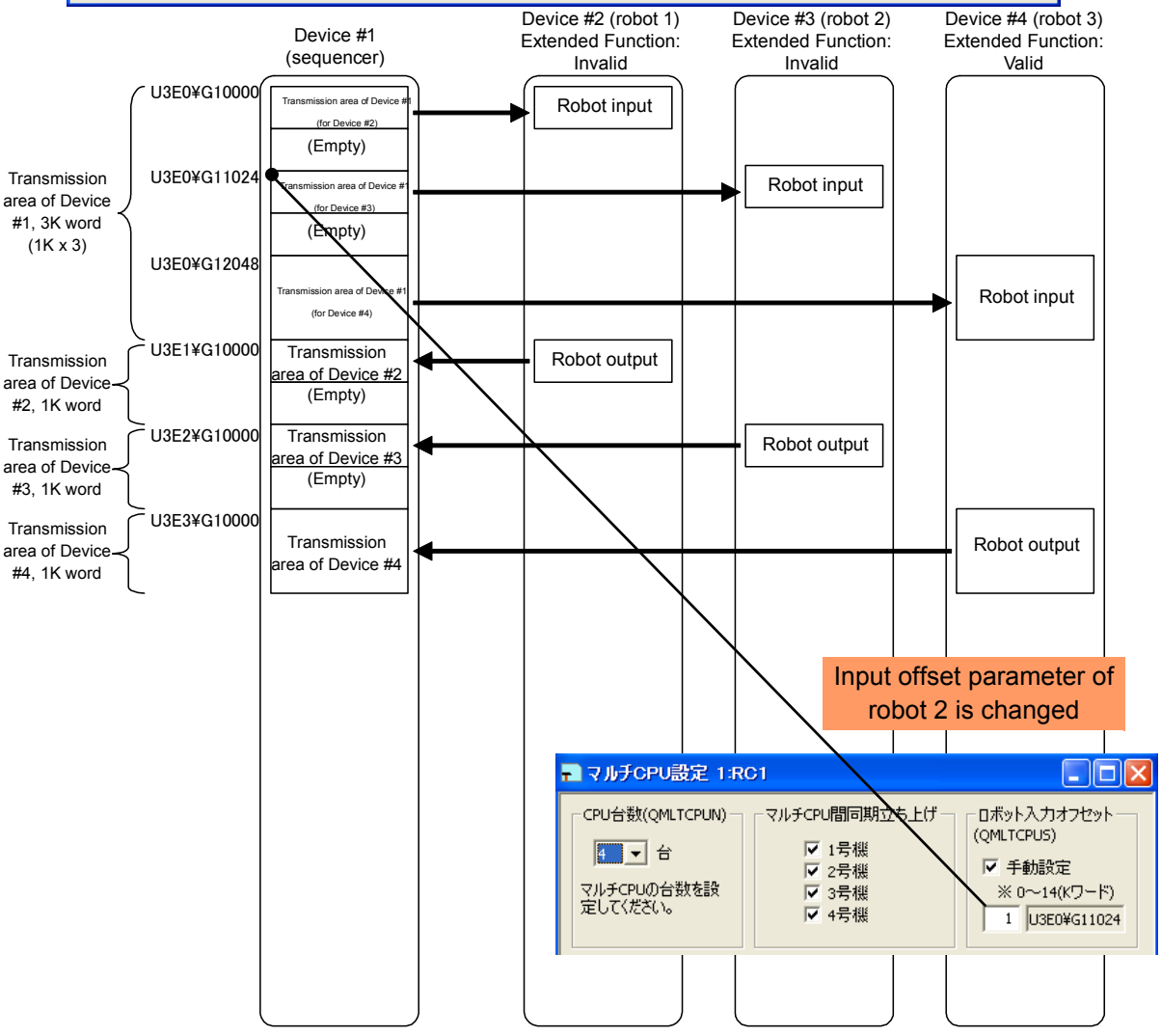
<CR800-R series>





<CR800-Q series>





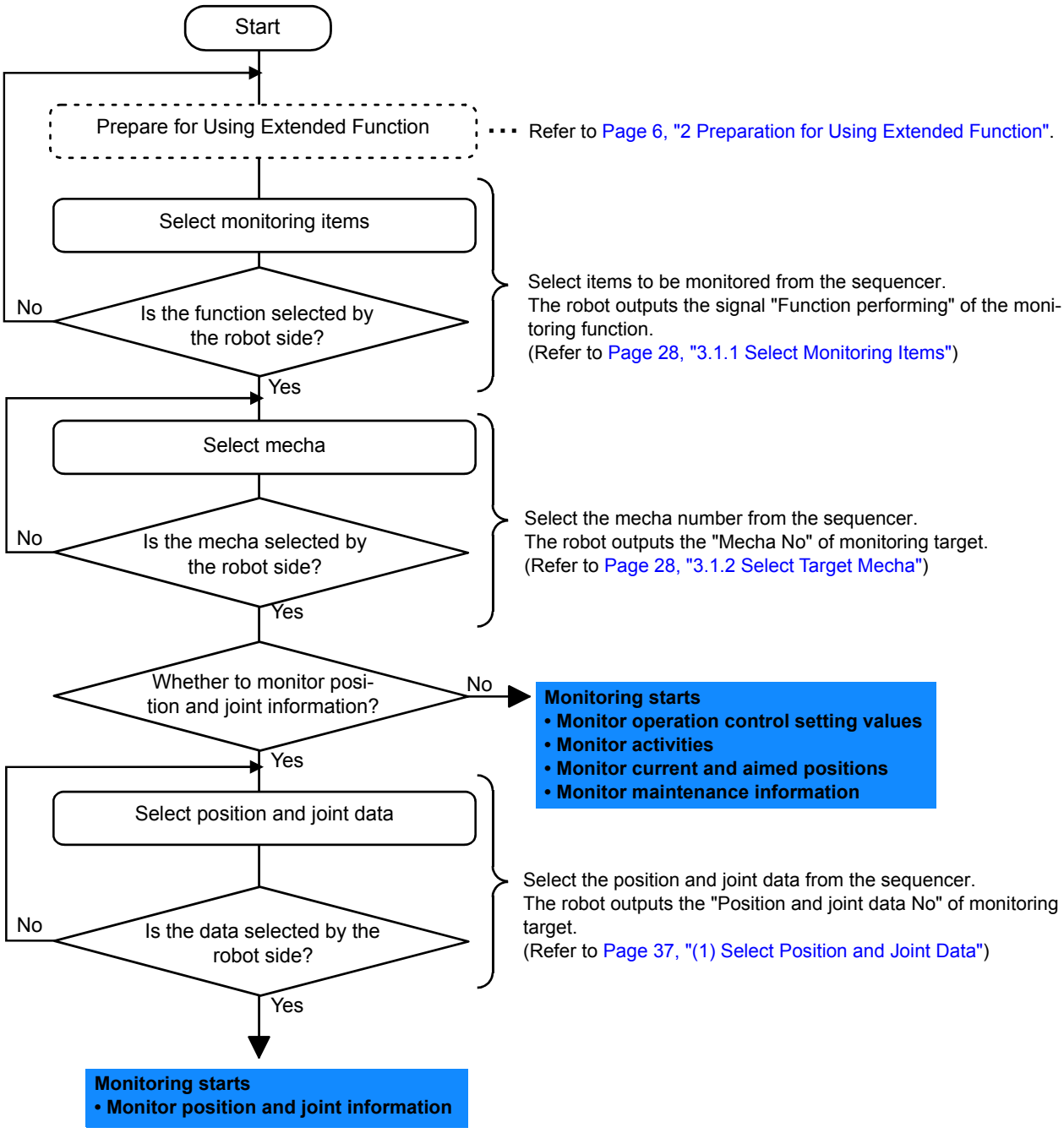
3 Monitor Robot Information

The Table 3-1 lists the robot information monitored from sequencer.
Setting values are also monitored during performing sequencer direct.

Table 3-1:Monitoring item list

No	Item	Description	I/F betw Robots	Update Cycle	Mecha No Setting	Section No
1	Monitor operation control setting values	Monitors the setting values relating to operation control command and operation control	Monitoring output (Robot side periodically updates the data in CPU buffer memory	3.5ms(CR800-R 7.1ms(CR800-Q)	○ (necessary)	"3.2.1"
2	Monitor activities	Monitors the robot's activities (current speed, arrival factor to the aimed position, etc.)		3.5ms(CR800-R 7.1ms(CR800-Q)	○	"3.2.2"
3	Monitor current and aimed positions	Monitors current and aimed positions of robot		3.5ms(CR800-R 7.1ms(CR800-Q)	○	"3.2.3"
4	Monitor position and joint information	Monitors various position type data (orientation at collision, etc.) and joint type data (current value, load factor, etc.)		Differ according to items	○	"3.2.4"
5	Monitor maintenance information	Monitors the maintenance information (battery and grease remaining times)		Depending on the parameter MFINTVL	○	"3.2.5"

3.1 Operation Flow



3.1.1 Select Monitoring Items

Here, selects the monitoring functions output by the robot from the sequencer.
Only the data specified by items (set to "1") selected with each bit can be monitored. For more information on each monitoring data, refer to Page 32, "3.2 Monitoring Item" and after.

(1) Sequencer output data

a) Word data

Sequencer Addr (offset)	Description	Remarks
512	Function selection [Allocated to each bit, 0: invalid, 1: valid] bit15 0 0000000000000000 |||||||+--bit0: (Reserved) ||||||+---bit1: (Reserved) |||||+----bit2: Monitor operation control settings ||||+-----bit3: Monitor activities |||+------bit4: Monitor current and aimed positions ||+-------bit5: Monitor position and joint information |+--------bit6: Monitor maintenance information +---------bit7: (Reserved)	

(2) Robot output data

a) Word data

Sequencer Addr (offset)	Description	Remarks
512	Function performing [allocated to each bit, 0: invalid, 1: valid] bit15 0 0000000000000000 |||||||+--bit0: (Reserved) ||||||+---bit1: (Reserved) |||||+----bit2: Monitor operation control settings ||||+-----bit3: Monitor activities |||+------bit4: Monitor current and aimed positions ||+-------bit5: Monitor position and joint information |+--------bit6: Monitor maintenance information +---------bit7: (Reserved)	

3.1.2 Select Target Mecha

Here, selects the target mecha number of monitoring data output by the robot from the sequencer.
The robot outputs the data with selected mecha number. The number (1 to 3) is selectable for mecha numbers. When the number other than 1 - 3 is specified, the data is initialized (zeros are put in the whole target area)

(1) Sequencer output data

a) Word data

Sequencer Addr (offset)	Description	Remarks
841	Specify a mecha number [1 - 3]	

(2) Robot output data

a) Word data

Sequencer Addr (offset)	Description	Remarks
731	Mecha number [1 - 3]	

3.1.3 Timing Chart

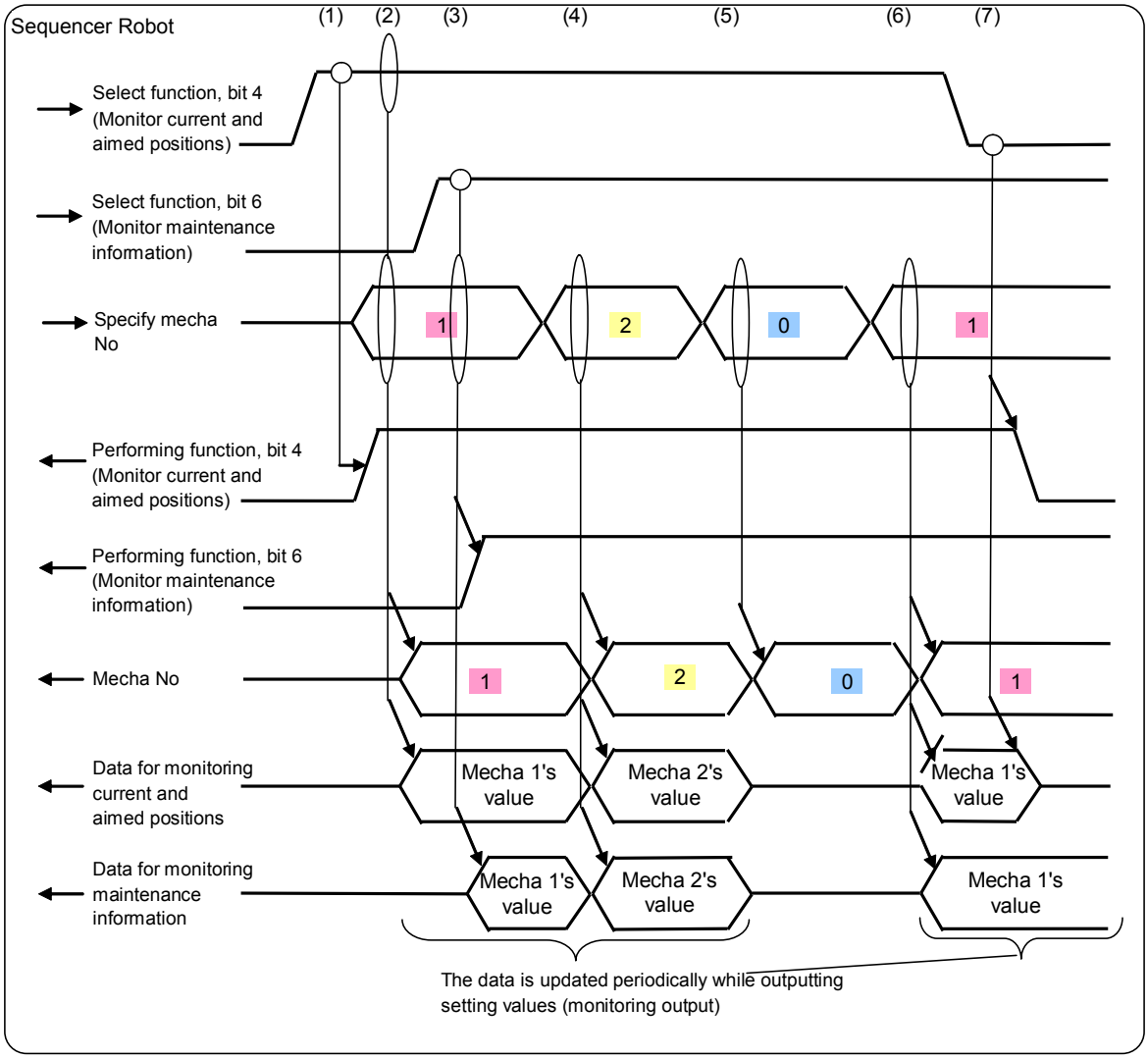


Fig.3-1:Timing chart for selecting monitoring items and target mecha

(1) When the sequencer sets the target bit of "Select function" to "ON", the robot sets the target bit of "Performing function" to "ON" to start the monitoring output of target item. Here, when "Specify mecha number" is other than 1 - 3, the robot waits to update the data.
(2) When the sequencer sets "Specify mecha number" to one, the robot starts to update mecha 1's data.
(3) When the target bit of "Select function" is set to "ON" while the sequencer sets "Specify mecha number", the robot starts to update the data of target item while at the same time the robot sets the target bit of "Performing function" to "ON".
(4) When the sequencer changes "Specify mecha number", the robot outputs the data of specified mecha.
(5) When the sequencer sets "Mecha number" to other than 1 - 3, the robot clears the output data.
(6) When the sequencer re-sets "Mecha number", the robot outputs the data of target mecha.
(7) When the sequencer sets the target bit of "Select function" to "OFF", the robot sets the target bit of "Performing function" to "OFF" to initialize the output data.

⚠ CAUTION The synchronization of data in CPU buffer memory is guaranteed on a per 32bit (2 word) basis. But, the synchronization in the unit more than this bit cannot be guaranteed. Therefore, be aware that the position type and joint type data is guaranteed for each axis, the data is not guaranteed as a whole.

3.1.4 Sample Ladder

Here, shows the sample ladder to retrieve current and aimed positions and maintenance information into the internal device by specifying the monitoring item and mecha number.

[Target function]
Select monitoring items (monitoring current and aimed positions, monitoring data of maintenance information) and mecha

[Target robot]
The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

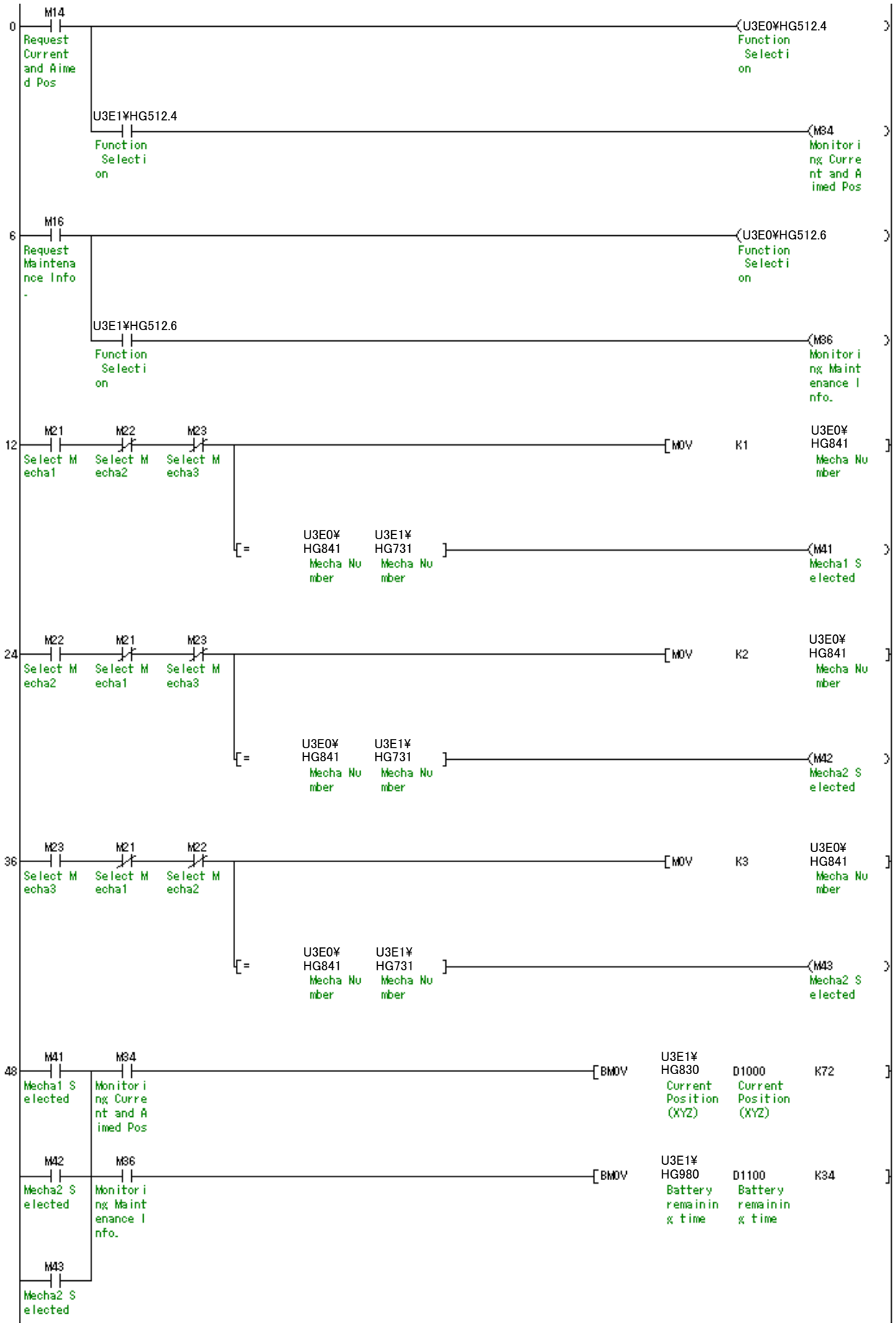
[Description]
When the device with desired function to be monitored is set to "ON" of monitoring request (M14) for current and aimed positions, monitoring request (M15) for maintenance information, and device of selecting mechas 1 - 3 (M21 - M23), target data is output to the internal device for monitoring.
The data of current and aimed positions is stored in D1000 - D1071.
The data of maintenance information is stored in D1100 - D1133.

Example:
To output the current and aimed positions data of mecha 1, set M14 and M21 to "ON" (M22 and M23 are "OFF") to output the monitoring data to D1000 to D1071.
To output the maintenance information data of mecha 1, set M16 and M21 to "ON" (M22 and M23 are "OFF") to output the monitoring data to D1100 to D1133.

[Device details]
M14: Request to monitor the current and aimed positions
M16: Request to monitor the maintenance information
M21: Selects mecha 1
M22: Selects mecha 2
M23: Selects mecha 3

M34: Monitoring the current and aimed positions
M36: Monitoring the maintenance information
M41: Mecha 1 selected
M42: Mecha 2 selected
M43: Mecha 3 selected
D1000 - D1071: Stores the current and aimed positions data from the robot
D1100 - D1133: Stores the monitoring data of maintenance information from the robot

[Ladder]


M14
Request Current and Aimed Pos
U3E0¥HG512.4 Function Selection
U3E1¥HG512.4 Function Selection
M34 Monitoring Current and Aimed Pos
M16
Request Maintenance Info.
U3E0¥HG512.6 Function Selection
U3E1¥HG512.6 Function Selection
M36 Monitoring Maintenance Info.
M21 Select Mecha1
M22 Select Mecha2
M23 Select Mecha3
MOV K1 U3E0¥HG841 Mecha Number
= U3E0¥HG841 Mecha Number U3E1¥HG731 Mecha Number
M41 Mecha1 Selected
MOV K2 U3E0¥HG841 Mecha Number
M42 Mecha2 Selected
MOV K3 U3E0¥HG841 Mecha Number
M43 Mecha2 Selected
BMOV U3E1¥HG830 Current Position (XYZ) D1000 Current Position (XYZ) K72
BMOV U3E1¥HG980 Battery remaining time D1100 Battery remaining time K34

3.2 Monitoring Item

3.2.1 Monitor Operation Control Setting Values

Here, periodically outputs the robot's operation control commands and the setting values for operation control to the CPU buffer memory.

(1) Monitoring data list

Sequencer Addr (Offset)	Description		Supported State Variable	Update Cycle
777	ColChk setting value	Collision detection setting [0: Invalid/ 1: Valid (error occurred)/ 2: Valid (error not occurred)		3.5ms(CR800-R) 7.1ms(CR800-Q)
778	ColLvl setting value	Collision detection level, J1 axis [%: 1 - 500]		
779		Collision detection level, J2 axis [%: 1 - 500]		
780		Collision detection level, J3 axis [%: 1 - 500]		
781		Collision detection level, J4 axis [%: 1 - 500]		
782		Collision detection level, J5 axis [%: 1 - 500]		
783		Collision detection level, J6 axis [%: 1 - 500]		
784		(Reserved)		
785		(Reserved)		
794	CMP Pos/Tool/Jnt setting values	Compliance coordinate type [0: Invalid/ 1: Perpendicular/ 2: Tool/ 3: Joint]		
795		Specify a compliance coordinate type [Specify target axis with bit] [Setting values to specify compliance axis of CMP Pos/Tool/Jnt setting values] The values below are set by setting up bit: bit7 0 00000000 |||||||+---bit0:J1/X axis ||||||+----bit1:J2/Y axis |||||+-----bit2:J3/Z axis ||||+------bit3:J4/A axis |||+-------bit4:J5/B axis ||+--------bit5:J6/C axis |+---------bit6:(Reserved) +----------bit7:(Reserved)		
796	CmpG setting value	Compliance J1/X axis gain [10^{-2}: 1 - 100]		
797		Compliance J2/Y axis gain [10^{-2}: 1 - 100]		
798		Compliance J3/Z axis gain [10^{-2}: 1 - 100]		
799		Compliance J4/A axis gain [10^{-2}: 1 - 100]		
800		Compliance J5/B axis gain [10^{-2}: 1 - 100]		
801		Compliance J6/C axis gain [10^{-2}: 1 - 100]		
802		(Reserved)		
803		(Reserved)		
804	MvTune/Prec setting values	Operation characteristic [1: Standard/ 2: High-speed/ 3: Track preferred/ 4: Vibration restricted]		

<Precautions>

● When the target mecha does not exist, outputs the data zero.

● The value below is output as ColChk:

- When multiple mechas are in use or when the element 1 of parameter COL is zero (collision detection unavailable),

  → zero is output

- Otherwise (collision detection available):

  When being in operation (including step feed, position jump operation, and sequencer direct performance),

  → the initial value is the value of element 2 of parameter COL, and then the output value is the value changed by ColChk command.

  When not being in operation (including suspension and jog operation),

  → it is set to the value of element 3 of parameter COL.

● The value below is output as ColLvl:

- When multiple mechas are in use or when the element 1 of parameter COL is zero (collision detection unavailable) and

  being in operation,

  → the initial value is the value of parameter COLLVL, and then the output value is the value changed by ColLvl command.

  When not being in operation,

  → it is the value during automatic operation is held when being in suspension, and it is the value of parameter COLLVL when being stopped.

- Otherwise (collision detection available),

  When being in operation,

  → the initial value is the value of parameter COLLVL, and then the output value is the value changed by ColLvl command.

  When not being in operation,

  → it is the value of parameter COLLVLJG.

● CMP Pos/Tool/Jnt setting values are set to zero when mechas 2, 3 are selected during using multiple mechas.
(User mecha cannot use compliance)

3.2.2 Monitor Activities

Here, periodically outputs the robot's activities (current speed, arrival factor to the aimed position, etc.) to the CPU buffer memory.

(1) Monitoring data list

Sequencer Addr (offset)	Description	Supported State Variable	Update Cycle
810	Current instruction speed [10^{-4}mm/s]	M_RSpd	3.5ms(CR800-R) 7.1ms(CR800-Q)
811			
812	Current distance remained [10^{-4}mm]	M_RDst	
813			
814	Distance between instructed and feedback positions [10^{-4}mm]	M_Fbd	
815			
816	Arrival factor [%] to the current aimed position	M_Ratio	
817	Current acceleration and deceleration state [0: Stopped/ 1: Accelerated/ 2: Constant speed/ 3: Decelerated]	M_AclSts	
818	Collision detection [1: Collided/ 0: Otherwise] Note1)	M_ColSts	
819	Going over the limit during performing compliance [1: Almost go over the limit/ 0: Does not go over the limit]	M_CmpLmt	
820	Deviance amount between instructed and actual positions during performing compliance [10^{-4}mm]	M_CmpDst	
821			

Note1) Robot state variable (M_ColSts) is "1" for about 3.5ms (CR800-R)/7.1ms (CR800-Q) between collision detection and servo OFF. But, the data "1" is output to the CPU buffer memory for 1sec after the collision is detected.

<Precautions>

- When the target mecha does not exist, outputs the data zero.
- When the data is dependent on a slot and the slot does not exist which has the control of target mecha, outputs the data zero. The data dependent on a slot is as follows:
  - Current distance remained (M_RDst)
  - Arrival factor to the current aimed position (M_Ratio)
  - Current acceleration and deceleration state (M_ActSts)

3.2.3 Monitor Current and Aimed Positions

Here, periodically outputs robot's current and aimed positions to the CPU buffer memory.

(1) Monitoring data list

Sequencer Addr (offset)	Description		Update Cycle
830	Current position (perpendicular)	X coordinate value [10^{-4}mm/10^{-4}deg]	3.5ms(CR800-R) 7.1ms(CR800-Q)
831			
832		Y coordinate value [10^{-4}mm/10^{-4}deg]	
833			
834		Z coordinate value [10^{-4}mm/10-4deg]	
835			
836		A coordinate value [10^{-4}mm/10^{-4}deg]	
837			
838		B coordinate value [10^{-4}mm/10^{-4}deg]	
839			
840		C coordinate value [10^{-4}mm/10^{-4}deg]	
841			
842		L1 coordinate value [10^{-4}mm/10^{-4}deg]	
843			
844		L2 coordinate value [10^{-4}mm/10^{-4}deg]	
845			
846		Structure flag	
847			
848		Multi-turn data	
849			
850	Aimed position (perpendicular)	X coordinate value [10^{-4}mm/10^{-4}deg]	
851			
852		Y coordinate value [10^{-4}mm/10^{-4}deg]	
853			
854		Z coordinate value [10^{-4}mm/10^{-4}deg]	
855			
856		A coordinate value [10^{-4}mm/10^{-4}deg]	
857			
858		B coordinate value [10^{-4}mm/10^{-4}deg]	
859			
860		C coordinate value [10^{-4}mm/10^{-4}deg]	
861			
862		L1 coordinate value [10^{-4}mm/10^{-4}deg]	
863			
864		L2 coordinate value [10^{-4}mm/10^{-4}deg]	
865			
866		Structure flag	
867			
868		Multi-turn data	
869			

Sequencer Addr (offset)	Description		Update Cycle
870	Current position (joint)	J1 coordinate value [10^{-4}mm/10^{-4}deg]	
871			
872		J2 coordinate value [10^{-4}mm/10^{-4}deg]	
873			
874		J3 coordinate value [10^{-4}mm/10^{-4}deg]	
875			
876		J4 coordinate value [10^{-4}mm/10^{-4}deg]	
877			
878		J5 coordinate value [10^{-4}mm/10^{-4}deg]	
879			
880		J6 coordinate value [10^{-4}mm/10^{-4}deg]	
881			
882		J7 coordinate value [10^{-4}mm/10^{-4}deg]	
883			
884		J8 coordinate value [10^{-4}mm/10^{-4}deg]	
885			
886	Aimed position (joint)	J1 coordinate value [10^{-4}mm/10^{-4}deg]	
887			
888		J2 coordinate value [10^{-4}mm/10^{-4}deg]	
889			
890		J3 coordinate value [10^{-4}mm/10^{-4}deg]	
891			
892		J4 coordinate value [10^{-4}mm/10^{-4}deg]	
893			
894		J5 coordinate value [10^{-4}mm/10^{-4}deg]	
895			
896		J6 coordinate value [10^{-4}mm/10^{-4}deg]	
897			
898		J7 coordinate value [10^{-4}mm/10^{-4}deg]	
899			
900		J8 coordinate value [10^{-4}mm/10^{-4}deg]	
901			

<Precautions>
- When the target mecha and axis do not exist, outputs the data zero.
- When the origin is not established, outputs zero for the both perpendicular and joint components of current position.

(2) Data description

[Perpendicular data]

- The unit is 10^{-4}mm or 10^{-4}deg.
- Only lower one word is used for the structure flag. Upper one word is a reserved area.

[Joint data]

- The unit is 10^{-4}mm or 10^{-4}deg.

3.2.4 Monitor Position and Joint Information

Here, periodically outputs the robot's various position type and joint type data to the CPU buffer memory.
The sequencer selects the data output by the robot. The area exists for one pieces of position type data and three pieces of joint type data and the data output for monitoring can be individually set by the sequencer.

(1) Select Position and Joint Data

In the sequencer, set up the number for position and joint data output by the robot.
The robot outputs the monitoring data corresponding to the selected data number.
The area exists for one pieces of position type data and three pieces of joint type data and the data can be individually set.
When the sequencer specifies the data with the number which is out of range, the robot sets all monitoring data to zero.

(1) Data list

a) Sequencer output

Sequencer Addr (offset)	Description
850	Position data selection [1 - 4] 1: (Reserved) 2: (Reserved) 3: (Reserved) 4: Direction at the time of collision
851	Joint data selection-1 [1 - 13] 1: (Reserved) 2: (Reserved) 3: Difference between estimated and actual torques when detecting a collision 4: (Reserved) 5: Current instruction 6: Maximum current instruction 1 7: Maximum current instruction 2 8: Current feedback 9: Allowable current instruction, minus side 10: Allowable current instruction, plus side 11: Effective current 12: Axis load level 13: Maximum axis load level
852	Joint data selection-2 [1 - 13] For setting values, refer to 851 above.
853	Joint data selection-3 [1 - 13] For setting values, refer to 851 above.

b) Robot output

Sequencer Addr (offset)	Description
906	Position data number [1 - 4]
907	Joint data number-1 [1 - 13]
908	Joint data number-2 [1 - 13]
909	Joint data number-3 [1 - 13]

(2) Timing chart

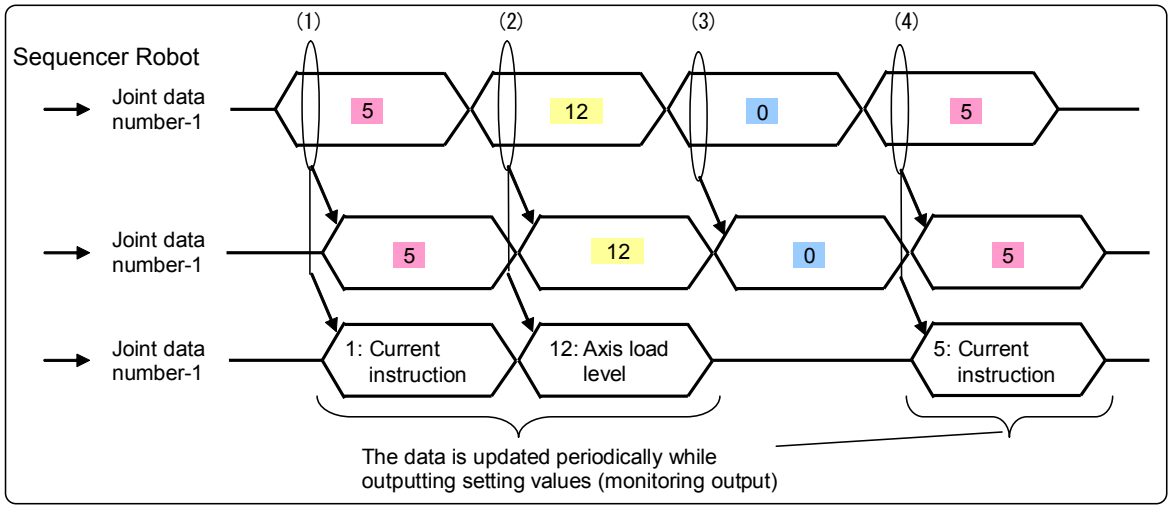


Fig.3-2:Joint data output, Timing chart

(1) When the sequencer selects "Joint data selection-1," the robot outputs the target data to "Joint data-1" area.
(2) When the sequencer changes "Joint data selection-1," the robot outputs the changed target data to "Joint data-1" area.
(3) When the sequencer selects the data out of valid range for "Joint data selection-1," the robot clears "Joint data-1" (set all components to zero) and outputs zero for "Joint number-1."
(4) When the sequencer reselects "Joint data selection-1", the robot outputs the target data to "Joint data-1" area.

* The same applies to Joint data-2, 3 and position data.

(2) Position and Joint Data

(1) Data list

b) Robot output

Sequencer Addr (offset)	Description	
910	Position data [1 - 4] 1: (Reserved) 2: (Reserved) 3: (Reserved) 4: Direction at the time of collision	X coordinate value
911		
912		Y coordinate value
913		
914		Z coordinate value
915		
916		A coordinate value
917		
918		B coordinate value
919		
920		C coordinate value
921		
922		L1 coordinate value
923		
924		L2 coordinate value
925		
926		Structure flag
927		
928		Multi-turn data
929		
930	Joint data-1 [1 - 13] 1: (Reserved) 2: (Reserved) 3: Difference between estimated and actual torques when detecting a collision 4: (Reserved) 5: Current instruction 6: Maximum current instruction 1 7: Maximum current instruction 2 8: Current feedback 9: Allowable current instruction, minus side 10: Allowable current instruction, plus side 11: Effective current 12: Axis load level 13: Maximum axis load level	J1 coordinate value
931		
932		J2 coordinate value
933		
934		J3 coordinate value
935		
936		J4 coordinate value
937		
938		J5 coordinate value
939		
940		J6 coordinate value
941		
942		J7 coordinate value
943		
944		J8 coordinate value
945		
946	Joint data-2 [1 - 13] * The data is similar to Joint data-1.	J1 coordinate value
947		
948		J2 coordinate value
949		
950		J3 coordinate value
951		
952		J4 coordinate value
953		
954		J5 coordinate value
955		
956		J6 coordinate value
957		
958		J7 coordinate value
959		
960		J8 coordinate value
961		

Sequencer Addr (offset)	Description	
962	Joint data-3 [1 - 13] * The data is similar to Joint data-1	J1 coordinate value
963		
964		J2 coordinate value
965		
966		J3 coordinate value
967		
968		J4 coordinate value
969		
970		J5 coordinate value
971		
972		J6 coordinate value
973		
974		J7 coordinate value
975		
976		J8 coordinate value
977		

<Precautions>
• When the target mecha and axis do not exist, outputs the data zero.

(2) Data description

The table below lists the content of each data item.

Item		Description	Setting Value (unit)	Supported State Variable	Update cycle
Position data	4: Direction at the time of collision Note1)	Robot's direction when the collision is detected	Divides the direction at the time of collision to components X, Y, Z. Specify the value with the proportion when the maximum moving axis value is set to ±100.	P_ColDir	3.5ms(CR800-R Note2)) 7.1ms(CR800-Q)
Joint data	3: Difference between estimated and actual torques when detecting a collision Note1)	Maximum difference value between estimated and actual torques when detecting a collision is valid	[10^{-3}%]	J_Colmxl	3.5ms(CR800-R Note2)) 7.1ms(CR800-Q)
	5: Current instruction	Outputs the current instruction value.	[10^{-3}Arms]		28ms
	6: Maximum current instruction 1	Outputs the maximum current instruction value after power-up. Reset when the robot power supply is shut off.	[10^{-3}Arms]		0.9sec
	7: Maximum current instruction 2	Outputs the maximum current instruction value for past 2sec.	[10^{-3}Arms]		0.9sec
	8: Current feedback	Outputs the current value generated in the servo motor.	[10^{-3}Arms]		3.5ms(CR800-R Note2)) 7.1ms(CR800-Q)
	9: Allowable current instruction, minus side	Outputs the maximum allowable value (minus side) of the current generated in the servo motor. * The value may vary according to jog and automatic operations.	[10^{-3}Arms]		3.5ms(CR800-R Note2)) 7.1ms(CR800-Q)
	10: Allowable current instruction, plus side	Outputs the maximum allowable value (plus side) of the current generated in the servo motor. * The value may vary according to jog and automatic operations.	[10^{-3}Arms]		3.5ms(CR800-R Note2)) 7.1ms(CR800-Q)
	11: Effective current	Outputs the effective value of current feedback.	[10^{-3}Arms]		28ms
	12: Axis load level	Outputs the motor's load level. The bigger this value, the heavier the load on the motor. Roughly it should be 80% or less. * It takes a few minutes until the value will stable.	[10^{-3}%]		0.9sec
	13: Maximum axis load level	Outputs the maximum value of axis load level after power-up. Reset when the power supply is shut off.	[10^{-3}%]		0.9sec

Note1) Because the collision detection function is unavailable during using multiple mechas, outputs zero.
Note2) Update cycle is 7.1ms when user mechanism is enabled in CR800-R.

3.2.5 Monitor Maintenance Information

Here, periodically outputs the robot's scheduled maintenance data (grease and belt remaining times) to the CPU buffer memory.

(1) Monitoring data list

Sequencer Addr (offset)	Description	Update Cycle
980	(Reserved)	Updated at scheduled interval set up in the second element of parameter "MFINTVL"
981		
982	Grease remaining time - J1 axis [Hr]	
983		
984	Grease remaining time - J2 axis [Hr]	
985		
986	Grease remaining time - J3 axis [Hr]	
987		
988	Grease remaining time - J4 axis [Hr]	
989		
990	Grease remaining time - J5 axis [Hr]	
991		
992	Grease remaining time - J6 axis [Hr]	
993		
994	Grease remaining time - J7 axis [Hr]	
995		
996	Grease remaining time - J8 axis [Hr]	
997		
998	Belt remaining time - J1 axis [Hr]	
999		
1000	Belt remaining time - J2 axis [Hr]	
1001		
1002	Belt remaining time - J3 axis [Hr]	
1003		
1004	Belt remaining time - J4 axis [Hr]	
1005		
1006	Belt remaining time - J5 axis [Hr]	
1007		
1008	Belt remaining time - J6 axis [Hr]	
1009		
1010	Belt remaining time - J7 axis [Hr]	
1011		
1012	Belt remaining time - J8 axis [Hr]	
1013		

<Precautions>

- When the target mecha does not exist, outputs all the data with zero.
- When the target mecha exists but the maintenance schedule is not supported, outputs all the data with "-1".
- When the target axis is not updated by the maintenance schedule, outputs the data "-1".

(2) Data description

[Grease remaining time]: Outputs the remaining time until the grease-up of each axis.
[Belt remaining time]: Outputs the remaining time until the belt exchange of each axis.

4 Reads/Writes Robot's Variables

4.1 Function Description

(1) Function list

The table below lists the variable operations performed from the sequencer:

Table 4-1:Variable operation function list

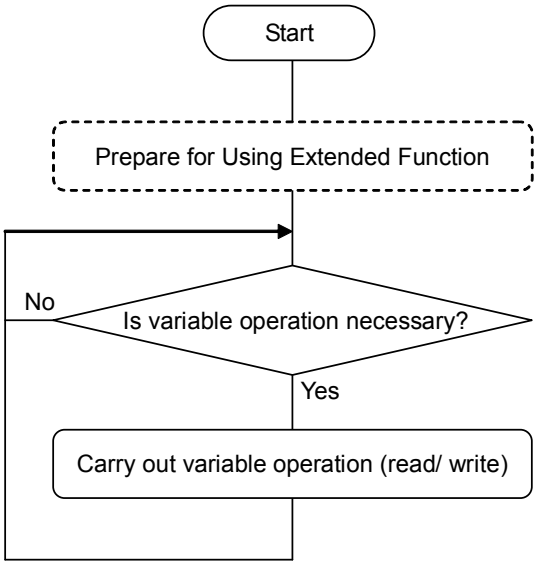
No	Item	Description	Robot's response time
1	Read numeric variable	Reads variable content by specifying slot number and variable name.	Answered within 1sec (it may vary according to the robot control's load state)
2	Write numeric variable	Rewrites variable content by specifying slot number, variable name, and variable content.	
3	Read position variable	Reads variable content by specifying slot number and variable name.	
4	Write position variable	Rewrites variable content by specifying slot number, variable name, and variable content.	
5	Read joint variable	Reads variable content by specifying slot number and variable name.	
6	Write joint variable	Rewrites variable content by specifying slot number, variable name, and variable content.	

(2) Functional requirements

Always available when a program is selected for robot's target slot and the target variable exists.
When the target is external variable, the variable operation is possible by specifying zero for a slot number, even when a program is not selected.

⚠ CAUTION **Be careful fully to change variable value.**
The robot's location and behavior may be changed by changing the variable value, thereby interfering with surrounding devices. Because it is especially dangerous when the robot is in operation, sufficiently check the value to be changed.

4.2 Operation Flow



··· Refer to Page 6, "2 Preparation for Using Extended Function".

···Refer to Page 44, "4.3 How to Operate Variables".

4.3 How to Operate Variables

Here, in the sequencer, operates the robot's variables (read/ write variables) by specifying function number, slot number, variable name, and variable data.
Function number setting allows you to select work type (read/ write variable) and variable type (numeric/ position/ joint variables) and specify a variable name (designation of ASCII character).

4.3.1 Data List

(1) Sequencer output data

1) Word data

Setting values when specifying ASCII character for variable and program names

Sequencer Addr (offset)	Item	Setting Value for Specifying ASCII Character									
		Numeric Var (Integer)		Position Var		Joint Var		Numeric Var (Long-precision integer number)		Numeric Var (Single-precision real number)	
		Read	Write	Read	Write	Read	Write	Read	Write	Read	Write
701	(Reserved)	(Reserved)									
702	Function No	101	102	104	105	107	108	111	112	121	122
703	Slot No	Slot number [0, 1 to the value of parameter TASKMAX]									
704	Program name (Not used)	(Not used)									
705											
706											
707											
708											
709											
710	Variable name	Variable name [ASCII data, up to 16 characters]									
711											
712											
713											
714											
715											
716											
717											
718	Variable data	(Not used)	Integer	(Not used)	X coordinate value	(Not used)	J1 coordinate value	(Not used)	Long-precision integer number value	(Not used)	Single-precision real number value
719											
720			(Not used)		Y coordinate value		J2 coordinate value		(Not used)		(Not used)
721											
722					Z coordinate value		J3 coordinate value				
723											
724					A coordinate value		J4 coordinate value				
725											
726					B coordinate value		J5 coordinate value				
727											
728					C coordinate value		J6 coordinate value				
729											
730					L1 coordinate value		J7 coordinate value				
731											
732					L2 coordinate value		J8 coordinate value				
733											
734					Structure flag		(Not used)				
735											
736					Multi-turn data						
737											

2) Bit signal

Sequencer Address		Description
Addr (offset)	Bit position	
700	0	Request for variable operation

(2) Robot output data

1) Word data

Setting values when specifying ASCII character for variable and program names

Sequencer Addr (offset)	Item	Setting Value for Specifying ASCII Character									
		Numeric Var (Integer)		Position Var		Joint Var		Numeric Var (Long-precision integer number)		Numeric Var (Single-precision real number)	
		Read	Write	Read	Write	Read	Write	Read	Write	Read	Write
551	Completion status	Completion status [1: OK/ other than 1: NG]									
552	Function No	101	102	104	105	107	108	111	112	121	122
553	Slot No	Slot number [0, 1 to the value of parameter TASKMAX]									
554	Program name	Program name, ASCII data, up to 12 characters]									
555											
556											
557											
558											
559											
560	Variable name	Variable name [ASCII data, up to 16 characters]									
561											
562											
563											
564											
565											
566											
567											
568	Variable data	Integer		X coordinate value		J1 coordinate value		Long-precision integer number value		Single-precision real number value	
569		(Not used)	(Not used)								
570				Y coordinate value		J2 coordinate value		(Not used)	(Not used)	(Not used)	(Not used)
571											
572				Z coordinate value		J3 coordinate value					
573											
574				A coordinate value		J4 coordinate value					
575											
576				B coordinate value		J5 coordinate value					
577											
578				C coordinate value		J6 coordinate value					
579											
580				L1 coordinate value		J7 coordinate value					
581											
582				L2 coordinate value		J8 coordinate value					
583											
584				Structure flag		(Not used)	(Not used)				
585											
586				Multi-turn data							
587											

2) Bit signal

Sequencer Address		Description
Addr (offset)	Bit position	
550	0	Variable operation completed

(3) Completion status

The values below are established as completion status:

Setting Value	Description
1	Successfully completed
2	Specified data (function number, slot number, variable number, element number, or external variable specification) out of range
3	Program not selected for the target slot
4	Target variable does not exist
5	(Reserved)
6	Not the formal variable data (at the time of writing variable)
7	Target variable not writable (at the time of writing variable)
8	Target variable value out of range at the time of reading variable: Not in the range between -32768 and 32767 (at the time of reading numeric variable)
10	NG because of a factor other than 2 to 8

(4) Data description

[Function No]

Select the target function.

Function number setting allows you to select work type (read/ write variable) and variable type (numeric/ position/ joint variables) and specify a variable name (designation of ASCII character).

[Slot number]

Select the target slot.

In general, specify a value between 1 and the value of parameter TASKMAX (factory default: 8). In case of external variable, "0" can be specified.

[Program name]

Program name is displayed in ASCII character.

● Specifying ASCII character

- Specify ASCII program name in 6 words area (12 characters).
- To specify ASCII characters, specify all 12 characters or string data including terminating code. However, leading and ending blank characters (space) are ignored.
- When target is an external variable and zero is specified for slot number, the program name becomes NULL.

[Variable name]

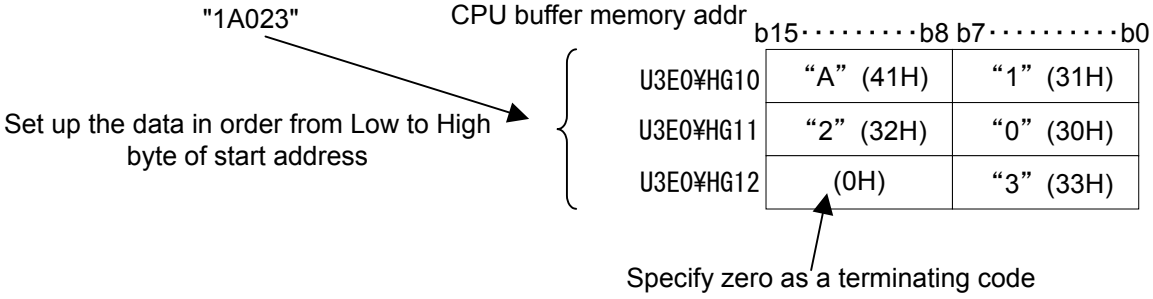
To specify a variable name, specify ASCII characters.

● Specifying ASCII character

- Specify the variable name (including leading character) in the 8 words area (16 characters, robot specification).
- To specify ASCII characters, specify all 16 characters or string data including terminating code. However, leading and ending whitespace characters (space) are ignored.
- The character underscore (_) used in array and external variable is also available, and array or external variable can be specified.

<ASCII data setting example>

- Set up the data in order from low to high byte of start address.
- Specify zero as a terminating code.
  (Be compliant with the character input specification of the sequencer)



<Available character>

Available characters are compliant with robot specification. (Refer to the table below.)

Category	Available Characters	Program Name	Variable Name
Alphabetic character	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	○	○
	a b c d e f g h i j k l m n o p q r s t u v w x y z	×	○
Figure	0 1 2 3 4 5 6 7 8 9	○	△ Note1)
Symbol	" ' & ( ) * + - . , / : ; = < > ? [\ ] ^ { } ~ | ! # $ %	×	× Note2)
	'_' (underscore)	×	△ Note3)
White space	Whitespace character	×	×

Note1) Only the alphabetic characters are available at the beginning of variable name. A figure is available for second and after characters.

Note2) Parentheses "()" for specifying an array are available.

Note3) Available for second and after characters. The variable whose second character is underscore '_' is an external variable.

[Variable data: numeric variable (Integer)]

- One word is prepared for a numeric variable and only an integer can be specified.
- Therefore, its range is between -32768 and 32767, and digits after decimal point are discarded.

[Variable data: numeric variable (Long-precision integer number)]

- Two words are prepared for a numeric variable and only an integer can be specified.
- Therefore, its range is between -2147483648 and 2147483647, and digits after decimal point are discarded.

[Variable data: position, joint, and numeric (Single-precision real number) variables]

- The unit is 10^{-4}mm or 10^{-4}deg.
  However, the number of significant figures for position and joint variable data output from the robot is dependent on the parameter PRGDPNTM (digits after decimal point: factory default is 2 or 3 digits (it may vary according to the robot model)), and the portion less than the significant figures is rounded off.
  For example, when PRGDPNTM is two, to round off 1.2345 gives 12300 and to round off 6.7890 gives 67900.
- Only lower one word is used for the structure flag of position variable, and upper one word is a reserved area.
- When a variable in undefined state (a variable exists but its data is empty) is read, zero is set to the undefined portion of data.
- Because each component value is handled as a single-precision floating type real number in the robot, the number of significant figures is about 7 digits.
  (The value which can be expressed with 24bit when expressed in binary number is about 7 digits when expressed in decimal number).

• When the data is successfully written into a variable, the variable data in the robot after the writing is read again and sent.
Therefore, even when writing into a position or joint variable is successfully ended, the data specified by the sequencer may be different from the data to be sent by the robot. The robot's posture data or the number of significant figures of data's digits after decimal point may differ.

4.3.2 Timing Chart

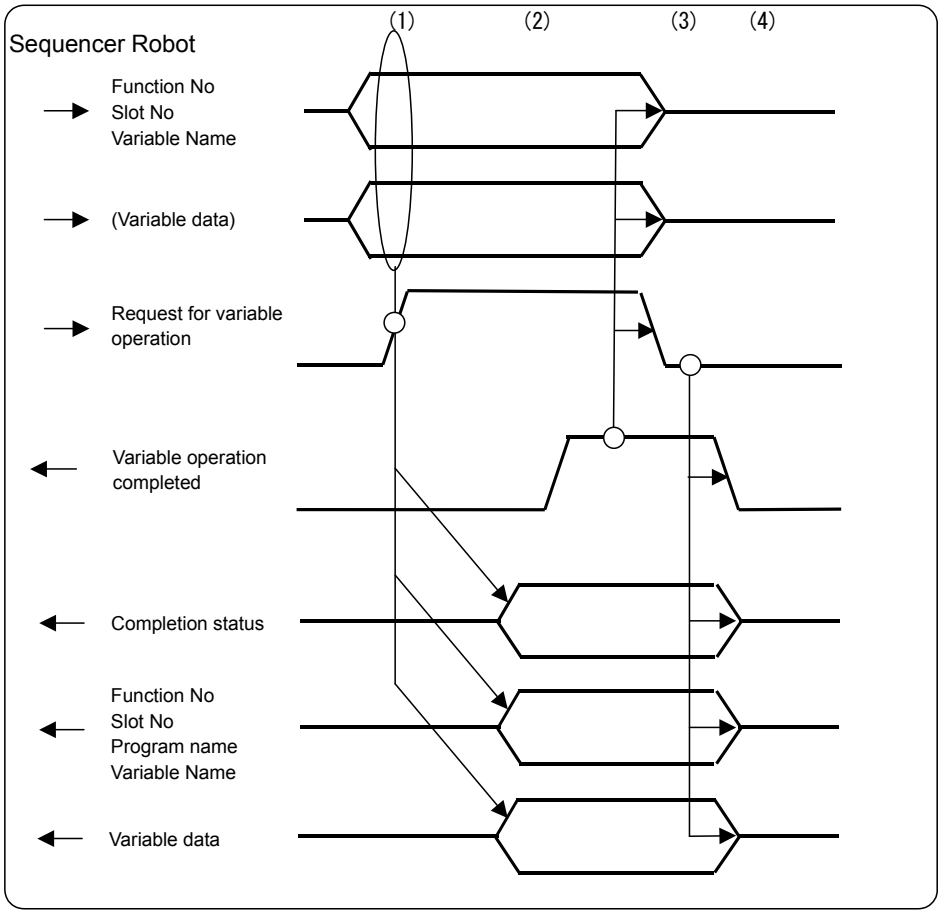


Fig.4-1:Variable operation timing chart

(1) The sequencer sets up "Function number", "Slot number", "Variable name", and "Variable data" (only for writing variable) and turns ON "Request for variable operation".
(2) When the robot receives "Request for variable operation ON", the robot operates the variable based on received data. When "Function number", "Slot number", "Variable name", "Variable data", and "Completion status" are specified after the operation, the robot turns ON "Variable operation completed".
When the operation cannot be carried out, the robot specifies a number indicating NG and turns ON "Variable operation completed".
(3) When "Variable operation completed ON" is received, the sequencer turns OFF "Request for variable operation".
(4) When received "Request for variable operation OFF", the robot turns OFF "Variable operation completed" and clears the data.

4.3.3 Sample Ladder

Here, describes a ladder example which reads the data by specifying a position variable name.

[Target function]

Reads position variable (designation of ASCII character)

[Target robot]

The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Description]

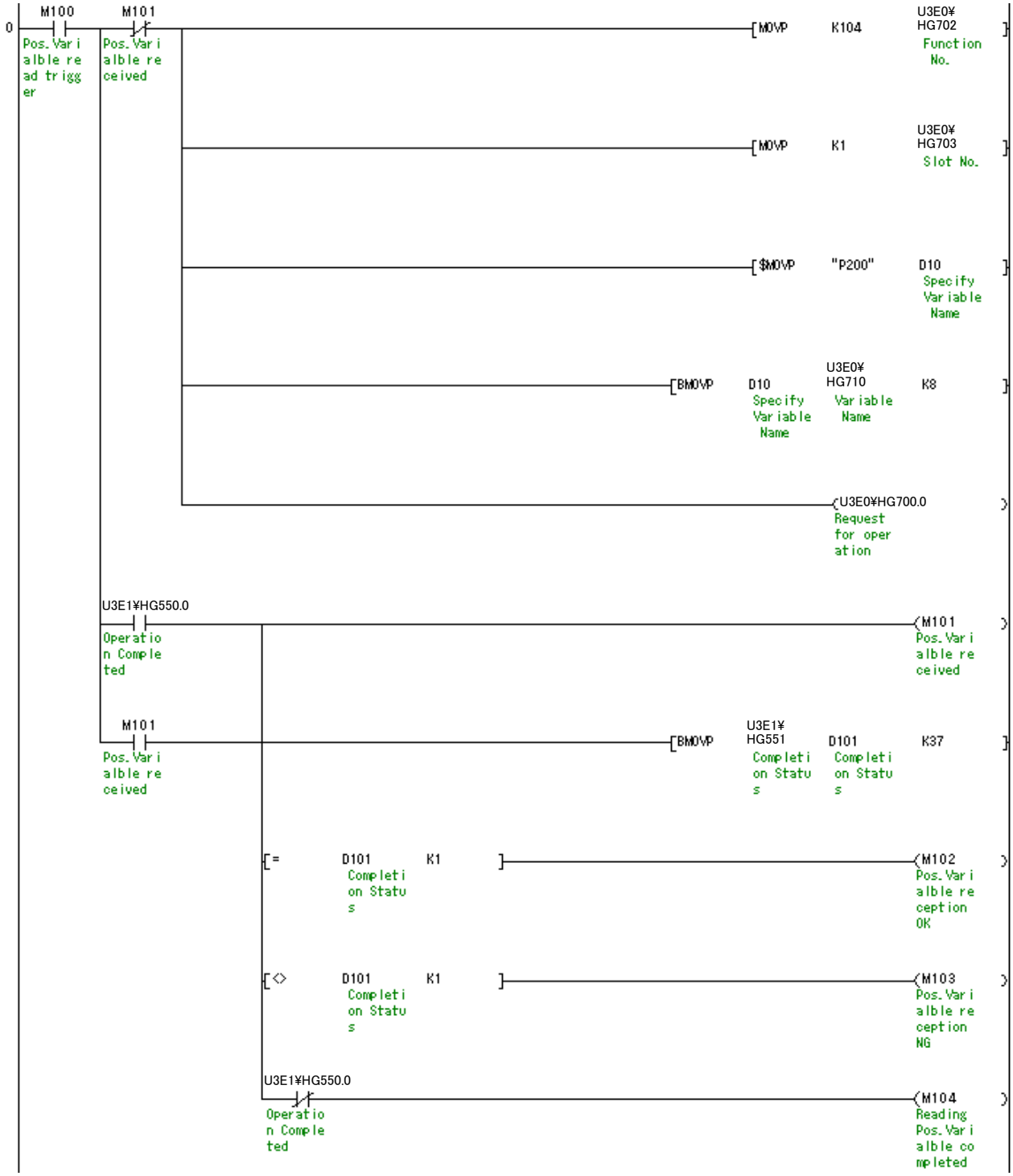
Turn ON the position variable read trigger (M100) to read the data of position variable P200 in slot 1.
The read position variable data is stored in D118 and after.
When the read handling is completed, M104 is turned ON. In this case, successful completion turns M102 ON and abnormal completion turns M103 ON.
When M104 is turned ON, turn OFF the position variable read trigger (M100).

[Device details]

D10 - D17: Specify variable name
M100: Position variable read trigger
M101: Position variable successfully received
M102: Position variable reception OK completed
M103: Position variable reception NG completed
M104: Reading position variable completed

D101: Received data from the robot (completion status)
D102: Received data from the robot (function number)
D103: Received data from the robot (slot number)
D104 - D109: Received data from the robot (program name)
D110 - D117: Received data from the robot (variable name)
D118 - D137: Received data from the robot (position variable data)

[Ladder]


0
M100
Pos.Variable read trigger
M101
Pos.Variable received
MOVP
K104
U3E0¥HG702
Function No.
MOVP
K1
U3E0¥HG703
Slot No.
$MOVP
"P200"
D10
Specify Variable Name
BMOVP
D10
Specify Variable Name
U3E0¥HG710
Variable Name
K8
U3E0¥HG700.0
Request for operation
U3E1¥HG550.0
Operation Completed
M101
Pos.Variable received
M101
Pos.Variable received
BMOVP
U3E1¥HG551
Completion Status
D101
Completion Status
K37
=
D101
Completion Status
K1
M102
Pos.Variable reception OK
<>
D101
Completion Status
K1
M103
Pos.Variable reception NG
U3E1¥HG550.0
Operation Completed
M104
Reading Pos.Variable completed

5 Read Current Line of Robot Program

5.1 Function Description

(1) Function list

The Table 5-1 lists the program operations performed from the sequencer.

Table 5-1:Program operation function list

No	Item	Description	Robot's Response Time
1	Read program's current line	● Reads currently performing robot program (one line, 128 characters) by specifying a slot number. ● Practicable when a program is selected for robot's slot.	Responds within 1s (it may vary according to the robot control's load state)

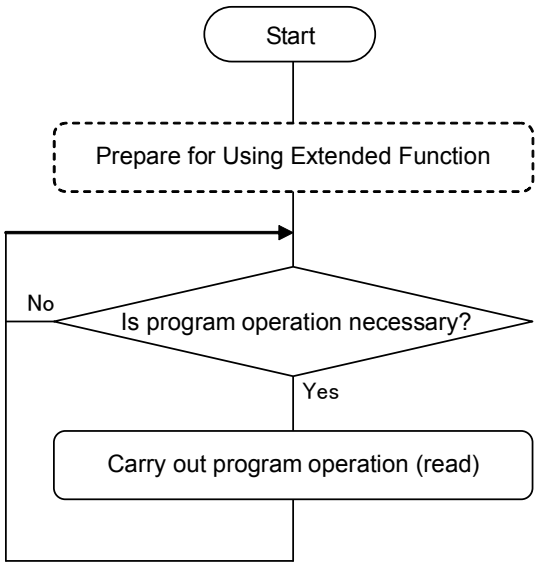
(2) Program data

The program data is as follows:

- The data is one line of program (up to 128 characters) in ASCII.
- When the data is less than 128 characters, terminating code 0 (NULL) is added at the end of string.
- Shift JIS codes are used for kanji character (similar to GOT specification).

⚠ CAUTION When a program line can be longer than 128 characters, the data after 128th character cannot be read.
Consequently, when the program whose line is longer than 128 characters is read and the data is written as-is into the robot, be careful that the data which exceeds 128 characters will be deleted.

5.2 Operation flow



··· Refer to Page 6, "2 Preparation for Using Extended Function".

··· Refer to Page 52, "5.3 How to Operate Program".

5.3 How to Operate Program

Here, in the sequencer, operates the robot program by specifying function number, slot number, program name, and program data.
Setting function number to '103' allows you to select a work type (read current line) and specify a program name (designation of ASCII character).

5.3.1 Data List

(1) Sequencer output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value for Specifying ASCII Character
		Program
		Read current line
740	(Reserved)	(Reserved)
741	Function No	103
742	Slot No	Slot number [1 to the value of parameter TASKMAX]
743	Program name	(Not used)
744		
745		
746		
747		
748		
749	Line No	(Not used)
750	(Reserved)	(Reserved)
751	Program data	(Not used)
752		
.		
.		
.		
813		
814		

2) Bit signal

Sequencer Address		Description
Addr (offset)	Bit position	
700	1	Request for program operation

(2) Robot output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value for Specifying ASCII Character Program Read current line
590	Completion status	Completion status [1: OK/ other than 1: NG]
591	Function No	103
592	Slot No	Slot number [1 to the value of parameter TASKMAX]
593	Program name	Program name, ASCII data, up to 12 characters
594		
595		
596		
597		
598		
599	Line No	Line No [1 - 32767]
600	Number of program characters	Number of program characters
601	Program data	Program to be read [ASCII data, up to 128 characters] * Shift JIS code for kanji
602		
.		
.		
.		
663		
664		

2) Bit signal

Sequencer Address		Description
Addr (offset)	Bit position	
550	1	Program operation completed

(3) Completion status

The values below are established as completion status:

Setting Value	Description
1	Successfully completed
2	Specified data (function number, slot number, program number) out of range
3	Program not selected for the target slot
4	(Reserved)
5	(Reserved)
6	(Reserved)
7	(Reserved)
10	NG because of a factor other than 2 to 7

(4) Data description

[Function No]

Selects the target function.
Function number setting allows you to select a work type (read current line) and specify a program name (designation of ASCII character).

[Slot number]

Select the target slot. Specify a value (factory default: 8) in the range between 1 and the value of parameter TASKMAX.

[Program name]

ASCII characters of the output program name.

- Specifying ASCII character
- Specify ASCII program name in 6 words area (12 characters).
- To specify ASCII characters, specify all 12 characters or string data including terminating code. However, leading and ending whitespace characters (space) are ignored.

For information about ASCII data, available characters, refer to Page 46, "(4) Data description".

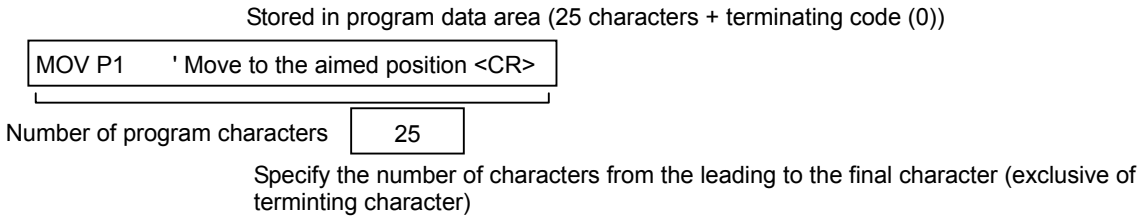
[Line No]

The line number of the read line is output.
When a program is selected but program is in abeyance (program is not running), the line number of first line is output.

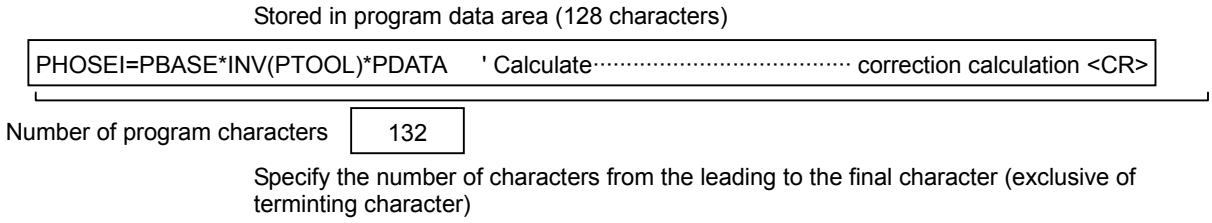
[Number of program characters]

Outputs the number of characters of target line in the target program.
Count and specify the number of characters from the leading to final character (exclusive of line feed/ terminating characters) including comment line (exclusive of line number).
When the target line is longer than 128 characters, up to 128 characters are read as a program data, but the number of counted characters is set as-is as the number of program characters.
When writing into a program, the number of characters of written line is set.

Example 1: A line is less than 128 characters:



Example 2: A line is more than 128 characters:



[Program data]

- The data is in ASCII format and up to 128 characters of program content are stored.
- Shift JIS codes are used for kanji.
- Line number is excluded from the program data.

5.3.2 Timing Chart

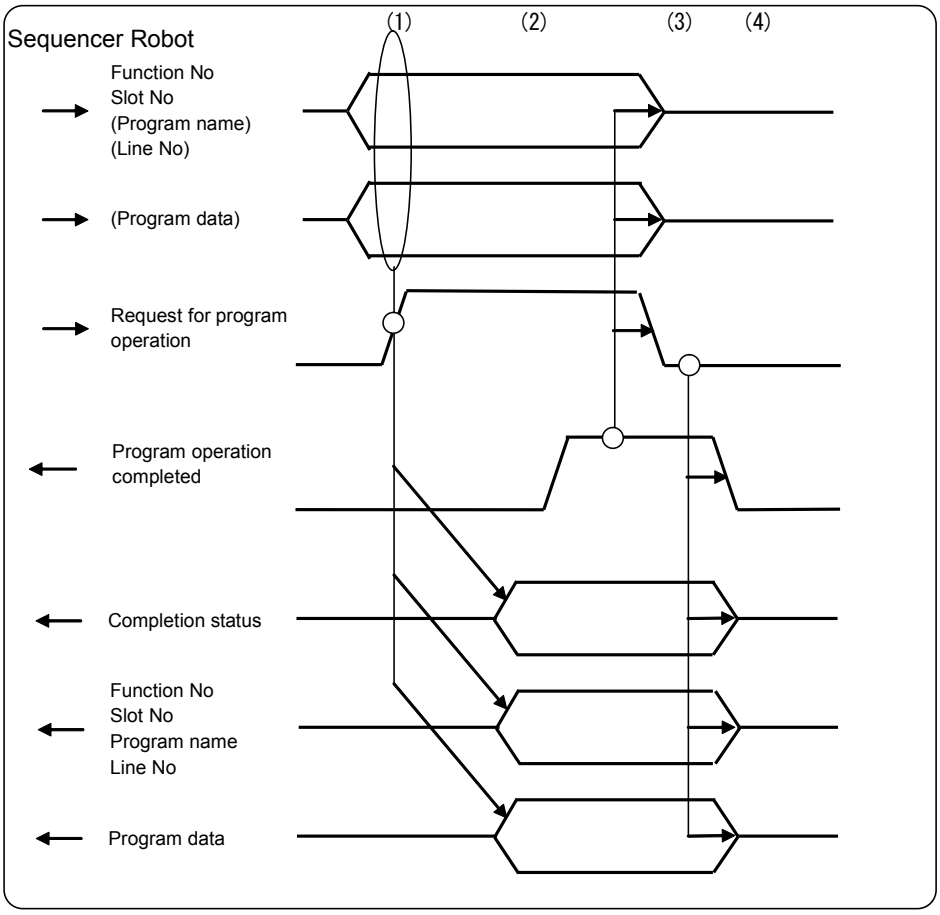


Fig.5-1:Program operation timing chart

(1) The sequencer sets up necessary data of "Function number", "Slot number", "Program name", "Line number", and "Program data" and turns ON "Request for program operation".
(2) When the robot receives "Request for program operation ON", the robot operates the program based on received data. When "Function number", "Slot number", "Program name", "Program data", and "Completion status" are specified after the operation, the robot turns ON "Program operation completed".
When the operation cannot be carried out, the robot specifies a number indicating NG and turns ON "Program operation completed".
(3) When "Program operation completed ON" is received, the sequencer turns OFF "Request for program operation".
(4) When received "Request for program operation OFF", the robot turns OFF "Program operation completed" and clears the data.

5.3.3 Sample Ladder

Here, describes a ladder example which reads the current line of a program performed by the robot.

[Target function]
Reads program's current line (designation of ASCII character)

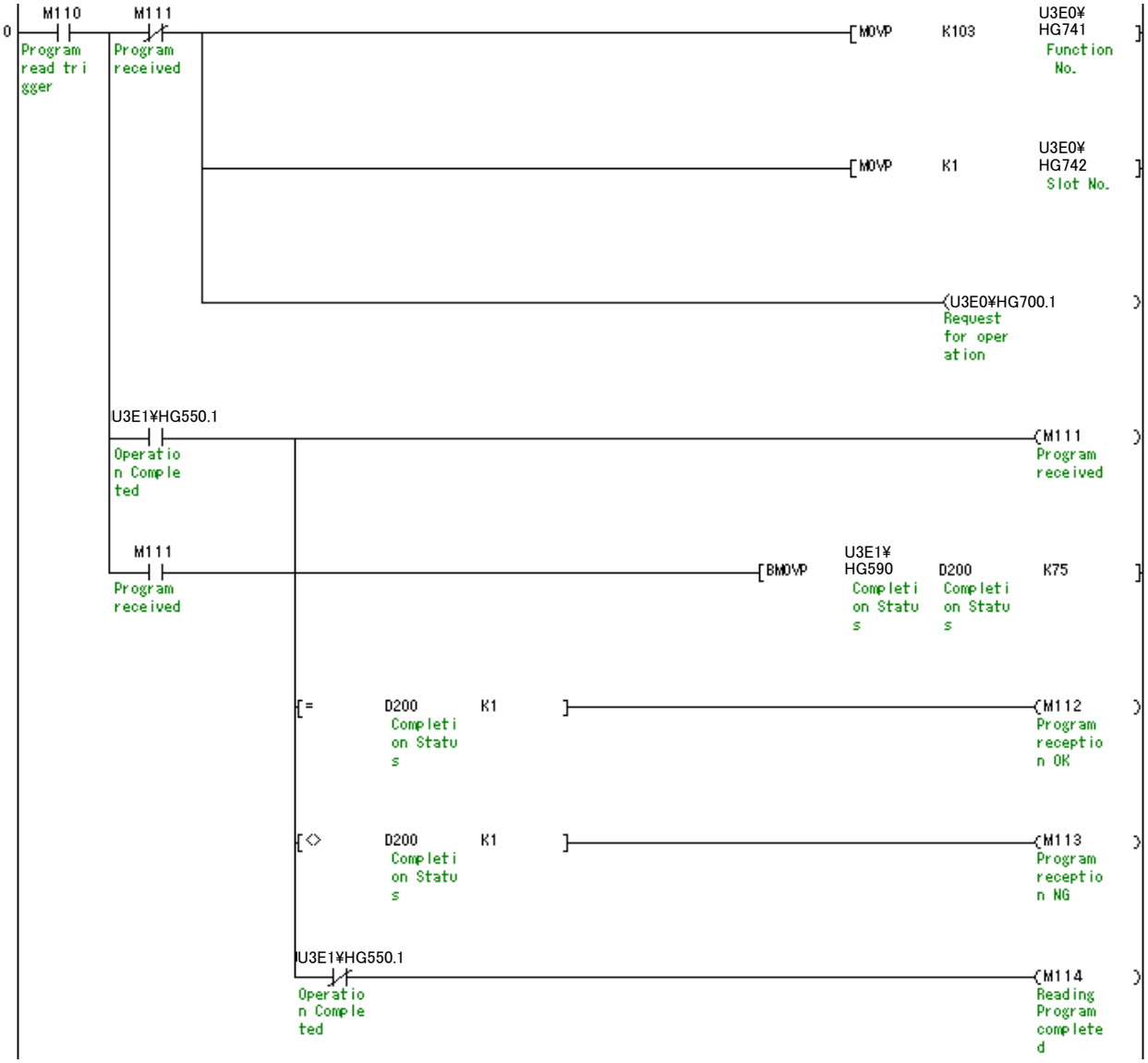
[Target robot]
The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Description]
Turn ON the program read trigger (M110) to read the program data of current line in slot 1.
The read program data is stored in D210 and after.
When the read handling is completed, M114 is turned ON. In this case, successful completion turns M112 ON and abnormal completion turns M113 ON.
When M114 is turned ON, turn OFF the program read trigger (M110).

[Device details]
M110: Program read trigger
M111: Program successfully received
M112: Program reception OK completed
M113: Program reception NG completed
M114: Reading program completed

D200: Received data from the robot (completion status)
D201: Received data from the robot (function number)
D202: Received data from the robot (slot number)
D203 - D208: Received data from the robot (program name)
D209: Received data from the robot (line number)
D210 - D273: Received data from the robot (program data)

[Ladder]


0
M110
Program read trigger
M111
Program received
MOVP K103 U3E0¥HG741 Function No.
MOVP K1 U3E0¥HG742 Slot No.
U3E0¥HG700.1 Request for operation
U3E1¥HG550.1
Operation Completed
M111 Program received
M111
Program received
BMOVP U3E1¥HG590 Completion Status D200 Completion Status K75
[= D200 Completion Status K1] M112 Program reception OK
[<> D200 Completion Status K1] M113 Program reception NG
U3E1¥HG550.1
Operation Completed
M114 Reading Program completed

6 Set up Robot's Maintenance

6.1 Function Description

(1) Function list

The Table 6-1 lists the maintenance setting performed from the sequencer.

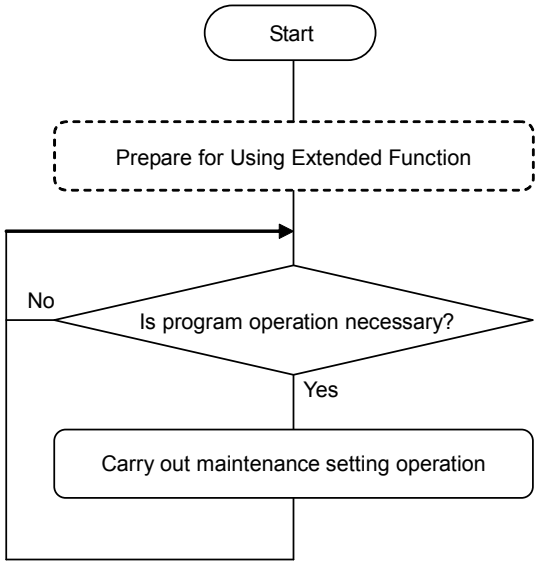
Table 6-1:Maintenance setting function list

No	Item	Description	Robot's Response Time
1	Reset maximum servomotor value	Resets the servo monitor's maximum values (current value, load factor, etc.) stored by robot to zero.	Responds within 1s (it may vary according to the robot control's load state)

(2) Functional requirements

Always practicable.

6.2 Operation flow



··· Refer to Page 6, "2 Preparation for Using Extended Function".

··· Refer to Page 60, "6.3 How to Operate Maintenance".

6.3 How to Operate Maintenance

Here, in the sequencer, operates the maintenance setting by specifying function number and setting data corresponding to the function.
Function number setting allows you to select function items.

6.3.1 Data List

(1) Sequencer output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value
		Reset Servo Monitor's Maximum/Minimum Values
820	(Reserved)	(Reserved)
821	Function No	6
822	Mecha No	Mecha No[1-3]
823	Mecha No	(Not used)
824		
825		
826		
827		
828		

2) Bit signal

Sequencer Address		Description
Addr (offset)	Addr (offset)	
700	2	Request for maintenance setting

(2) Robot output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value
		Reset Servo Monitor's Maximum/Minimum Values
670	Completion status	Completion status [1: OK/ other than 1: NG]
671	Function No	6
672	Mecha No	Mecha No[1-3]
673	Mecha No	(Not used)
674		
675		
676		
677		
678		

2) Bit signal

Sequencer Address		Description
Addr (offset)	Addr (offset)	
550	2	Maintenance setting completed

(3) Completion status

The values below are established as completion status:

Setting Value	Description
1	Successfully completed
2	Specified "Function number" and "Mecha number" are out of range (including the case that the target mecha does not exist).
3	(Not used)
4	No target function (the function specified by target mecha does not exist)
10	NG because of a factor other than 2 to 4

(4) Data description

[Function No]

Selects the target function.

[Mecha No]

Select the target mecha. Specify a mecha in the range of mechas 1 - 3.

6.3.2 Timing Chart

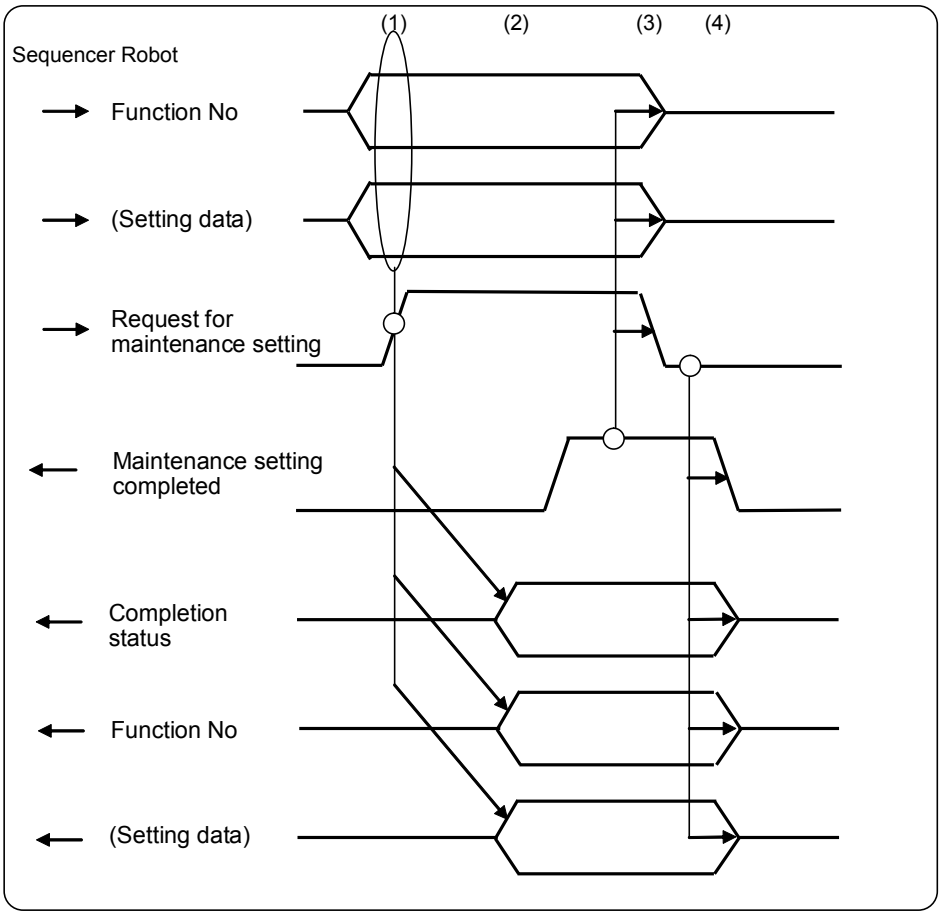


Fig.6-1:Maintenance function timing chart

(1) The sequencer sets up necessary data of "Function number" and "Setting data" and turns ON "Request for maintenance setting."
(2) When the robot received "Request for maintenance setting ON," the robot operates the maintenance setting based on received data. When "Function number", "Setting data", and "Completion status" are specified after the operation, the robot turns ON "Maintenance setting completed."
When the operation cannot be carried out, the robot specifies a number indicating NG and turns ON "Maintenance setting completed."
(3) When "Maintenance setting completed ON" is received, the sequencer turns OFF "Request for maintenance setting."
(4) When "Request for maintenance setting OFF" is received, the robot turns OFF "Maintenance setting completed" and clears the data.

6.3.3 Sample Ladder

Here, describes a ladder example which resets the servo data's maximum values (current value, load factor) stored in the robot.

[Target function]

Reset the maximum servo monitor value

[Target robot]

The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Description]

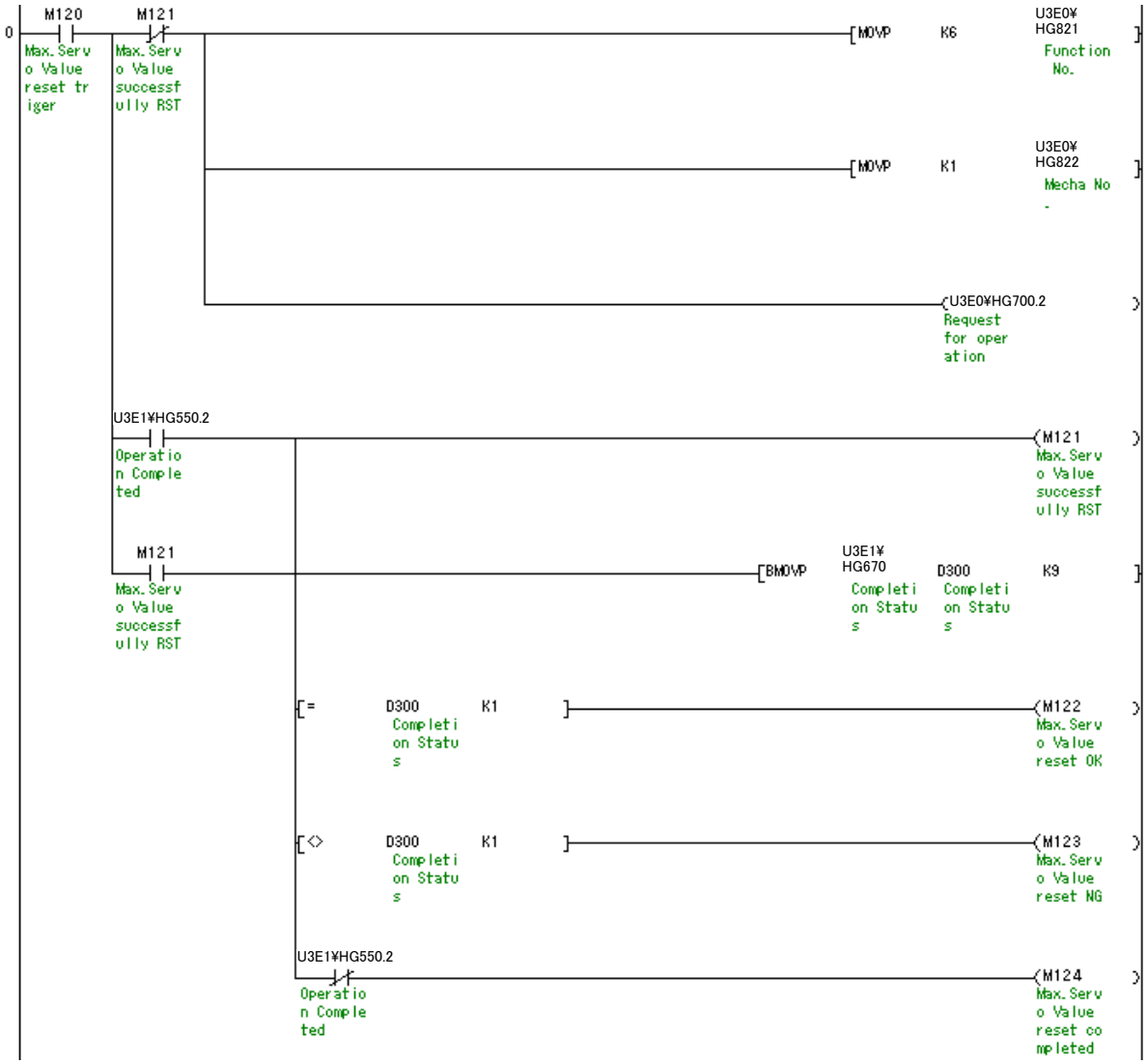
Turn ON the maintenance setting read trigger (M120) to reset the maximum servo monitor values.
Output data from the robot is stored in D302 and after.
When the reset handling is completed, M124 is turned ON. In this case, successful completion turns M122 ON and abnormal completion turns M123 ON.
When M124 is turned ON, turn OFF the maintenance setting read trigger (M120).

[Device details]

M120: Maximum servo monitor value reset trigger
M121: Maximum servo monitor value successfully reset
M122: Maximum servo monitor value reset OK completed
M123: Maximum servo monitor value reset NG completed
M124: Maximum servo monitor value reset completed

D300: Received data from the robot (completion status)
D301: Received data from the robot (function number)
D302: Received data from the robot (mecha number)

[Ladder]


0
M120
Max. Servo Value reset triger
M121
Max. Servo Value successfully RST
MOVP
K6
U3E0¥HG821
Function No.
MOVP
K1
U3E0¥HG822
Mecha No.
U3E0¥HG700.2
Request for operation
U3E1¥HG550.2
Operation Completed
M121
Max. Servo Value successfully RST
M121
Max. Servo Value successfully RST
BMOVP
U3E1¥HG670
Completion Status
D300
Completion Status
K9
=
D300
Completion Status
K1
M122
Max. Servo Value reset OK
<>
D300
Completion Status
K1
M123
Max. Servo Value reset NG
U3E1¥HG550.2
Operation Completed
M124
Max. Servo Value reset completed

7 Read Robot Information

7.1 Function Description

(1) Function list

The Table 7-1 lists the robot information reading performed from the sequencer.

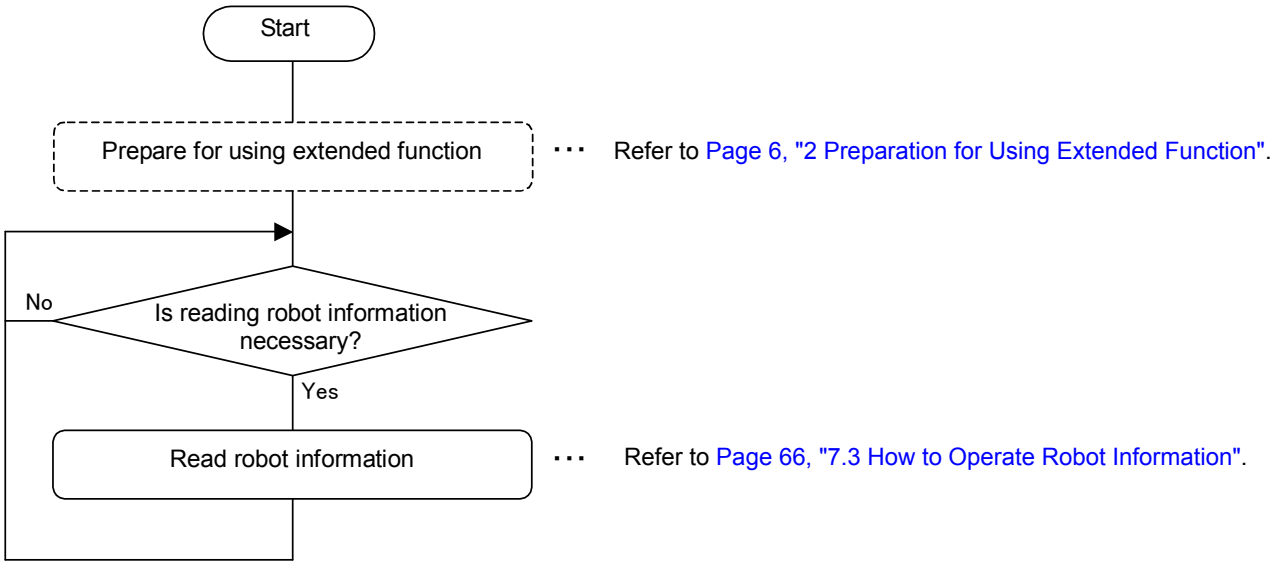
Table 7-1:Robot information reading function list

No	Item	Description	Robot's Response Time
1	Read error information	Reads the detailed error information generated in the robot. When multiple errors occur, three information can be read at the same time, and the information to be read can be changed by specifying the start number.	Responds within 1s (it may vary according to the robot control's load state)
2	Read product information	Read the robot's product information.	

(2) Functional requirements

Always practicable.

7.2 Operation flow



7.3 How to Operate Robot Information

Here, reads the robot information from the sequencer by specifying function number and setting data.
Function number allows you to select the robot information to be read.

7.3.1 Data List

(1) Sequencer output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value	
		Read Error Information	Read Product Information
830	(Reserved)	(Reserved)	
831	Function No	3	4
832	Setting No	Start number [1 -]	(Not used)

2) Bit signal

Sequencer Address		Description
Addr (offset)	Addr (offset)	
700	3	Request for reading information

(2) Robot output data

1) Word data

Sequencer Addr (offset)	Item	Setting Value	
		Read Error Information	Read Product Information
680	Completion status	Completion status [1: OK/ other than 1: NG]	
681	Function No	3	4
682	Read data	Start number [1 -]	(Not used)
683		Number of errors occurred	Robot type name [ASCII data, up to 20 characters]
684		Information 1 (error No)	
685		Information 1 (error occurred program name) [ASCII data, up to 12 characters]	
686			
687			
688			
689			
690			
691		Information 1 (occurred line No)	
692		Information 1 (detailed error No)	
693			Controller version [ASCII data, up to 6 characters]
694		Information 1 (occurred slot No)	
695		(Reserved)	
696			Controller serial No [ASCII data, up to 16 characters]
697			
698		Information 2 (error No)	
699		Information 2 (error occurred program name) [ASCII data, up to 12 characters]	
700			
701			
702			
703			
704			Robot serial No [ASCII data, up to 16 characters]
705		Information 2 (occurred line No)	
706		Information 2 (detailed error No)	
707			
708		Information 2 (occurred slot No)	
709		(Reserved)	
710			
711			
712		Information 3 (error No)	(Not used)
713		Information 3 (error occurred program name) [ASCII data, up to 12 characters]	
714			
715			
716			
717			
718			
719		Information 3 (occurred line No)	
720		Information 3 (detailed error No)	
721			
722		Information 3 (occurred slot No)	
723		(Reserved)	
724			
725			

2) Bit signal

Sequencer Address		Description
Addr (offset)	Addr (offset)	
550	3	Reading information completed

(3) Completion status

The values below are established as completion status:

Setting Value	Description
1	Successfully completed
2	Specified "Function number" out of range
3	Specified "Setting data" out of range
10	NG because of a factor other than 2 and 3

(4) Data description

[Function No]

Selects the target function.

[Start No of read data]

Specify the information's start number to be read.

The robot reads and stores three pieces of information from the specified number in the CPU buffer memory.

Specify 1: Reads first to third pieces of registered information.

Specify 2: Reads second to fourth pieces of registered information.

Specify 3: Reads third to fifth pieces of registered information.

Of information 1 - 3, the information with small number is a new error.

When the target information with the specified number does not exist, the robot sets all read data to zero.

7.3.2 Timing Chart

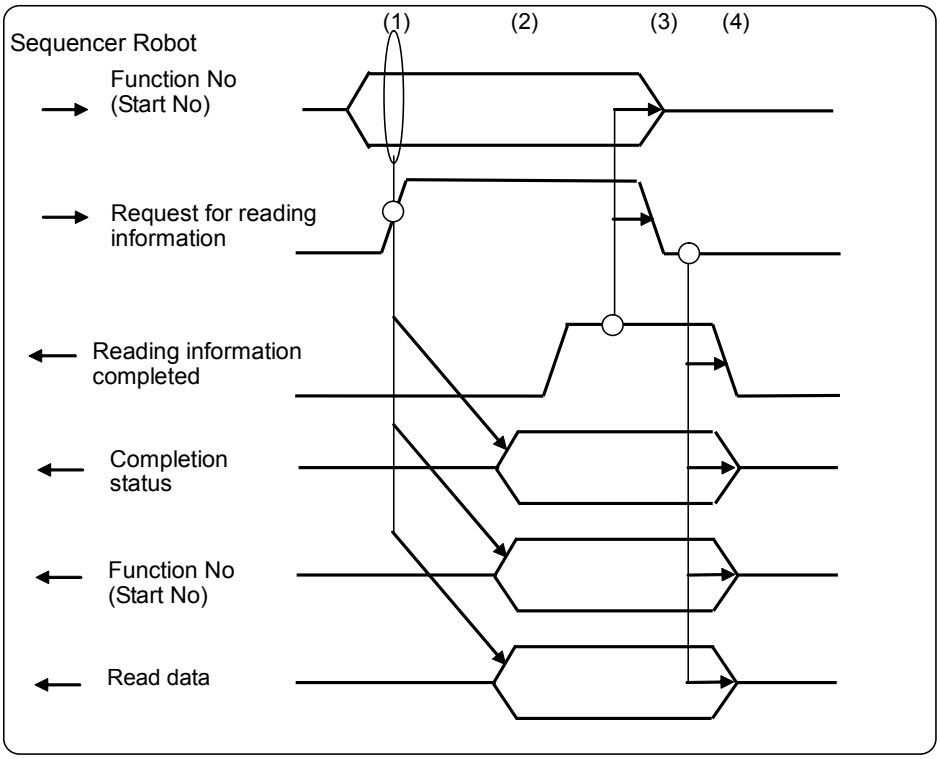


Fig.7-1:Information reading timing chart

(1) The sequencer sets up necessary data of "Function number" and "Start number" and turns ON "Request for reading information."
(2) When "Request for reading information ON" is received, the robot specifies requested "Read data" and "Completion status" and turns ON "Reading information completed."
When the operation cannot be carried out, the robot specifies a number indicating NG and turns ON "Reading information completed."
(3) When "Reading information completed" is received, the sequencer turns OFF "Request for reading information."
(4) When "Request for reading information OFF" is received, the sequencer turns OFF "Reading information completed."

7.3.3 Sample Ladder

Here, describes a ladder example which reads the detailed error information occurred in the robot.

[Target function]

Read error information

[Target robot]

The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Description]

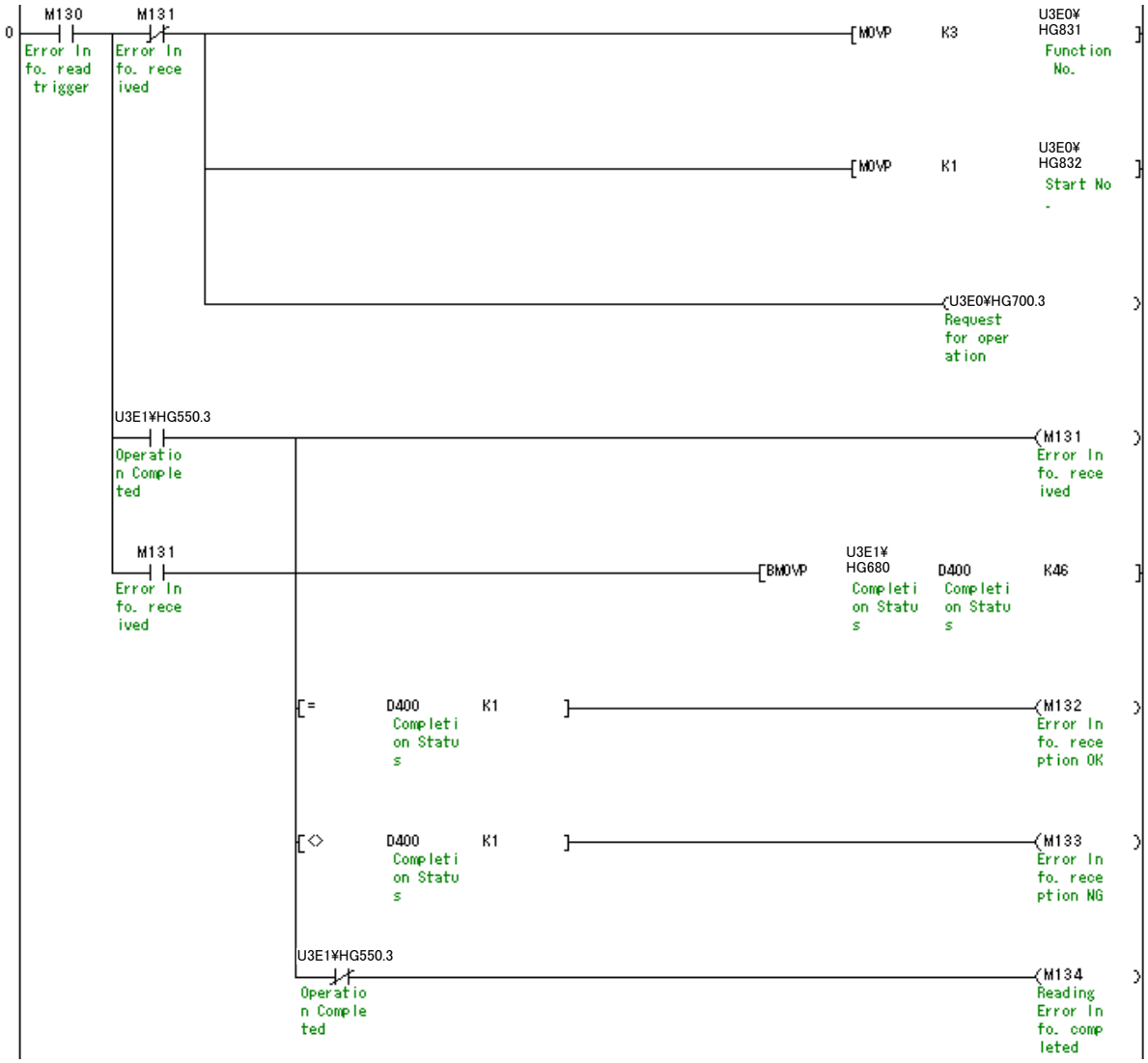
Turn ON the error information read trigger (M130) to read the robot error information (first thee pieces of information from start).
The read error information is stored in D403 and after.
When the read handling is completed, M134 is turned ON. In this case, successful completion turns M132 ON and abnormal completion turns M133 ON.
When M134 is turned ON, turn OFF the error information read trigger (M130).

[Device details]

M130: Error information read trigger
M131: Error information received successfully
M132: Error information reception OK completed
M133: Error information reception NG completed
M134: Reading error information completed

D400: Received data from the robot (completion status)
D401: Received data from the robot (function number)
D402: Received data from the robot (start number)
D403 - D445: Received data from the robot (error information)

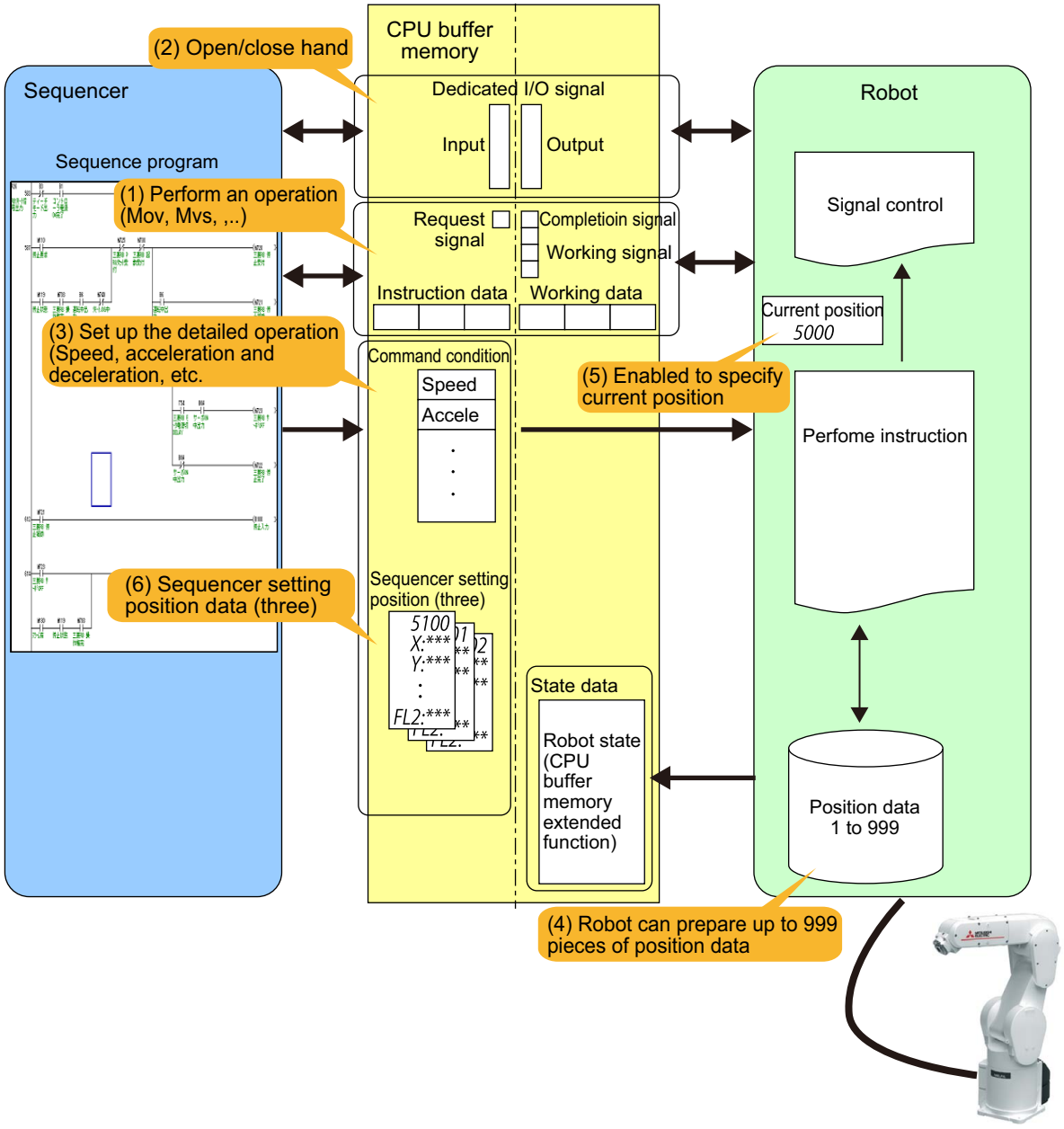
[Ladder]


0
M130
Error In fo. read trigger
M131
Error In fo. rece ived
MOVP
K3
U3E0¥ HG831
Function No.
MOVP
K1
U3E0¥ HG832
Start No -
U3E0¥HG700.3
Request for oper ation
U3E1¥HG550.3
Operatio n Comple ted
M131
Error In fo. rece ived
M131
Error In fo. rece ived
BMOVP
U3E1¥ HG680
Completi on Statu s
D400
Completi on Statu s
K46
[=
D400
Completi on Statu s
K1
M132
Error In fo. rece ption OK
[<>
D400
Completi on Statu s
K1
M133
Error In fo. rece ption NG
U3E1¥HG550.3
Operatio n Comple ted
M134
Reading Error In fo. comp leted

8 Perform Sequencer Direct

8.1 Sequencer Direct Performance Function

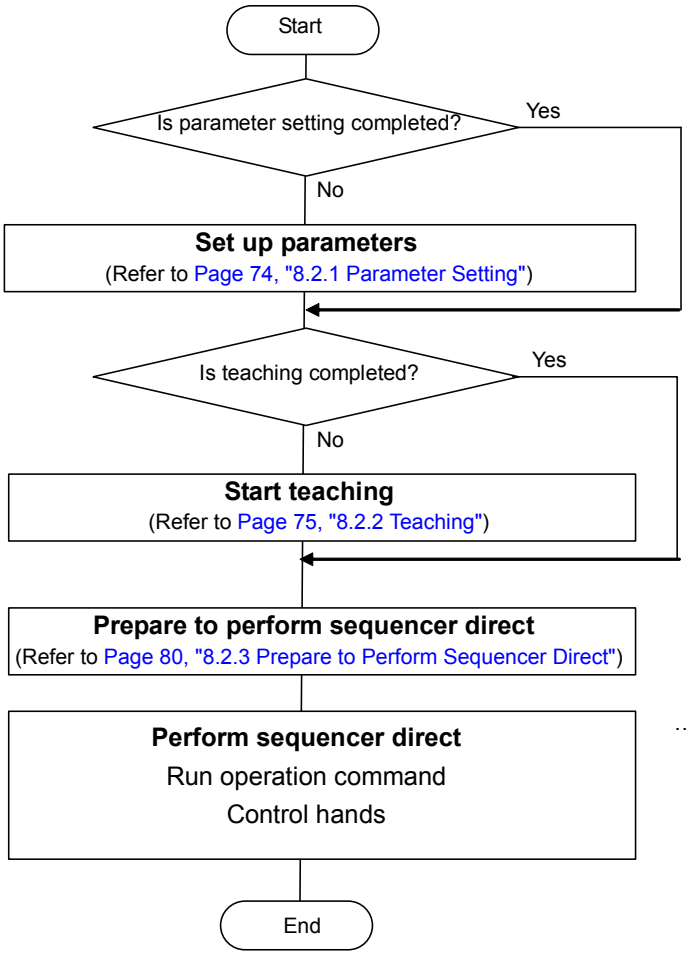
The sequencer direct performance function directly operates the robot by using the extended CPU buffer memory.
The performance function is composed of robot operation, hand open/close, working speed/ acceleration setting, position data management, etc.
The figure below shows the data flow between sequencer and robot when performing the sequencer direct.



(1) The sequencer specifies a command (instruction) to directly operate the robot (sequencer direct performance).
(2) The sequencer controls a hand (dedicated I/O signal control).
(3) The sequencer specifies the command conditions and speed, acceleration, tool setting, etc. for the sequencer direct performance.
(4) The sequencer prepares up to 999 pieces of position data for sequencer direct performance in the robot controller.(Teaching data does not occupy all the memory area of the sequencer device.)
(5) The sequencer can move the robot relatively with reference to robot's current position.
(6) The sequencer moves the robot to the position by specifying the position data.

8.2 Operation flow

The figure below shows the operation flow when performing the sequencer direct.



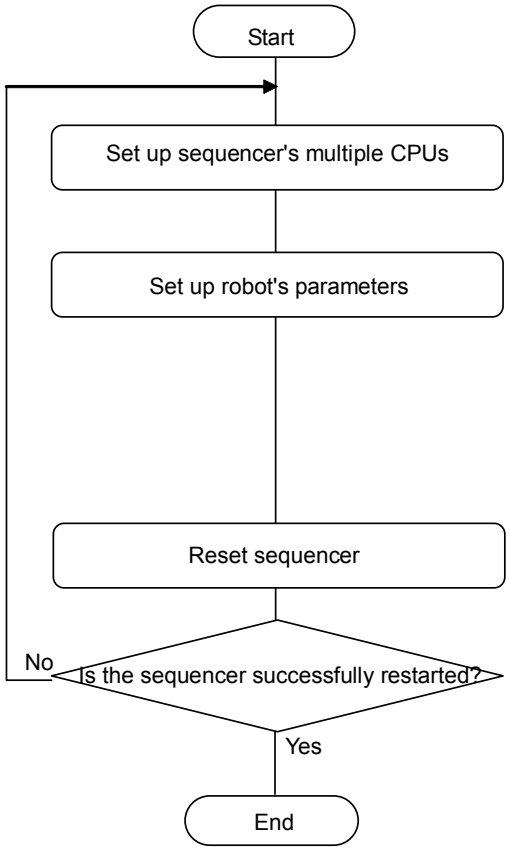
… ● Set up multiple CPUs for sequencer and robot CPUs. When already established, carry out the next teaching operation.

… ● Teach robot's working position. When teaching was already established, prepare the next sequencer direct performance and perform the sequencer direct.

… ● "8.3.1Operation Command"
● "8.3.4Control Robot Hand"

* In general, parameter setting and teaching are carried out only once on startup.
They are not necessary for operation after the startup.

8.2.1 Parameter Setting



… ● Sets up the sequencer's parameters (multiple CPU setting).
For robot CPU, allocate 1k words of free user area in high-speed communication area of multiple CPUs to robot's I/O area.
(Refer to Page 7, "2.1.1 Set up Sequencer's Multiple CPUs")

… ● Sets up multiple CPUs' parameters.
(Refer to Page 11, "2.1.2 Set up Robot's Multiple CPUs")
● Sets up parameter IQMEM
1) To enable the extended function, set bit 0 to one.
2) To enable the sequencer direct performance, set bit 1 to one.
(Refer to Page 12, "2.1.3 Set up Parameter for Selecting CPU buffer Memory Extended Function")
● Set robot language to "MELFA BASIC V" or "MELFA BASIC VI."
(Refer to Page 13, "2.1.4 Check Robot Language Setting")

… ● Carry out the reset operation of sequencer or reset the sequencer by turning ON the power.

For information on how to set up parameters, refer to Page 6, "2 Preparation for Using Extended Function".

8.2.2 Teaching

Here, teaches the position data for performing the robot's sequencer direct.

(1) Position Data

The position data handled in the sequencer direct performance shall be position type data only. The joint type data is not handled.

The table below lists the available positions:

Position No	Score	Description	Remarks
1 - 999	999	Position type data in robot controller	
5000	1	Robot's current position	State variable P_Curr is supported
5100 - 5102	3	Position type data specified by the sequencer in the CPU buffer memory	

● How to use positions 5100 - 5102

Use these positions when performing sequencer direct with the position data created in the sequencer. Sequencer direct can be performed by storing the position data created in the sequencer in the position type data area (5100 - 5102) in the sequencer CPU buffer memory and specifying the position number 5100 - 5102.

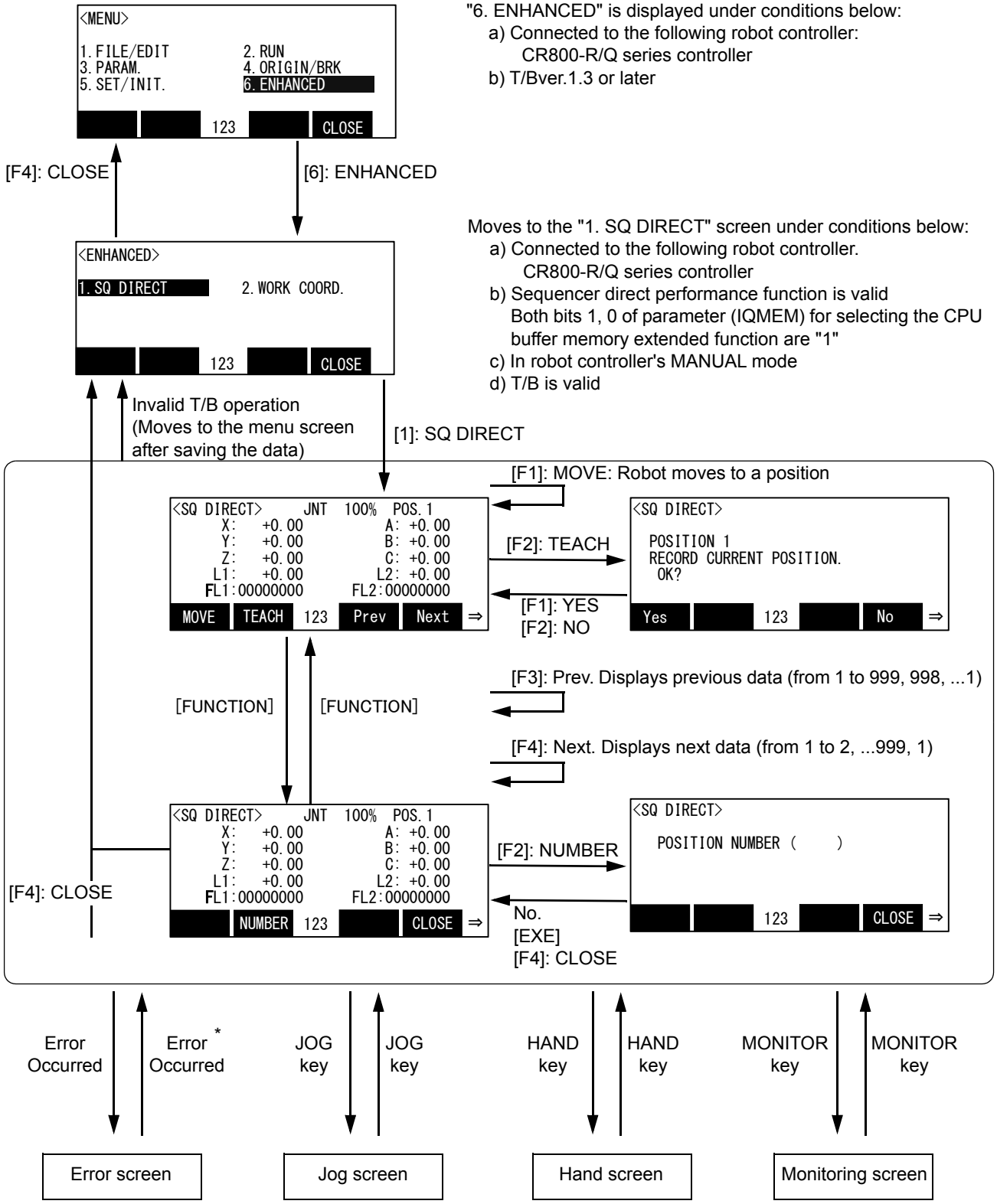
● Specifying the tool data

Before teaching, specify the tool data. The tool data specifies the control point of hand or tool mounted on the robot. For more information, refer to Page 87, "(8) Tool data setting"

Note that when tool data is specified or changed after teaching, previous teaching data will be unavailable. (When you try to move the robot to previous teaching position before setting or changing of tool data, the robot moves to the wrong position.)

(2) Position Teaching in Position Box (R32TB)

1) Screen change



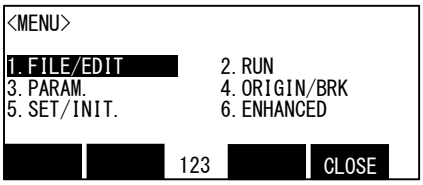
"6. ENHANCED" is displayed under conditions below:
- a) Connected to the following robot controller:
  CR800-R/Q series controller
- b) T/Bver.1.3 or later

Moves to the "1. SQ DIRECT" screen under conditions below:
- a) Connected to the following robot controller.
  CR800-R/Q series controller
- b) Sequencer direct performance function is valid
  Both bits 1, 0 of parameter (IQMEM) for selecting the CPU buffer memory extended function are "1"
- c) In robot controller's MANUAL mode
- d) T/B is valid

* When an error occurs while displaying the teach confirmation screen, the screen returns to the SQ DIRECT teach screen by resetting the error.

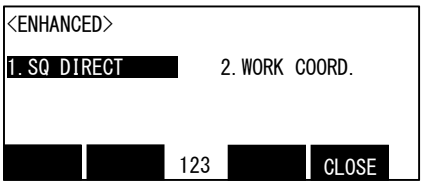
2) Description of screens

2-1) Menu screen



"6. ENHANCED" is displayed as a menu item. It is always possible to move to the ENHANCED menu screen.

2-2) ENHANCED function menu screen



["1. SQ DIRECT" display]

Although the sequencer direct function is valid or not, "1. SQ DIRECT " is displayed.

[Selecting "1. SQ DIRECT "]

When "1. SQ DIRECT " is selected, the sequencer direct teach screen is displayed.

<Condition>

All the conditions below should be met:

a) Connected to the following robot controller
CR800-R/Q series controller

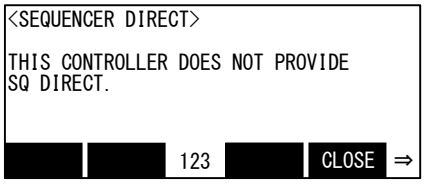
b) Sequencer direct performance function is valid
Both bits 1, 0 of parameter (IQMEM) for selecting the CPU buffer memory extended function are "1"

c) In robot controller's MANUAL mode

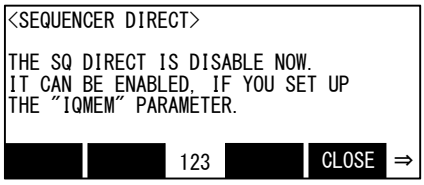
d) T/B is valid

<Action when screen change is impossible>

When above conditions a, b are not met, the warning screen below is displayed.



Warning screen is displayed when above condition "a)" is not met



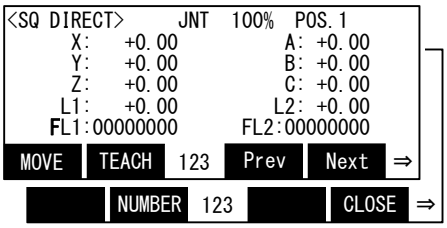
Warning screen is displayed when above condition "b)" is not met

- When [F4] (CLOSE) is selected or [RESET] key is pushed, comes back to the ENHANCED menu screen.
- When an error occurs, changes to the error screen. The screen returns to the ENHANCED menu screen by resetting the error.
- Push [JOG] key to change to the JOG screen. Push [JOG] key again to change to the ENHANCED menu screen.
- Push [HAND] key to change to the hand screen. Push [HAND] key again to change to the ENHANCED menu screen.
- Push [MONITOR] key to change to the monitoring screen. Push [MONITOR] key again to change to the ENHANCED menu screen.

● Disable T/B to change to the ENHANCED menu screen.
When the warning screen above is displayed while an error occurs, push [RESET] key to reset the error and then change to the ENHANCED menu screen.

When the conditions "c)", "d)" above are not met while the conditions "a)", "b)" above are met, the buttons are grayed out. The warning screen is not displayed.

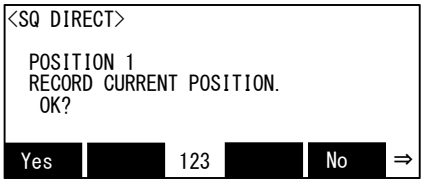
2-3) SQ DIRECT teach screen



For information on teaching operation, refer to the description about position edit screen in "Instruction Manual, Detailed Description of Functions and Operations."

[Position check]
The robot moves to the displayed position while pushing [F1] (MOVE) key.
<Condition>
All the conditions below should be met:
a) In robot controller's MANUAL mode
b) T/B is valid
c) T/B enabled switch (3-positioned switch) is turned ON (intermediate position)
d) Servo is turned ON

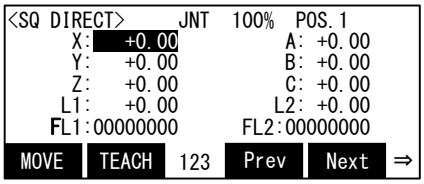
[Teaching position]
Push [F2] (TEACH) key to display the confirmation screen below.



Push [F1] (Yes) key in the confirmation screen to teach you that current position is at the displayed position and come back to the screen.
Push [F4] (No) key to come back to the screen without teaching.

[Teaching position]
Teaching in the SQ DIRECT teach screen always sets up current position for mecha 1.
Even when mecha 2 or 3 is selected by T/B, it sets up current position for mecha 1.

[MDI (Manual Data Input) registration/ modification of position]
A position can be registered by directly inputting numeric value to each axis's component of position data.
Push arrow key to move the cursor to the data to be modified, input numeric value, and push [EXE] key.



[Changing position display]

(a) Forward/ backward feed

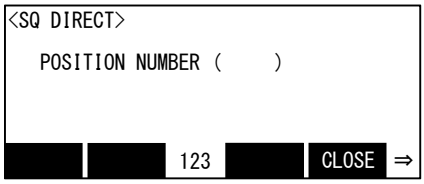
Each time [F3] (Prev) or [F4] (Next) key is pushed, displayed position is changed.
The display changes as follows:

[F3] (Prev): From 1 to 999, 998, 997, ..., 1

[F4] (Next): From 1 to 2, 3, ..., 999, 1

(b) Call number

Push [FUNCTION] key, change the function key allocation, and push [F2] (NUMBER) key to display the position number input screen below.



Input a position number and push [EXE] key to come back to the screen and display the target position.
When a number other than 1 to 999 is entered, the [EXE] key gets unavailable.
Push [F4] (CLOSE) key to come back to the previous screen.

[Displaying menu screen]

Push [FUNCTION] key, change the function key allocation, and push [F4] (CLOSE) key to come back to the ENHANCED menu screen.

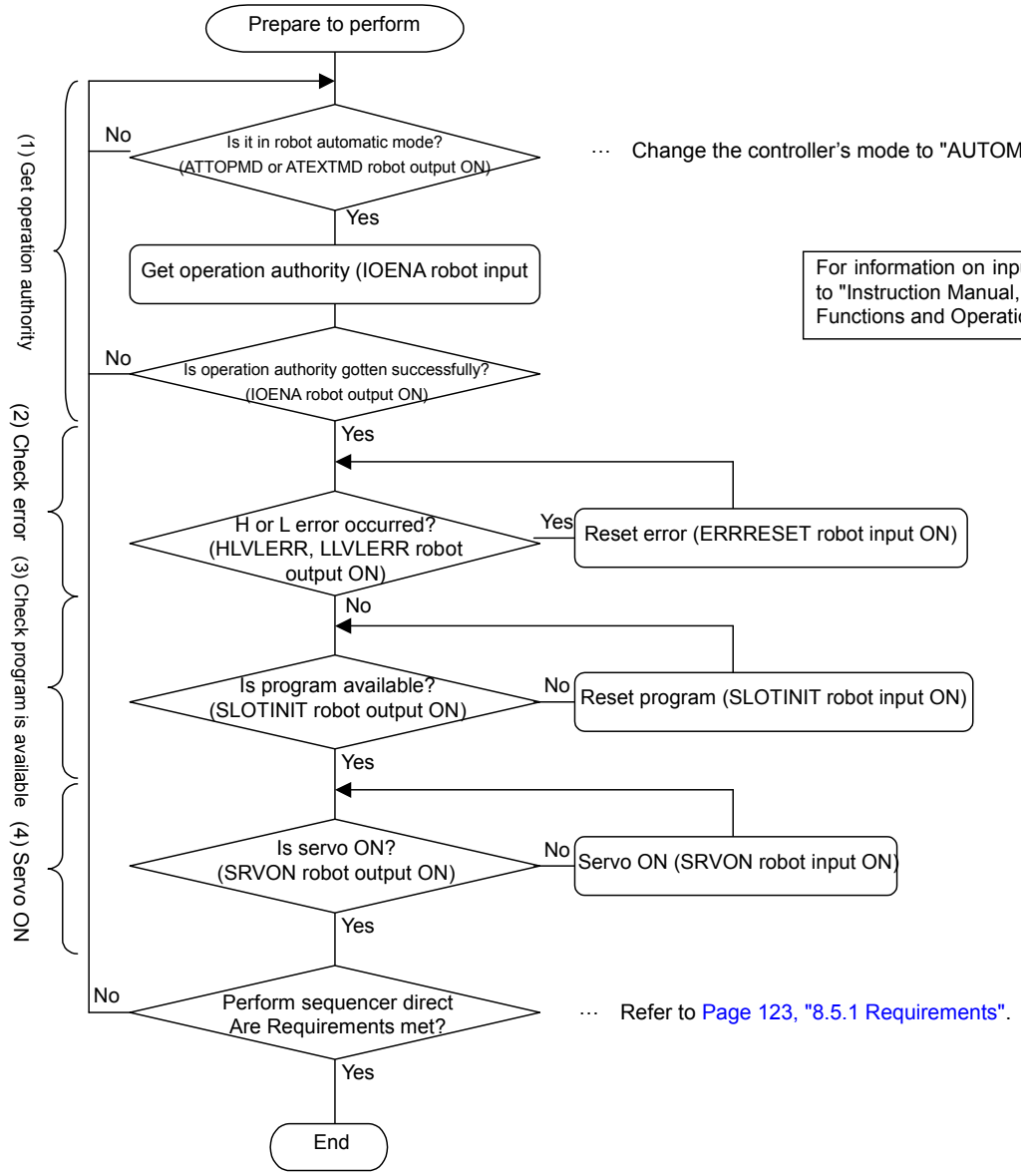
[Handling when T/B became invalid]

When T/B became invalid while displaying the sequencer direct teach screen (including the teach confirmation and position number input screens), saves the position data and comes back to the ENHANCED menu screen.

[Handling when entered into AUTOMATIC mode]

When changed to AUTOMATIC mode while displaying the sequencer direct teach screen (including the teach confirmation and position number input screens), an error "H5000 TB Enable key is ON " occurs and an error screen appears. The screen comes back to the previous screen by resetting the error.

8.2.3 Prepare to Perform Sequencer Direct



(1) In the sequencer, get the robot's external operation authority.
   1) Change the controller's mode to "AUTOMATIC".
   2) Turn ON the sequencer's operation authority signal (IOENA).
   3) Wait until the robot successfully gets the operation authority (robot's operation authority (IOENA) signal is ON).

(2) In the sequencer, check that H or L level error is not occurring in the robot.
   Check the robot's error state (robot's HLVLERR and LLVLERR signals OFF). When either one is turned ON, turn ON sequencer's ERRRESET signal to reset the error.

(3) In the sequencer, check that robot's program is available (program is not running).
   Check the robot's program availability (robot's SLOTINIT signal ON). When it is not turned ON, turn ON sequencer's SLOTINIT signal to reset the error.

(4) In the sequencer, turn ON the robot's servo.
   Check the robot's servo ON (robot's SRVON signal ON). When it is not turned ON, turn ON sequencer's SRVON signal and wait until the robot turns ON the servo.

* The order of steps (1) to (4) are not necessarily the same as above. But, in the sequencer, to reset a program and turn ON the servo, it is necessary to get external operation authority.

8.3 How to Operate Sequencer Direct

Here, describes the robot's operation commands and how to control hand.
To issue an operation instruction to the robot, set up the command data (command number + auxiliary data) and command condition data and turn ON the command request signal. The robot runs according to the specified command. Control the hand by turning ON/OFF the hand output signal.

8.3.1 Operation Command

Memory map of sequencer direct performance area corresponding to the robot operation commands is as follows:

(1) Sequencer output

Sequencer Output Addr (offset)	Description		Remarks
520	Bit signal	Command request signal Bit allocation bit15 0 0000000000000000 +- bit0: Command request	
521		(Reserved)	
522	Command data	(Reserved)	
523		(Reserved)	
524		Command No	
525		Command data 1	
526		Command data 2	
527	(Reserved)		
528	(Reserved)		
529	(Reserved)		
530	Command condition data	Override [%: 1 - 100, 0]	100% when zero
531		Acceleration rate [%: 1 - 100, 0]	100% when zero
532		Deceleration rate [%: 1 - 100, 0]	100% when zero
533		(Reserved)	
534		(Reserved)	
535		Speed setting [mm/s: 1 - 10000, 0]	When either 0 or 10000, it operates at maximum speed.
536		Shortcut/roundabout specification [0: Initial value/ 1: Opposite of initial value]	<Joint interpolation> 0: Roundabout (teaching posture) 1: Shortcut <Linear/ circular interpolation> 0: Shortcut /1: Roundabout
537		Auxiliary operation specification [0: Equivalent rotation/1: Orthogonal triaxis/2: Singularity pass]	Valid for linear/ circular interpolation
538		Tool setting [0: Current tool/ 1- 4: Tool number]	
539 - 639	(Reserved)		

Sequencer Output Addr (offset)	Description		Remarks
640	Position data 1 (5100)	X coordinate value [10^{-4}mm/10^{-4}deg]	
641			
642		Y coordinate value [10^{-4}mm/10^{-4}deg]	
643			
644		Z coordinate value [10^{-4}mm/10^{-4}deg]	
645			
646		A coordinate value [10^{-4}mm/10^{-4}deg]	
647			
648		B coordinate value [10^{-4}mm/10^{-4}deg]	
649			
650		C coordinate value [10^{-4}mm/10^{-4}deg]	
651			
652		L1 coordinate value [10^{-4}mm/10^{-4}deg]	
653			
654		L2 coordinate value [10^{-4}mm/10^{-4}deg]	
655			
656		Structure flag	
657			
658		Multi-turn data	
659			
660	Position data 2 (5101)	X coordinate value [10^{-4}mm/10^{-4}deg]	
661			
662		Y coordinate value [10^{-4}mm/10^{-4}deg]	
663			
664		Z coordinate value [10^{-4}mm/10^{-4}deg]	
665			
666		A coordinate value [10^{-4}mm/10^{-4}deg]	
667			
668		B coordinate value [10^{-4}mm/10^{-4}deg]	
669			
670		C coordinate value [10^{-4}mm/10^{-4}deg]	
671			
672		L1 coordinate value [10^{-4}mm/10^{-4}deg]	
673			
674		L2 coordinate value [10^{-4}mm/10^{-4}deg]	
675			
676		Structure flag	
677			
678		Multi-turn data	
679			
680	Position data 3 (5102)	X coordinate value [10^{-4}mm/10^{-4}deg]	
681			
682		Y coordinate value [10^{-4}mm/10^{-4}deg]	
683			
684		Z coordinate value [10^{-4}mm/10^{-4}deg]	
685			
686		A coordinate value [10^{-4}mm/10^{-4}deg]	
687			
688		B coordinate value [10^{-4}mm/10^{-4}deg]	
689			
690		C coordinate value [10^{-4}mm/10^{-4}deg]	
691			
692		L1 coordinate value [10^{-4}mm/10^{-4}deg]	
693			
694		L2 coordinate value [10^{-4}mm/10^{-4}deg]	
695			
696		Structure flag	
697			
698		Multi-turn data	
699			

(2) Robot output

Sequencer Input Addr (offset)	Description		Remarks
520	Bit signal	Command completion signal bit15 0 0000000000000000 |+- bit0: Completed +-- bit1: Working	
521		(Reserved)	
522	Command data	Completion status [1: OK/ other than 1: NG]	
523		(Reserved)	
524		Command No	
525		Command data 1	
526		Command data 2	
527 - 549	(Reserved)		

(3) Data description

[Command data]

● Command number (524), command data 1 (525), command data 2 (526)
Instruction data (sequencer output 524 - 526) to the robot for sequencer direct performance and returned data from the robot (robot output 524 - 526). The setting values according to the instructed operations are as follows:

Action		Symbol Note1)	Command No	Command data 1	Command data 2	Remarks
Joint interpolation	Moves to the specified position. Specify the Z direction tool distance to move by specified distance away to the Z axis direction in the tool coordinate system.	Mov	1	Destination Position No	Z direction tool distance [10^{-1}mm]	When the setting value of data 2 is set to zero, moves to the specified position.
	Moves the robot to the position with coordinate value of the destination position added by the coordinate of the approach coordinate position.	MovA	2	Destination position number	Approach coordinate position number	
Linear interpolation	Moves to the specified position. Specify the Z direction tool distance to move by specified distance away to the Z axis direction in the tool coordinate system.	Mvs	11	Destination position number	Z direction tool distance [10^{-1}mm]	When the setting value of data 2 is set to zero, moves to the specified position.
	Moves the robot to the position with coordinate value of the destination position added by the coordinate of the approach coordinate position.	MvsA	12	Destination position number	Approach coordinate position number	

Note1) These symbols MovA, MvsA are different from the robot language. They are used for description below.

● Completion status (sequencer input 522)

When the sequencer direct is successfully performed or when the sequencer direct cannot be received, the completion status is set.

The values below are established as completion status:

Setting Value	Description	Remarks
1	Successfully completed	
2	External operation authority invalid	The command cannot be received
3	H or L level error is occurring	
4	Program is not available (program running)	
5	Not robot servo ON	
6	Stop signal inputting	
7	Returning to retracting point	
8	Remote Jog working	
9	Variable not extended (For more information, refer to Page 133, "10.1 Parameter of Selecting CPU buffer Memory Extended Function")	
10	Origin not set	
11	Command number out of range	
12	Command data 1 out of range	
13	Command data 2 out of range	
14	Operating condition data out of range (Only the available operating conditions for the target operation are checked)	
20	Sequencer direct impracticable because of other causes	
30	Sequencer direct performance suspended	Command suspended

[Command condition data] (sequencer output 530 - 538)

The table below lists the data specified as command condition data:

Name	Description	Setup range	Operation for Initial Value (0)	Corresponding Command MELFA-BASIC
Override	Specify the speed rate [%] of robot operation [1 - 100, 0]	1 - 100, 0 (100% when zero)	100%	Ovrd
Acceleration rate	Specify the acceleration rate [%] of robot operation [1 - 100, 0]	1 - 100, 0 (100% when zero)	100&	Accel
Deceleration rate	Specify the deceleration rate [%] of robot operation	1 - 100, 0 (100% when zero)	100%	
Speed setting	Specify the speed [mm/s] of robot's linear interpolation	1 - 10000, 0 (When either 0 or 10000, it operates at maximum speed)	Maximum speed	Spd
Shortcut/roundabout specification Note1)	Specify the robot's shortcut/roundabout [0: Initial value/ 1: Opposite of initial value] <Joint interpolation> 0: Roundabout (teaching posture) /1: Shortcut <Linear interpolation> 0: Shortcut /1: Roundabout	Refer to the left	Joint interpolation → Roundabout Linear interpolation → Shortcut	Type specification of operation command
Auxiliary operation specification	Auxiliary specification for robot's linear interpolation	0: Equivalent rotation 1: Orthogonal triaxis 2: Singularity pass	Equivalent rotation	
Tool setting	Sets the tool number. Tool data (MEXTL 1 - 4) with specified number is used as the current tool data and is set to parameter MEXTL.	0: Current tool 1 - 4: Tool number	Current tool	M_Tool

Note1) Shortcut/roundabout specification value

When the shortcut/roundabout specification value is zero, it specifies the initial value (without Type specification) of the robot program commands (Mov, Mvs). When it is one, it specifies reverse initial value. They are different from the value set up by Type specification of robot program command.

[Position data 1 - 3] (sequencer output 640 - 699)

Used during setting up the position data in the sequencer when performing the sequencer direct.

The unit is 10^{-4}mm or 10^{-4}deg.

Only lower one word is used for the structure flag of position variable, and upper one word is a reserved area.

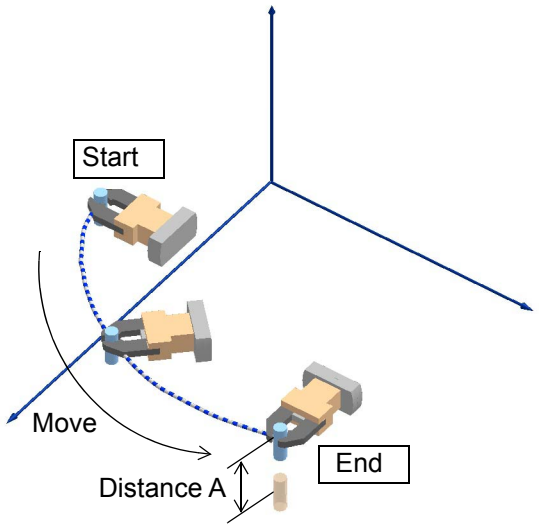
[Command/ command condition description]

(1) Joint interpolation: Mov, MovA

Evenly interpolate the robot's each axis difference between joint angles of start and end positions. Therefore, end's track draws a smooth arc.

MovA moves the robot to the position with coordinate values of the destination position added by the coordinate values of the approach coordinate position.

For example, when Z value of approach coordinate position is the distance A shown in the figure right below, joint interpolates to position A over the aimed position.



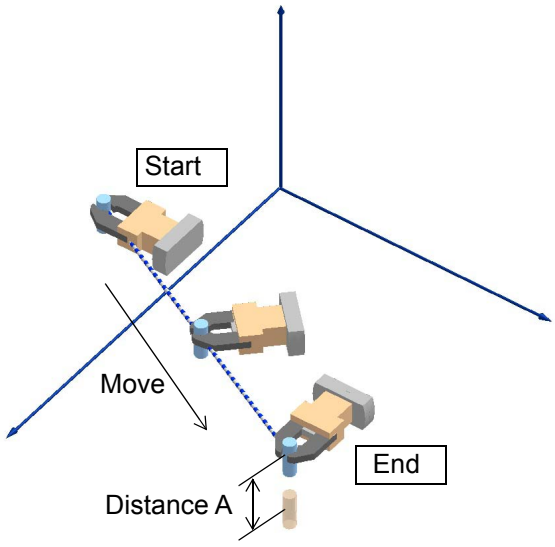
(2) Linear interpolation movement: Mvs, MvsA

Linear interpolation moves the robot so that the track of control points from robot's start to the end becomes a straight line.

The hand's posture changes evenly from the start to the end.

MvsA moves the robot to the position with coordinate values of the destination position added by the coordinate values of the approach coordinate position.

For example, when Z value of approach coordinate position is the distance A shown in the figure right above, linear interpolates to position A over the aimed position.



(3) Override: Ovrd

Specify the speed of robot operation with the value between 1 and 100%.

(4) Acceleration/ deceleration rate: Accel

Specify the acceleration and deceleration in rate (%) during robot operation.

Specify the acceleration/deceleration rate with the value between 1 and 100% with reference to the acceleration and deceleration time set up for robot in advance. The initial value is 100% (maximum acceleration and deceleration) for both acceleration and deceleration. Adjust the acceleration/deceleration rate according to the robot activity.

(5) Speed: Spd

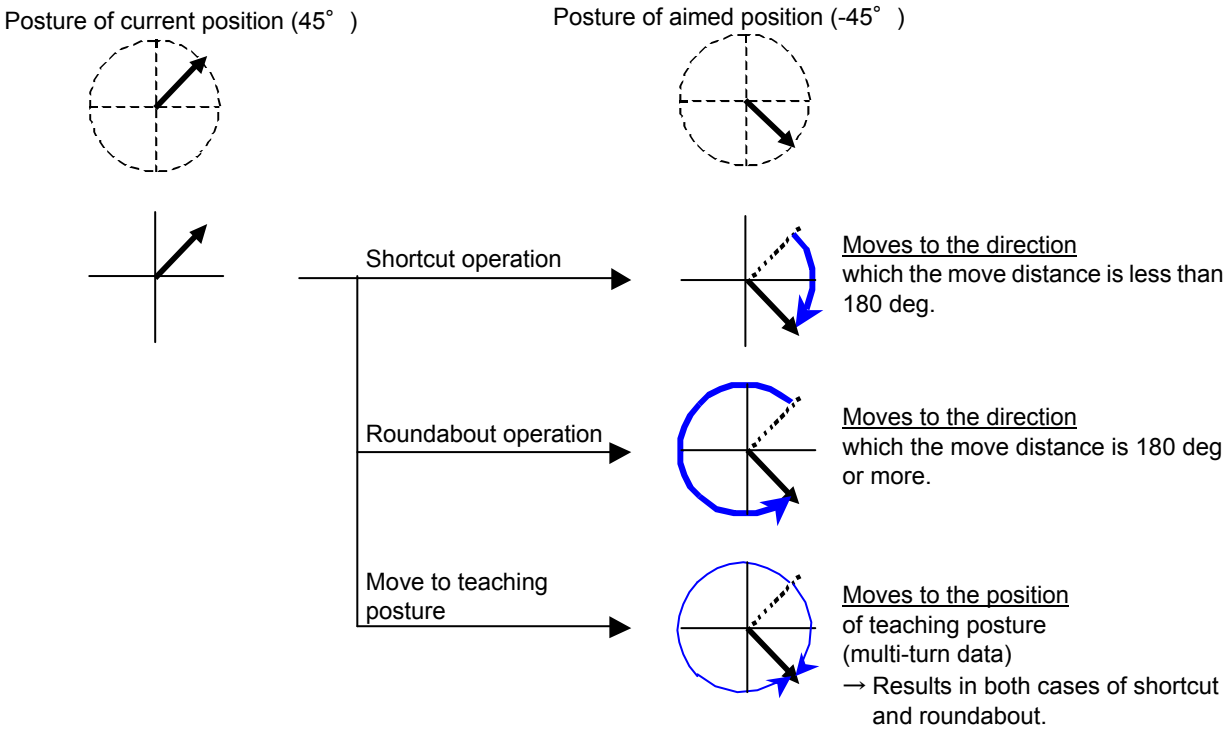
Specify the speed of the end when the robot moves for linear interpolation. The unit is mm/s.
This value does not impact on the joint interpolation command.
When zero or 10000mm/s is specified as the speed, the robot is in the maximum speed control mode.
The maximum speed control mode allows you to reduce the takt time by adjusting the motor speed of robot's each axis while keeping linear track. Consequently, linear speed may change.
Actual linear speed is a value specified by override command multiplied by a speed specified by this speed setting command.

Example: When override = 50% and speed setting = 300mm/s, actual linear speed = 0.5 x 300 = 150mm/s

(6) Shortcut/roundabout specification

There are three types of hand rotation direction below when performing a move command:

a) Shortcut specification
b) Roundabout specification
c) Move to teaching posture (Roundabout joint interpolation)



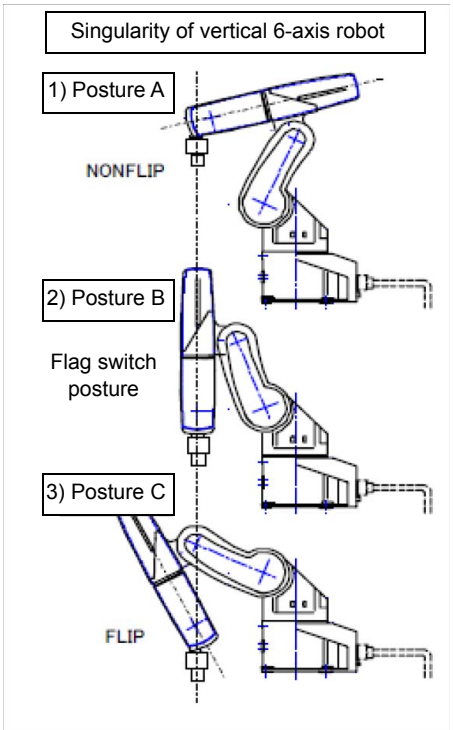
(7) Auxiliary operation specification

Specify the hand posture control type during linear interpolation.

a) Equivalent rotation: Evenly interpolates from start posture (A, B, C) to the posture (A, B, C) at aimed position.

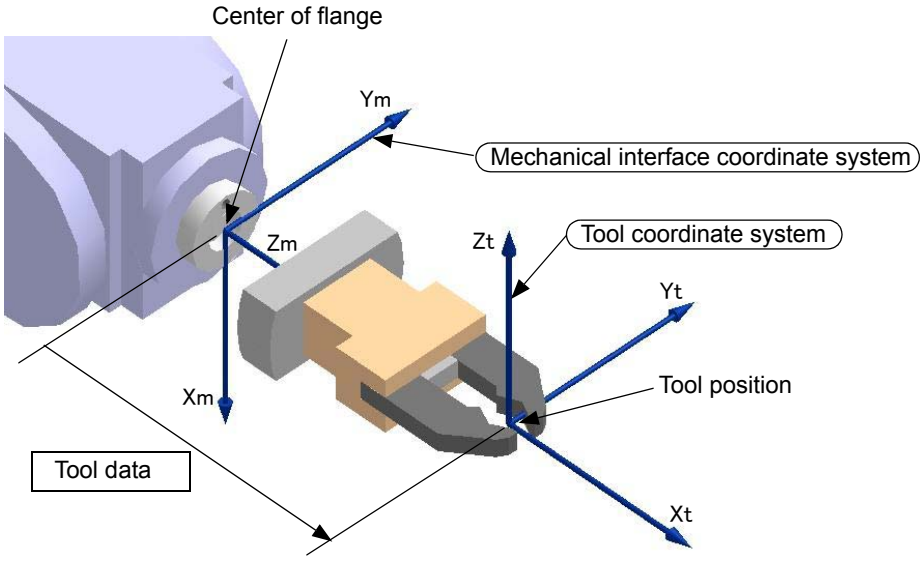
b) Orthogonal triaxis: Interpolates with joint angle (J4, J5, J6) instead of hand posture (A, B, C). Evenly interpolates from start posture (J4, J5, J6) to the posture (J4, J5, J6) at aimed position.
Effective when passing by near a singularity.

c) Singularity pass: Specification to pass by a singularity specific to six axes robot (singularity posture is posture B shown in the right figure (2)).
Restricted by some positions and postures.
For more information on the operation, refer to "Instruction Manual, Detailed Description of Functions and Operations".



(8) Tool data setting

Select the tool data (1 - 4) set up by parameters in advance.
The tool data indicates the end (grip point) of hand and is specified by shift amount from the center of robot flange and rotation angle.



[How to decide tool data]

The tool data has the same components as the position data.

X, Y, Z: Shift amount. Moving amount from the center of flange to the tool position. (Unit is mm)

A, B, C: Rotation angles of coordinate axes. (Unit is deg)

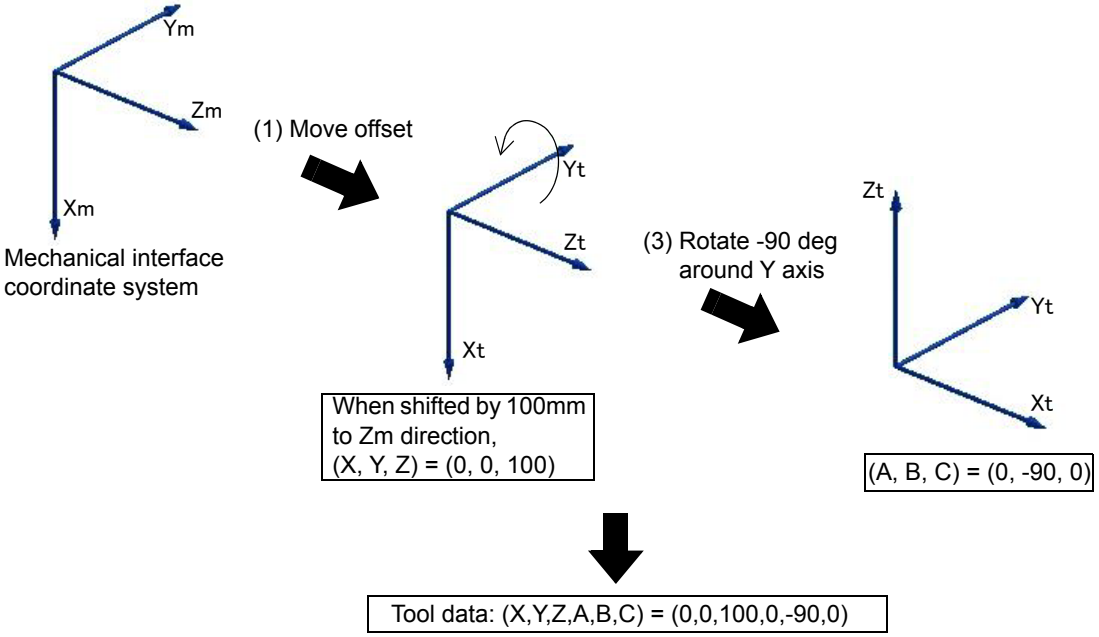
A: Rotation angle around X axis

B: Rotation angle around Y axis

C: Rotation angle around Z axis

To decide each data, move the mechanical interface coordinate system at the center of flange in order of (1) shift amount, (2) Z axis rotation, (3) Y axis rotation, (4) X axis rotation to accord with the aimed tool coordinate system. In this case, the move amounts (1) - (4) (rotation amounts) indicates the tool data.

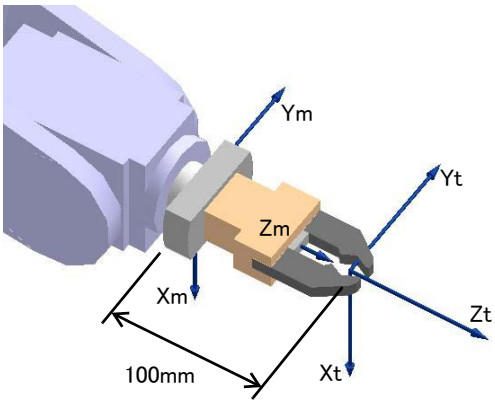
Based on the example shown in the figure above, the figure below shows the steps to decide each data. In this example, because steps 2 and 4 of steps (1) - (4) are not necessary, the steps 2 and 4 are omitted.

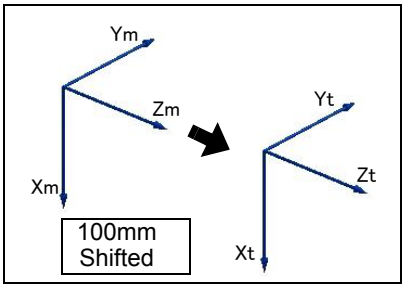


[Tool data setting example]

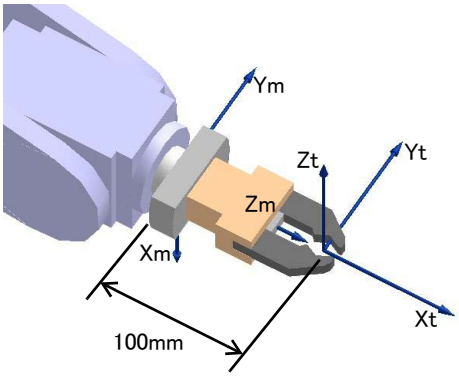
A sample hand attachment and sample tool data setting in the coordinate system are shown below:

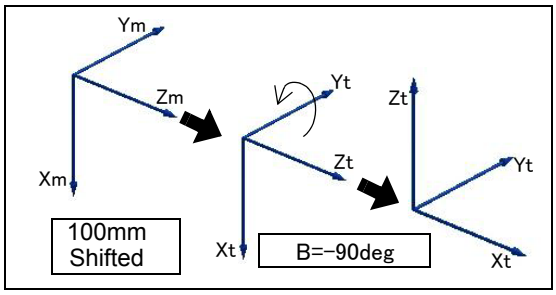
(1) Mechanical interface coordinate system is shifted as a whole:
Tool data: (0,0,100,0,0,0)



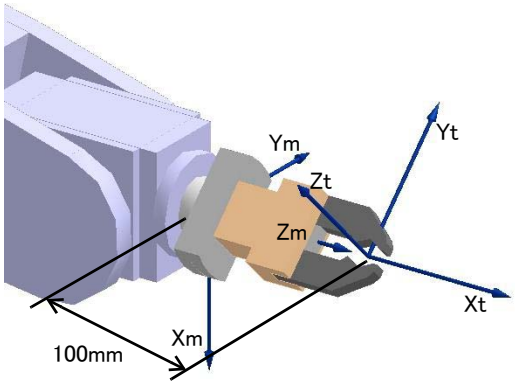


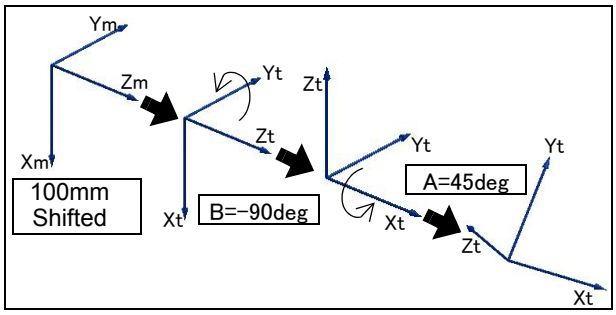
(2) Zt direction becomes perpendicular to mechanical interface coordinate system:
Tool data: (0,0,100,0,-90,0)



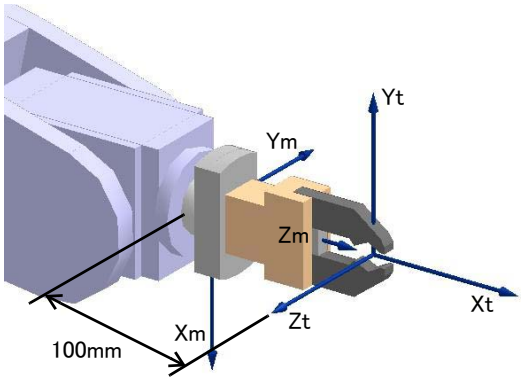


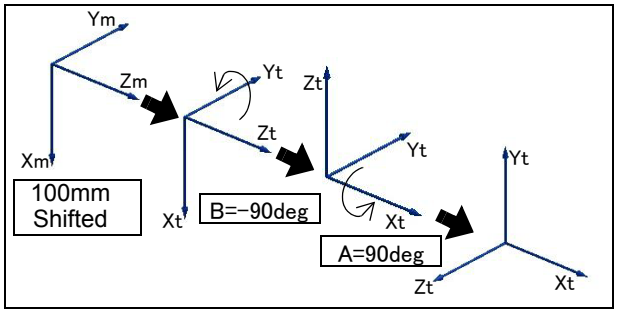
(3) Hand is tilted by 45deg to the Zm axis of mechanical interface coordinate system:
Tool data: (0,0,100,45,-90,0)





(4) Hand is tilted by 90deg to the Zm axis of mechanical interface coordinate system:
Tool data: (0,0,100,90,-90,0)

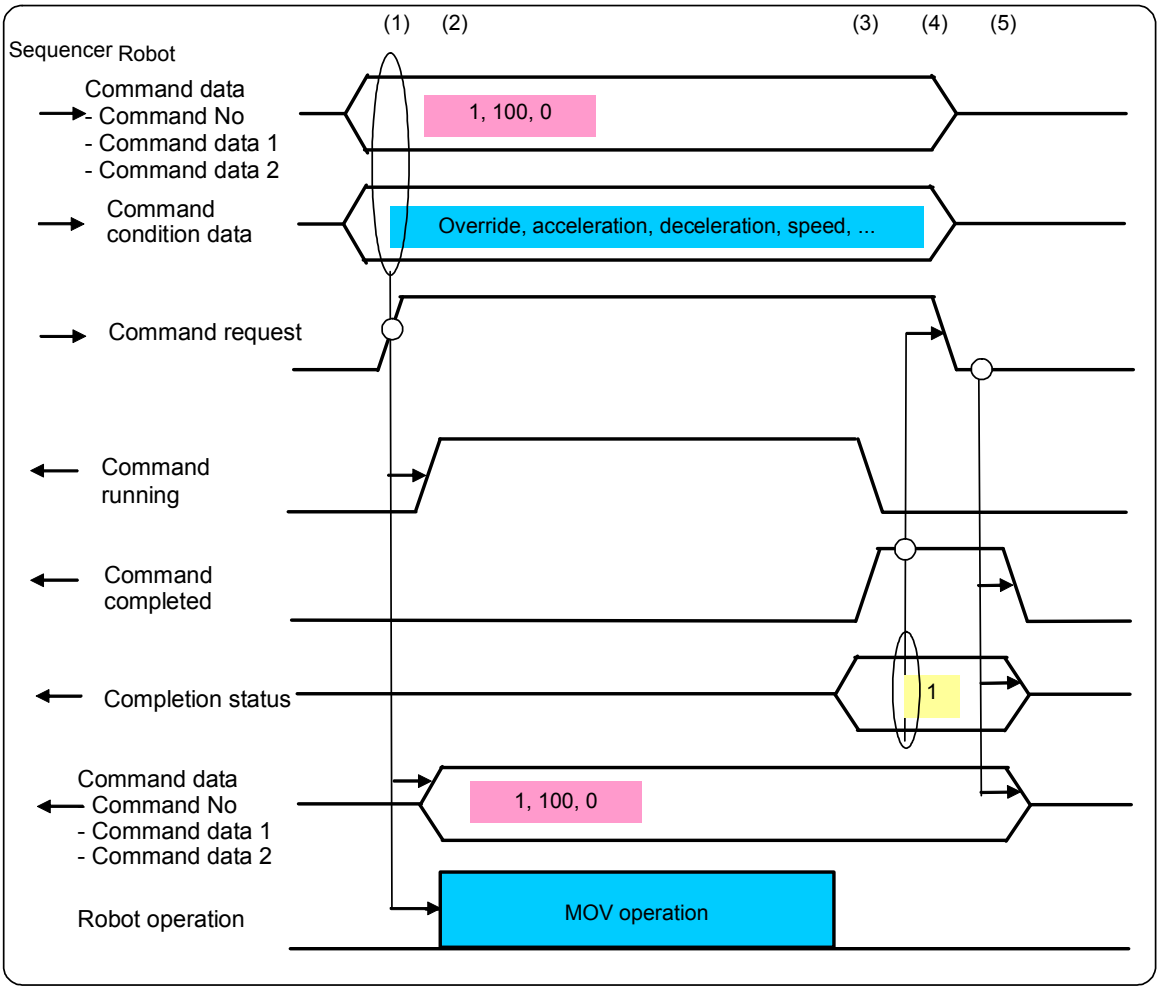




8.3.2 Timing Chart for Performing Operation Command

(1) Perform Operation (Normal Operation)

The sequencer operates the robot by setting the data for command number and command data 1, 2.
When the command condition data is set to zero, the robot runs based on the default setting.
The robot runs based on the instructed setting by setting value for condition data.

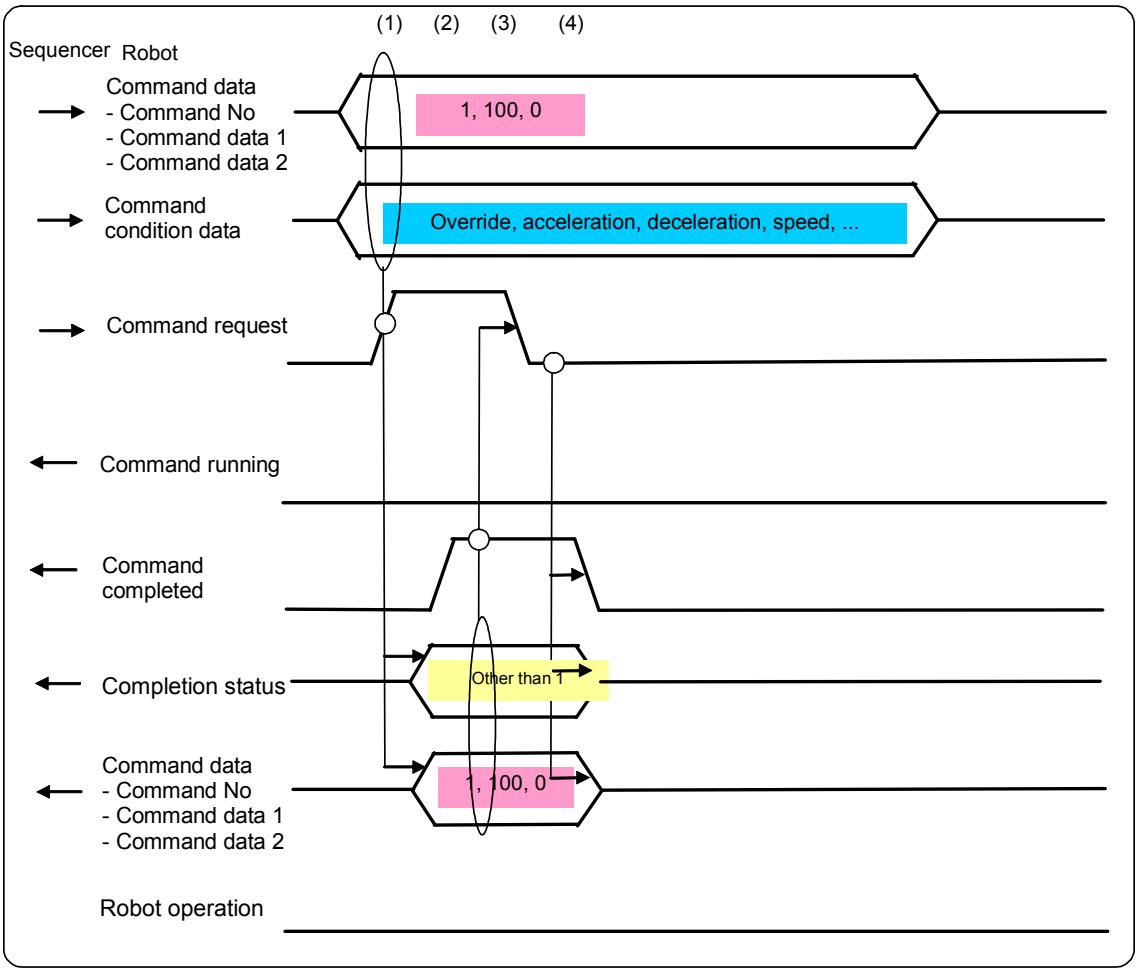


(1) The sequencer sets up "Command data" and "Condition data" and sends "Command request ON".
(2) When "Command request ON" is received, the robot imports "Command data" and "Condition data". When the imported data is formal and sequencer direct is practicable, the robot sets up "Command data" (returned data), sends "Command running ON", and carries out the instructed robot operation.
(3) When the robot successfully completed the operation, the robot sends "Command running OFF", sets "Completion status" to one, and sends "Command completed ON".
(4) When the sequencer received "Command completed ON", the sequencer imports "Completion status" and sends "Command request OFF".
(5) When the robot received "Command request OFF", the robot clears "Completion status" and "Command data" to zero and sends "Command completed OFF".

Robot's arrival point when operation command is completed
"Command completed" signal for performing an operation command is turned ON when the robot arrived at the aimed position (encoder feedback position is not checked).
Therefore, when performing operations continuously, the robot may perform next operation before arriving at the aimed position. In order to avoid this situation, make sure that the sequencer takes a delay time before carrying out next operation after sequencer direct was completed.

(2) Operation Command Is Impracticable:

When the command data instructed by the sequencer is not formal or when the robot is out of work, the robot stores a number indicating an impracticable reason in "Completion status" against "Command request" from the sequencer and then returns "Command completed".



(1) The sequencer sets up "Command data" and "Condition data" and sends "Command request ON."

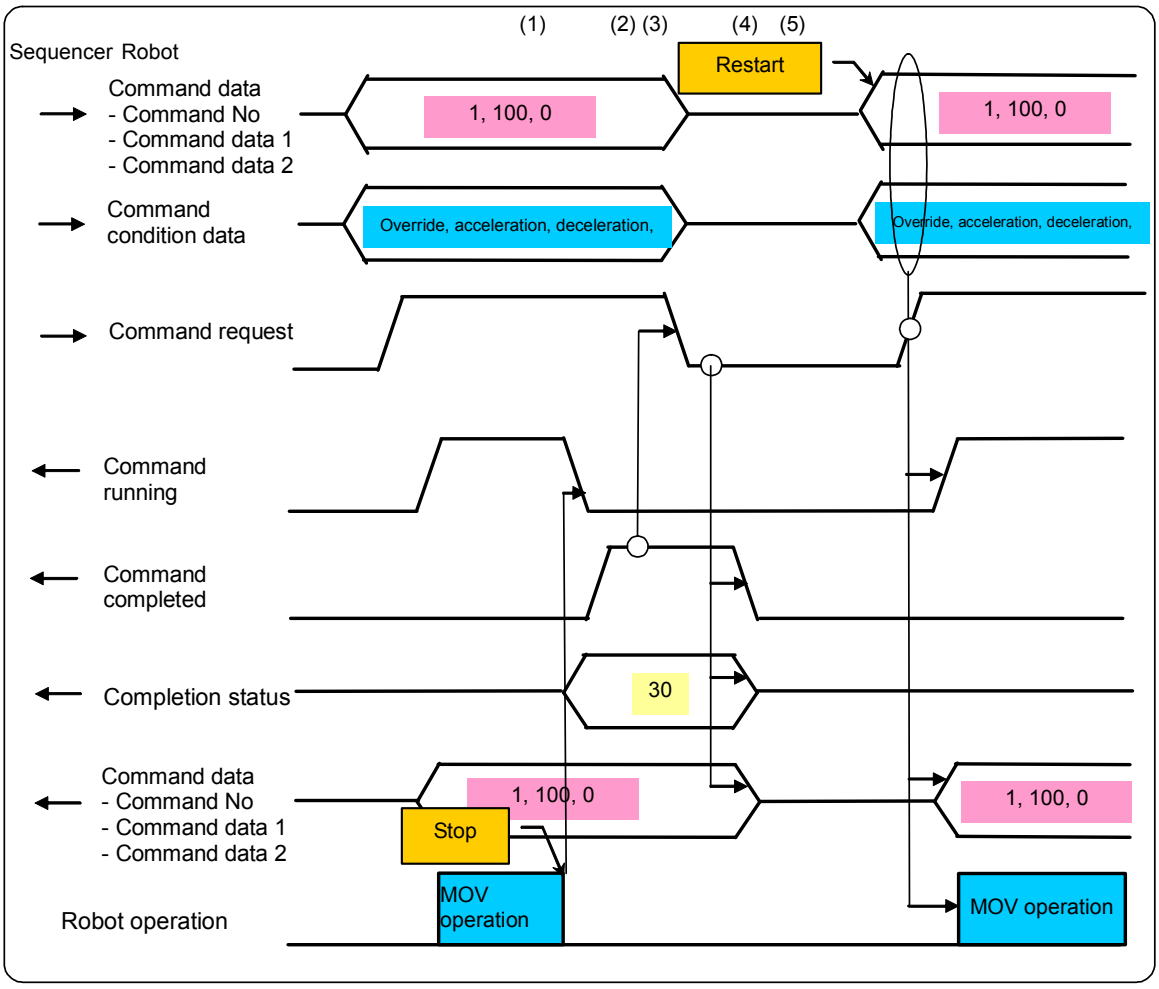
(2) When "Command request ON from OFF" is received, the robot imports "Command data" and "Condition data." When the imported data is not official or when the robot is impracticable of sequencer direct, the robot sets "Command data" (returned data) and "Completion status" to other than one, and sends "Command completed ON."

(3) When the sequencer receives "Command completed ON", the sequencer imports "Completion status" and sends "Command request OFF".

(4) When the robot receives "Command request OFF", the robot clears "Completion status" and "Command data" and sends "Command completed OFF".

(3) Suspend/Resume Operation

When the robot stops due to the robot's stop operation or stop input while performing the sequencer direct, the operation is suspended and the command is also suspended (Completion status = 30, suspended).
To resume after suspension, set up "Command data" and "Condition data" again and send "Command request" signal again.



<Suspension handling>

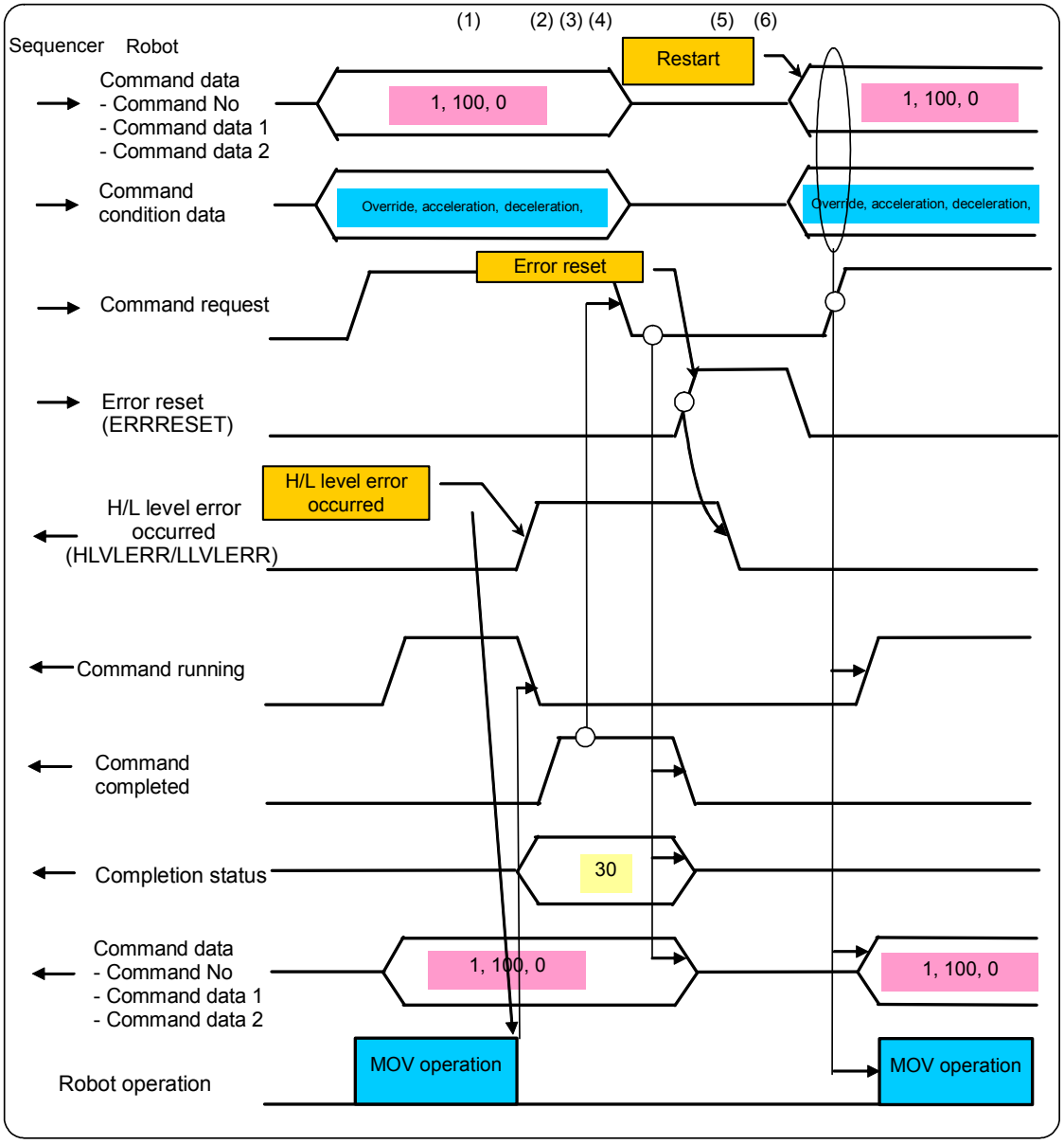
(1) When the robot stops while performing the sequencer direct, the robot sends "Command running OFF", sets "Completion status" to 30 (suspended), and sends "Command completed ON".
(2) When the sequencer receives "Command completed ON", the sequencer imports "Completion status" and sends "Command request OFF".
(3) When the robot receives "Command request OFF", the robot clears "Completion status" and "Command data" to zero and sends "Command completed OFF".

<Resume handling>

(4) When a resume operation is carried out in the sequencer, the sequencer sets up "Command data" and "Condition data" and sends "Command request ON".
(5) When "Command request ON" from OFF is received, the robot imports "Command data" and "Condition data", sets up "Command data" (returned data), sends "Command running ON", and carries out the instructed robot operation.

(4) Support on Occurrence of Error

When H or L level error occurs while performing the sequencer direct, the operation is suspended.
To resume after suspension, reset the error, re-set up "Command", "Command data", and "Condition data", and send "Command request" signal again (I/F for suspension and resume is the same as Page 92, "(3) Suspend/Resume Operation").



<Handling on error occurrence>

(1) When H or L level error occurs in the robot while performing the sequencer direct, the operation is suspended. The robot sends "Command running OFF", sets "Completion status" to 30 (suspended), and sends "Command completed ON".

(2) When the sequencer receives "Command completed ON", the sequencer imports "Completion status" and sends "Command request OFF".

(3) When the robot receives "Command request OFF", the robot clears "Completion status" and "Command data" to zero and sends "Command completed OFF".

<Error reset handling>

(4) Error reset operation clears the robot error.

<Resume handling>

(5) When a resume operation is carried out in the sequencer, the sequencer sets up "Command data" and "Condition data" and sends "Command request ON".

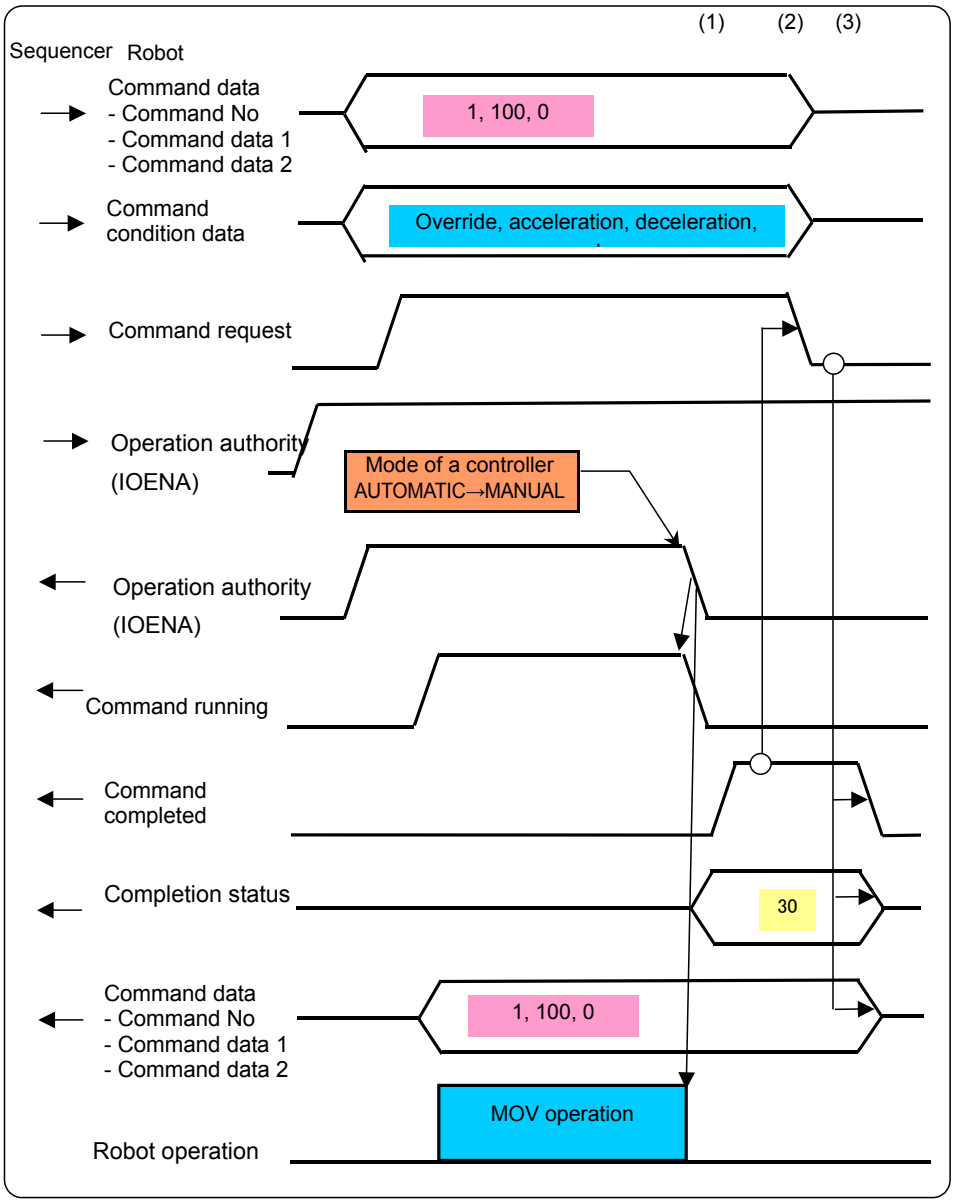
(6) When "Command request ON" from OFF is received, the robot imports "Command data" and "Condition data", sets up "Command data" (returned data), sends "Command running ON", and carries out the instructed robot operation.

(5) Suspension when Robot's External Operation Authority Gets Invalid

When the robot's external operation authority gets invalid while performing the sequencer direct (robot's dedicated signal - operation authority output (IOENA) is turned OFF), the operation is suspended.

The conditions which make the robot's external operation authority invalid are as follows:

1) The MODE switch is changed from AUTOMATIC to MANUAL on the robot operation panel (robot is turned servo OFF)
2) The sequencer set the robot's operation authority signal (IOENA) to "OFF"



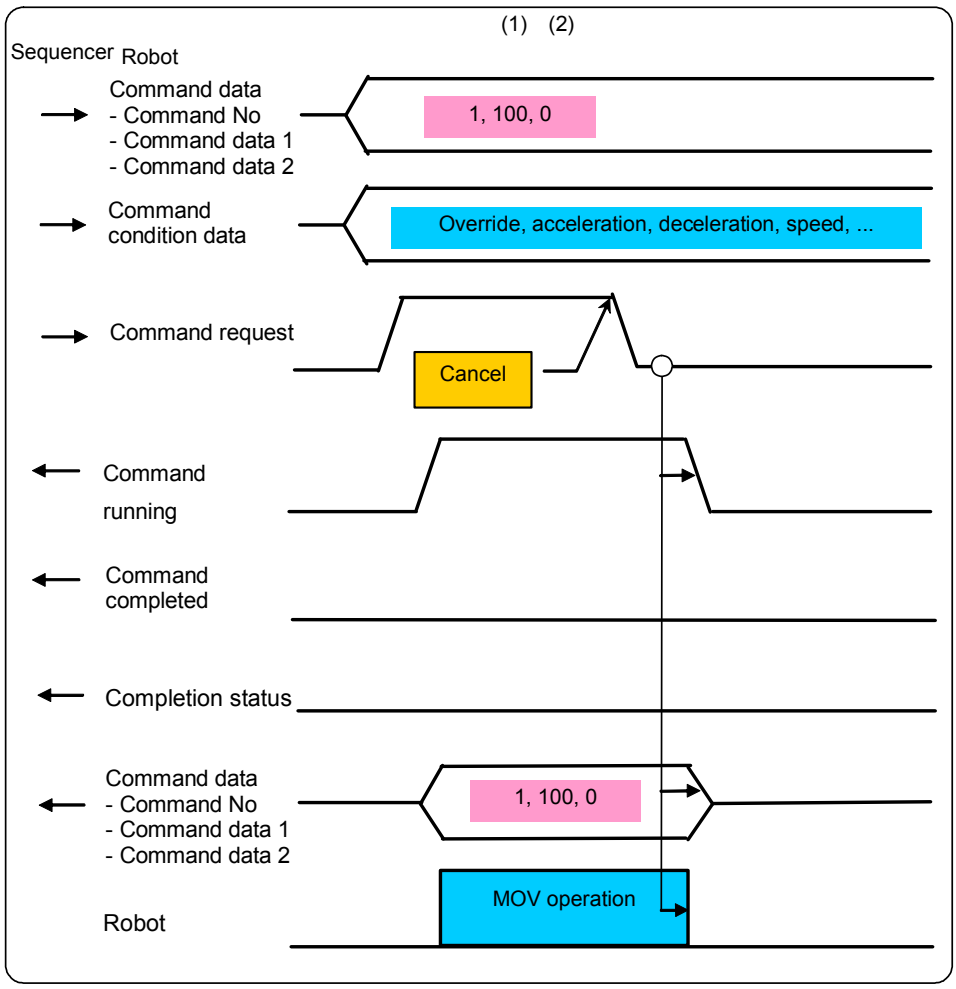
(1) When the robot's external operation authority gets invalid while performing the sequencer direct, the robot stops, sends "Command running OFF", sets "Completion status" to 30 (suspended), and sends "Command completed ON".

(2) When the sequencer receives "Command completed ON", the sequencer imports "Completion status" and sends "Command request OFF".

(3) When the robot receives "Command request OFF", the robot clears "Completion status" and "Command data" to zero and sends "Command completed OFF".

(6) Cancel Based on Command Request OFF Signal

When sequencer's "Command request" is turned OFF while performing the sequencer direct, the robot' operation can be terminated halfway (the robot slows down and stops in the same manner as stop input).



(1) When the sequencer wants to terminate the robot operation halfway while performing the sequencer direct, the sequencer sends "Command request OFF".

(2) When the robot receives "Command request OFF", the robot stops, clears "Command data" to zero and sends "Command running OFF".

8.3.3 Sample Ladder for Performing Operation Command

Here, describes a ladder program example which runs an operation command of sequencer direct performance function.

[Target function]

Runs an operation command of sequencer direct performance function (moves to position 1 with joint interpolation (command number: 1))

[Target robot]

The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Description]

Turn ON the sequencer direct performance trigger (M150) to run an operation command.
Operation result (completion status) is stored in D20.
When the operation is completed, M151 is turned ON. In this case, successful completion turns M152 ON, halfway suspension turns M153 ON, and abnormal completion turns M154 ON.
When M151 is turned ON, turn OFF the sequencer direct performance trigger (M150).

[Device details]

M150: Sequencer direct performance trigger
M151: Sequencer direct performed
M152: Sequencer direct performed successfully
M153: Sequencer direct performed suspendedly
M154: Sequencer direct performed abnormally

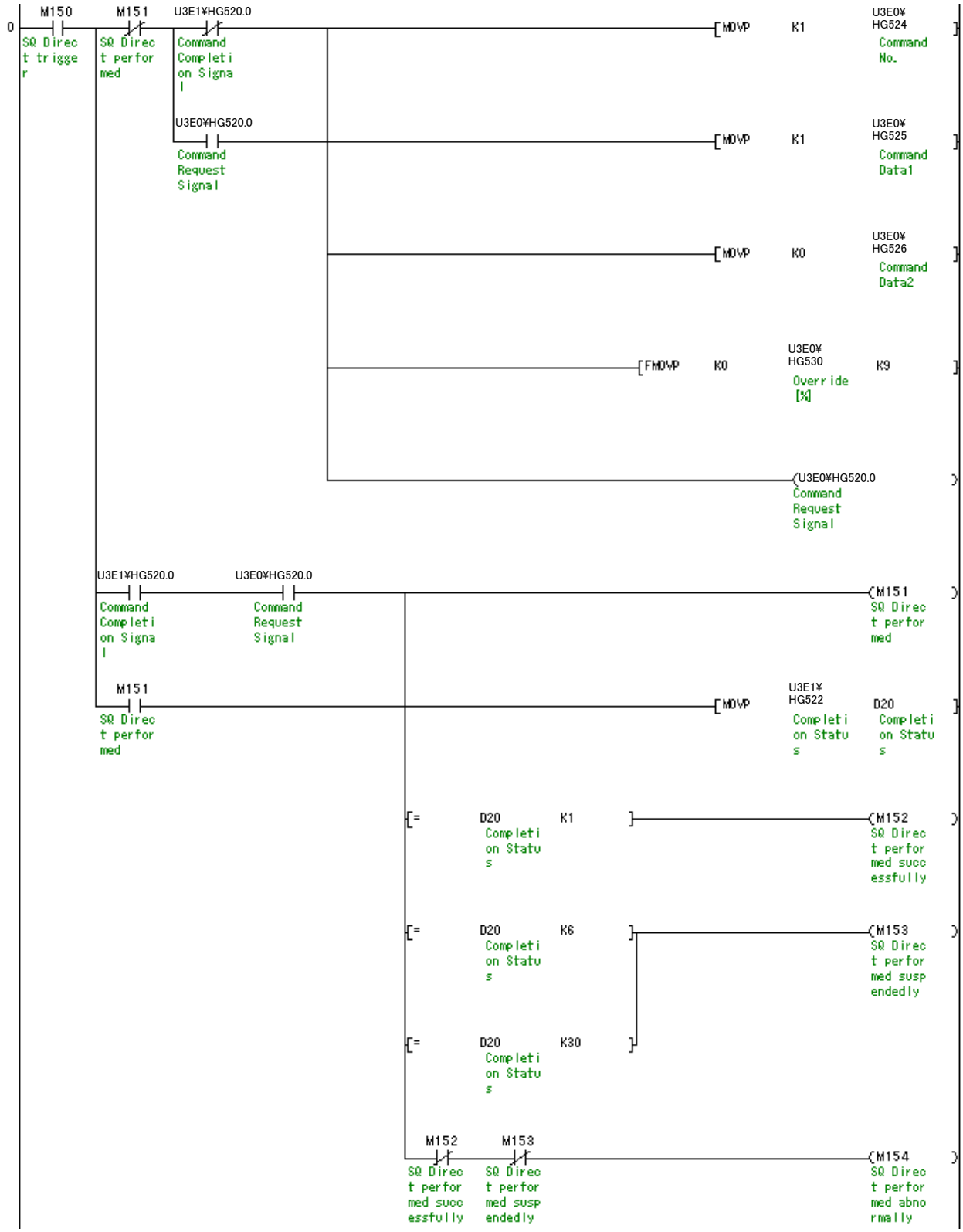
D20: Received data from the robot (completion status)

Precautions

To run an operation command, put the sequencer and robot in the state below in advance.
(For more information, refer to Page 80, "8.2.3 Prepare to Perform Sequencer Direct")

- The sequencer has gotten the robot's external operation authority.
- The robot is not in H or L level error state.
- The robot can select a program.
- The robot's servo is turned ON.

[Ladder]


0
M150
SQ Direct trigger
M151
SQ Direct performed
U3E1¥HG520.0
Command Completion Signal
U3E0¥HG520.0
Command Request Signal
MOVP K1 U3E0¥HG524 Command No.
MOVP K1 U3E0¥HG525 Command Data1
MOVP K0 U3E0¥HG526 Command Data2
FMOVP K0 U3E0¥HG530 Override [%] K9
U3E0¥HG520.0 Command Request Signal
U3E1¥HG520.0 Command Completion Signal
U3E0¥HG520.0 Command Request Signal
M151 SQ Direct performed
M151
SQ Direct performed
MOVP U3E1¥HG522 Completion Status D20 Completion Status
= D20 Completion Status K1 M152 SQ Direct performed successfully
= D20 Completion Status K6 M153 SQ Direct performed suspendedly
= D20 Completion Status K30
M152
SQ Direct performed successfully
M153
SQ Direct performed suspendedly
M154 SQ Direct performed abnormally

8.3.4 Control Robot Hand

Dedicated I/O signals allows you to control a robot hand.
Controls the hand by allocating an I/O signal number for hand control according to the parameters listed in the table below.
The condition to control the robot hand through external signal is "T/B invalid"

(1) Dedicated I/O parameters for hand control

Parameter Name	Category	Name	Function	Signal Level	Factory Default Signal No
HANDENA	Input	Hand control permission input	Permits (ON)/ prohibits (OFF) the robot hand control through external signal. Note: The robot can control a hand during automatic operation. For security purposes, make sure to interlock the robot and external equipment such as a sequencer.	Level	-1,
	Output	Hand control permission output	Outputs the permission (ON)/ prohibition (OFF) of the robot hand control through external signal. When the hand control permission input signal is turned ON while T/B is invalid, it gets permitted (ON).		-1
HANDOUT	Input	Hand output control signal	Sets up external input signal range for robot hand control. The specified external input signals are mapped in order to the hand signals established by the parameter HANDTYPE. Note1) Element 1: Start number of hand output control signal Element 2: End number of hand output control signal	Edge	-1,-1

Note1) Hand type
Factory default setting assumes that a hand of double solenoidal type is used. To use a single solenoidal type or to control the hand through general-purpose signals, change the parameter (HANDTYPE) as follows:

Table 8-1:Factory default parameter setting

Parameter Name	Initial value
HANDTYPE	D900, D902, D904, D906, , , ,

The values from left to right corresponds to the hand numbers 1, 2, ... The initial values are as follows:
Hand 1: Accesses the signal numbers 900, 901
Hand 2: Accesses the signal numbers 902, 903
Hand 3: Accesses the signal numbers 904, 905
Hand 4: Accesses the signal numbers 906, 907

<How to set up>
To use double solenoidal type, specify the number by attaching 'D' at the beginning of signal number.
For double solenoidal type, the hands 1 - 4 are available.
To use single solenoidal type, specify the number by attaching 'S' at the beginning of signal number.
For single solenoidal type, the hands 1 - 8 are available.
Example:
(1) To allocate two general-purpose signal numbers beginning with #10 to the hands of double solenoidal type:
HANDTYPE=D10, D12, , , , ,
(2) To allocate three general-purpose signal numbers beginning with #10 to the hands of single solenoidal type:
HANDTYPE=S10, S11, S12, , , , ,
(3) To allocate general-purpose signal #10 to the hand 1 of double solenoidal type, #12 to the hand 2 of single solenoidal type:
HANDTYPE=D10, S12, , , , ,

(2) Mapping hand signal with parameter HANDTYPE

When the parameter HANDTYPE setting is changed, robot hand signal corresponding to the hand output control signal may change. The signals allocated to hand signals correspond to the hand output control signals in order.

● The tables below list the correspondence to the robot hand output signals, when hand output control signals (HANDOUT) are set to "10080, 10087":

a) Parameter HANDTYPE=D900,D902,D904,D906, , , , (factory defaults):

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	900	901	902	903	904	905	906	907
Hand output cont sig	10080	10081	10082	10083	10084	10085	10086	10087

b) Parameter HANDTYPE=D10,D12, , , , , ,:

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	10	11	12	13	-	-	-	-
Hand output cont sig	10080	10081	10082	10083	-	-	-	-

The areas 10084 - 10087 are not used.

c) Parameter HANDTYPE=S10, , ,S13, , , , :

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	10		-	-	-	-	13	
Hand output cont sig	10080		-	-	-	-	10081	

The areas 10082 - 10087 are not used.

d) Parameter HANDTYPE=D10,S12, , , , , ,:

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	10	11	12		-	-	-	-
Hand output cont sig	10080	10081	10082		-	-	-	-

The areas 10083 - 10087 are not used.

● It also supports hands 5 - 8 of parameter HANDTYPE.
When parameter HANDTYPE=D900, D902, D904, D906, D10, D12, D14, D16,
hand output control signal (HANDOUT) are set to "10080, 10095":

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	900	901	902	903	904	905	906	907
Hand output cont sig	10080	10081	10082	10083	10084	10085	10086	10087

Hand No	5		6		7		8	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	10	11	12	13	14	15	16	17
Hand output cont sig	10088	10089	10090	10091	10092	10093	10094	10095

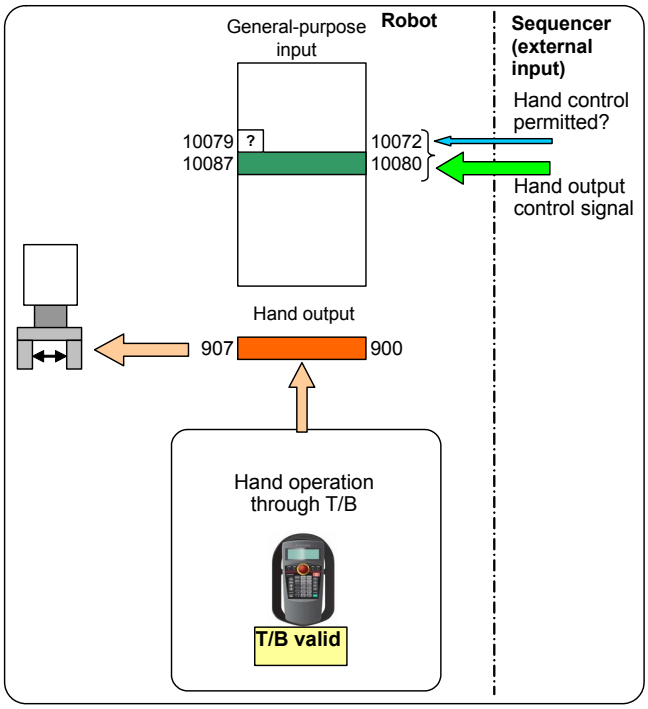
(3) Hand control image

The hand control image is shown below based on the robot parameter setting below (HANDTYPE is factory default):

- HANDENA (hand control permitted) = 10079,10079
- HANDOUT (hand output control signal) = 10080,10087

a) T/B is valid:

T/B controls a robot hand. Hand control through external signals is prohibited.



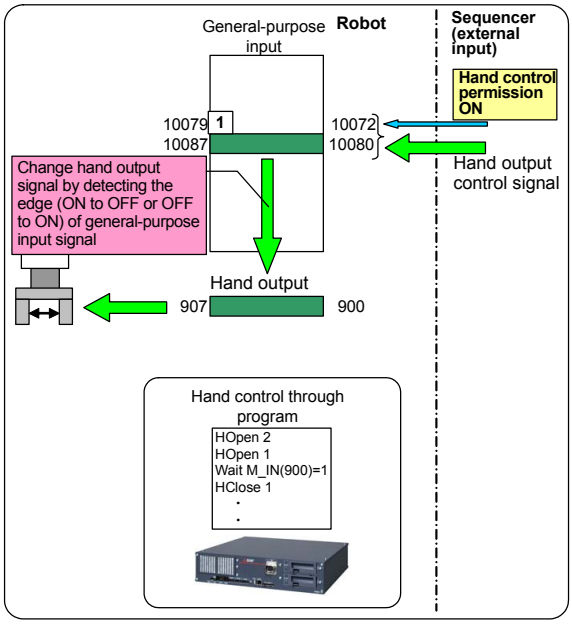
b) T/B is invalid:

Allows you to select either external signals or controller (robot program or forced output) for the robot hand control according to the hand control permission (HANDENA) signal.

b-1) HANDENA signal is ON:

External signal controls the robot hand.

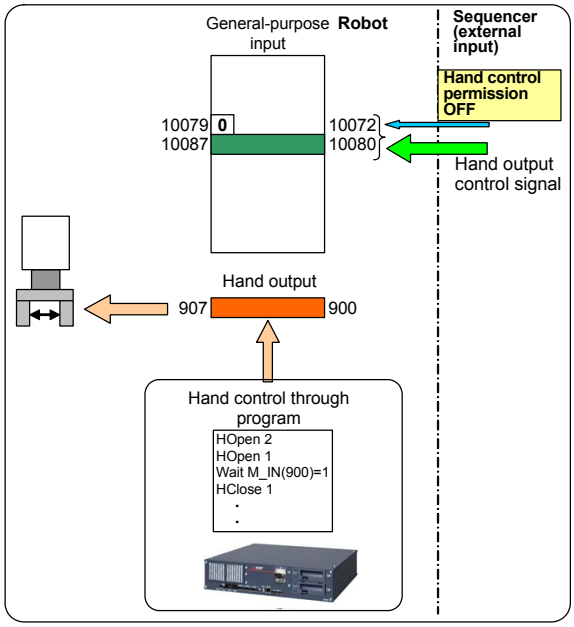
→ Changes the hand output signal by detecting the edge (ON to OFF or OFF to ON) of general-purpose input signal which is allocated to the hand output control signal.



b-2) HANDENA signal is OFF:

Controller (robot program or forced output) controls the robot hand.

→ The hand signal control through program command changes hand output signals.



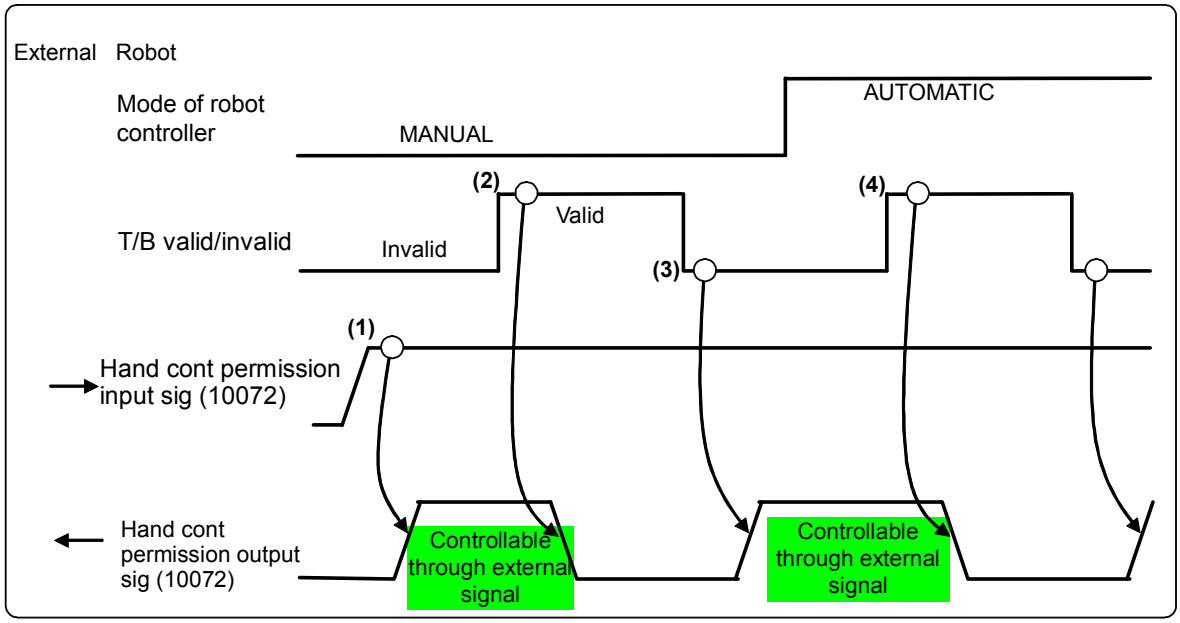
8.3.5 Timing Chart for Robot Hand Control

The figure below shows the hand control timing chart when the robot parameter is set up as follows:

- HANDENA (hand control permitted) = 10079,10079
- HANDOUT (hand output control signal) = 10080,10087

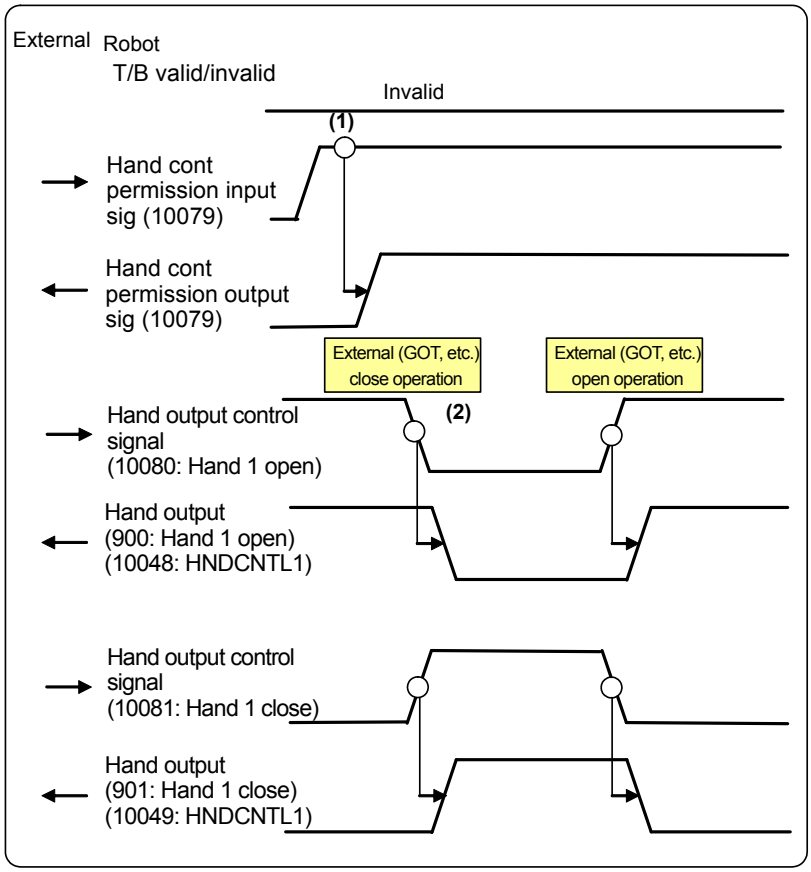
- HNDCNTL1 (hand output signal) = 10048, 10055 (factory defaults)
- HANDTYPE (hand type) = D900, D902, D904, D906, , , , (factory defaults)

a) Changes of hand control enabled state according to T/B valid/ invalid



(1) When an external controller (sequencer, etc.) sends "Hand control permission input ON" while the robot is in MANUAL mode and T/B is invalid, the robot sends "Hand control permission output ON" to enable the hand control through external signal (hand output control signal).
(2) When T/B gets valid, the robot sends "Hand control permission output OFF" to prohibit the hand control through external signal.
(3) When T/B gets invalid again, the robot sends "Hand control permission output ON" to enable the hand control through external signal.
(4) When T/B gets valid even while the robot is in MANUAL mode, the robot sends "Hand control permission output OFF" to prohibit the hand control through external signal.
(Error "H5000 Teaching" occurs.)

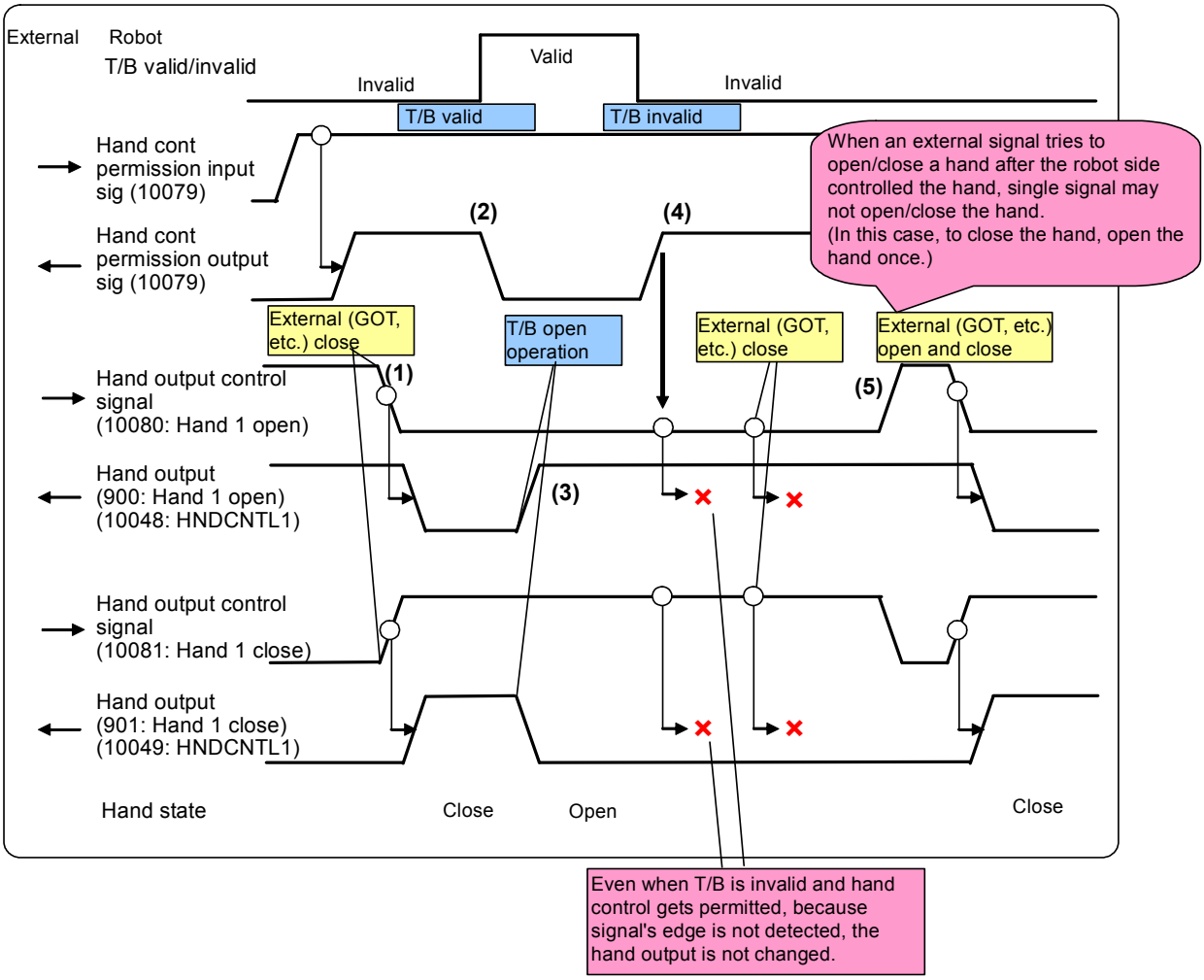
b) Robot hand control through external signal



(1) An external controller (sequencer, etc.) sends "Hand control permission input ON".
When T/B is invalid, the robot sends "Hand control permission output ON" to enable the hand control through external signal (hand output control signal).

(2) When "Hand control permission output" is ON, the robot hand output signal changes according to the edge (ON to OFF or OFF to ON) of signals which are allocated to the hand output control signal.

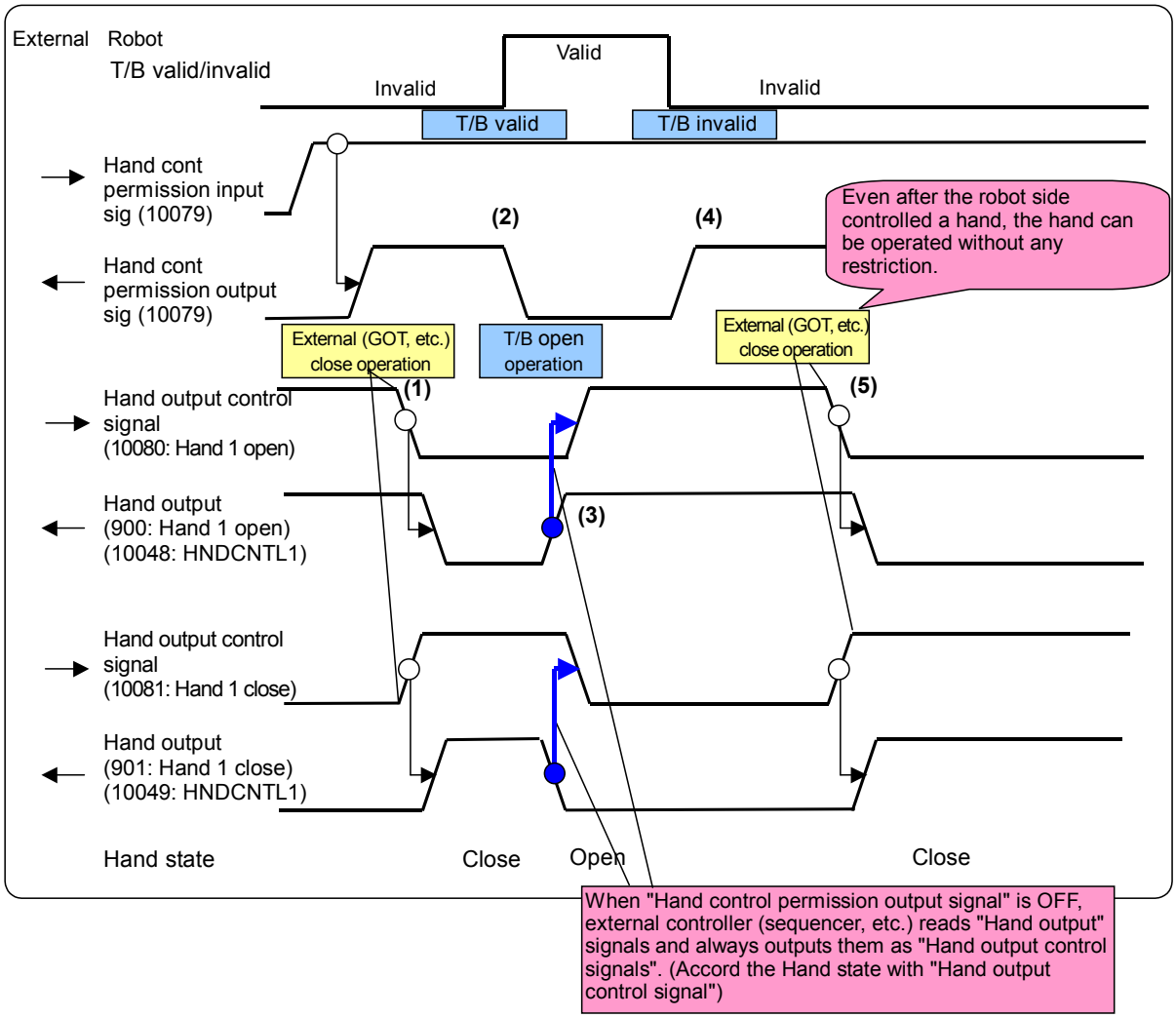
c) Robot hand control 1 when T/B gets invalid



(1) When "Hand control permission output" is ON, the robot hand output signal changes according to the edge (ON to OFF or OFF to ON) of signals which are allocated to the hand output control signal.
(2) When T/B gets valid, the robot sends "Hand control permission output OFF" to prohibit the hand control through external signal.
(3) When T/B opens/closes a hand while T/B is valid, the hand output signal changes and the hand operates.
(4) When T/B gets invalid, the robot sends "Hand control permission output ON" again to enable the hand control through external signal.
In this case, because the edge of "Hand output control signal" is not detected even when "Hand output control signal" is different from "Hand output", "Hand output" does not change.
(5) To operate a hand, change "Hand output control signal". (Refer to the caution below.)

⚠ CAUTION When T/B gets enabled halfway and T/B operates a hand, "Hand output control signal" from the sequencer may be different from actual "Hand output". In this case, when "Hand output control signal" from the sequencer is not accorded with actual "Hand output", as described in the timing chart above, single hand operation may not complete the hand operation.

d) Robot hand control 2 when T/B gets invalid (recommended)

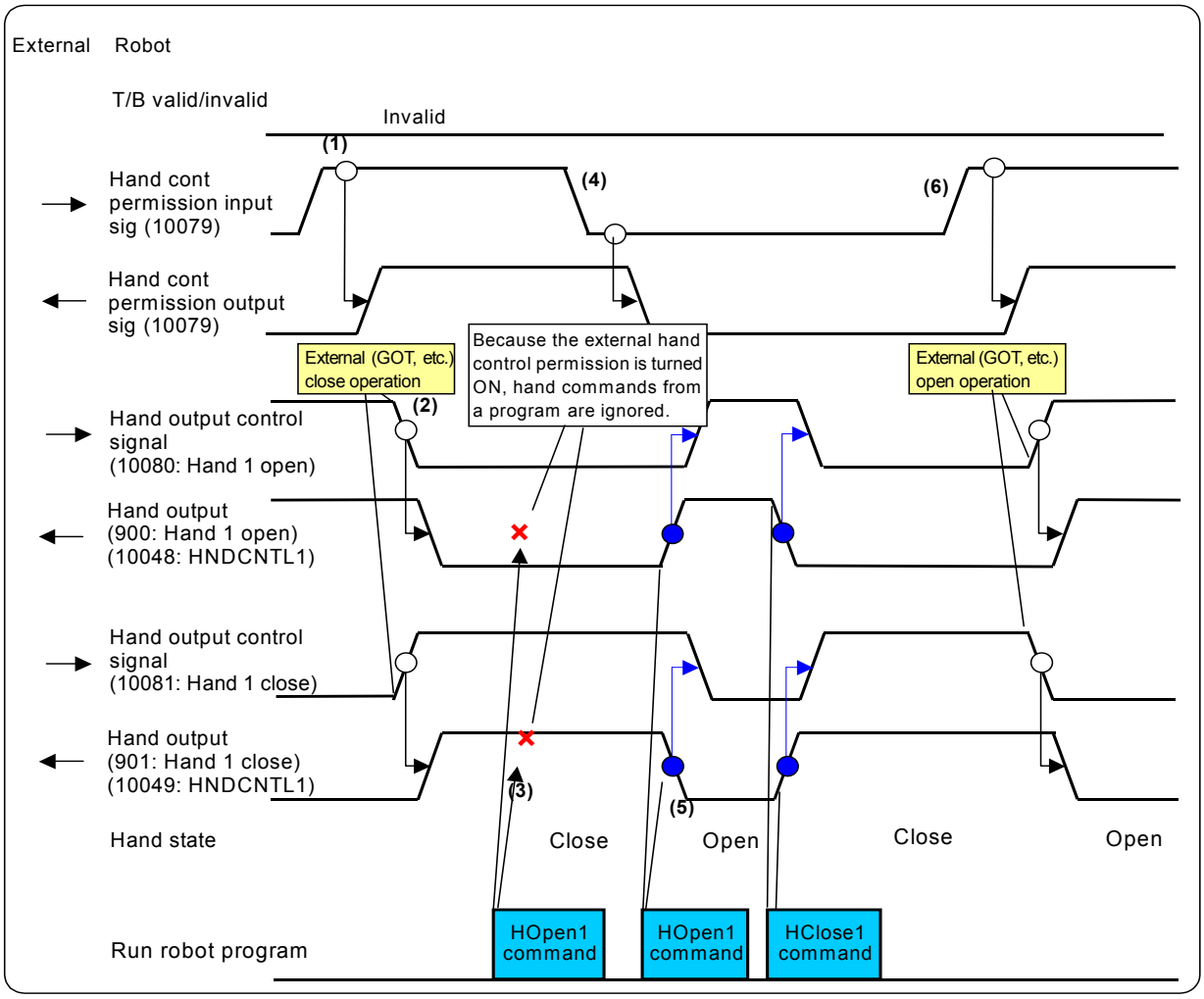


(1) When "Hand control permission output" is ON, the robot hand output signal changes according to the edge (ON to OFF or OFF to ON) of signals which are allocated to the hand output control signal.
(2) When T/B gets valid, the robot sends "Hand control permission output OFF" to prohibit the hand control through external signal.
(3) When T/B opens/closes a hand while T/B is valid, the hand output signal changes and the hand operates.

⚠ CAUTION When "Hand control permission output" is OFF, external controller (sequencer, etc.) reads "Hand output" signals and always output them as "Hand output control signals," thereby according the hand state with "Hand output control signal".

(4) When T/B gets invalid, the robot sends "Hand control permission output ON" again to enable the hand control through external signal.
(5) Even after the robot side controlled a hand, the hand can be controlled with "Hand output control signal" without any restriction.

e) Switch between hand control with external signal and hand control with robot program



(1) To control the robot hand with external signals, send "Hand control permission input ON".
When T/B is invalid, the robot sends "Hand control permission output ON" to enable the hand control through external signal (hand output control signal).

(2) When "Hand control permission output" is ON, a robot hand can be operated with "Hand output control signal".

(3) When "Hand control permission output" is ON, the hand will not operate even if a robot program issues a hand control command.

(4) To control a hand from the robot, send "Hand control permission input OFF".
When the robot confirmed that "Hand control permission output" is OFF, the robot sends "Hand control permission output OFF".

(5) When "Hand control permission output" is OFF, the robot program's hand control command can operate the hand. (T/B operation and forced output from RT ToolBox3 also can operate the hand).
When "Hand control permission output" is OFF, external controller (sequencer, etc.) reads "Hand output" signals and always output them as "Hand output control signals" to accord the hand state with "Hand output control signal". Then, when "Hand control permission output" is turned ON again, the hand can be controlled without any restriction.

(6) To control the robot hand with external signals again, send "Hand control permission input ON."
When T/B is invalid, the robot sends "Hand control permission output ON" to enable the hand control through external signal (hand output control signal).

8.3.6 Sample Ladder for Robot Hand Control

Here, describes a ladder program example which controls a robot hand with robot dedicated signals in the sequencer.

[Target function]

Controls a robot hand (opens/closes hand 1)

[Target robot]

The target robot is robot 2 of multiple CPUs (robot's multiple CPU input offset parameter is initial value)

[Robot parameter setting]

- HANDENA (hand control permitted) = 10079,10079
- HANDOUT (hand output control signal) = 10080,10087

- HNDCNTL1 (hand output signal status) = 10048, 10055 (factory defaults)
- HANDTYPE (hand type) = D900, D902, D904, D906, , , , (factory defaults)

[Description]

When the robot is in AUTOMATIC mode, the sequencer controls a robot hand.
When M160 is turned ON (M161 is turned OFF), hand 1 opens. When M161 is turned ON (M160 is turned OFF), hand 1 closes.
When the robot is in MANUAL mode ("Hand control permission output" is OFF), the sequencer reads "Hand output signal status" and always outputs it as "Hand output control signal", thereby according the hand state with "Hand output control signal" output from the sequencer.
Robot's dedicated output and input signals are batch transferred to B0/ B100 and after respectively.

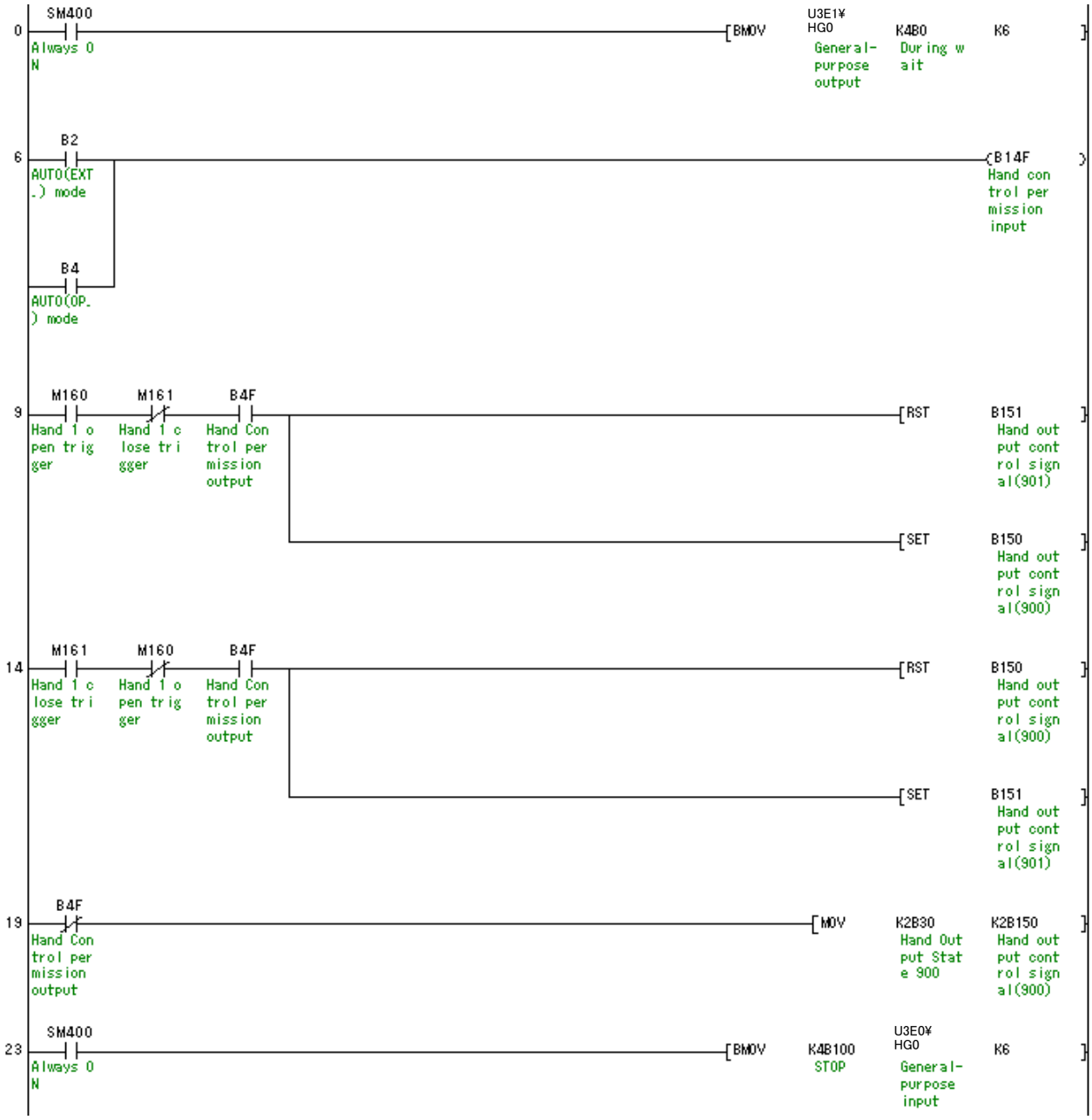
[Device details]

M160: Hand 1 open trigger
M161: Hand 1 close trigger

B14F: Hand control permission input signal
B150: Hand output control signal (open hand 1)
B151: Hand output control signal (close hand 1)

M4F: Hand control permission output signal
B30: Hand output signal status (open hand 1)
B31: Hand output signal status (close hand 1)

[Ladder]


0
SM400
Always ON
BMOV
U3E1¥HG0
General-purpose output
K4B0
During wait
K6
6
B2
AUTO(EXT.) mode
B4
AUTO(OP.) mode
B14F
Hand control permission input
9
M160
Hand 1 open trigger
M161
Hand 1 close trigger
B4F
Hand Control permission output
RST
B151
Hand output control signal(901)
SET
B150
Hand output control signal(900)
14
M161
Hand 1 close trigger
M160
Hand 1 open trigger
B4F
Hand Control permission output
RST
B150
Hand output control signal(900)
SET
B151
Hand output control signal(901)
19
B4F
Hand Control permission output
MOV
K2B30
Hand Output State 900
K2B150
Hand output control signal(900)
23
SM400
Always ON
BMOV
K4B100
STOP
U3E0¥HG0
General-purpose input
K6

8.4 Samples

Here, as samples for sequencer direct performance, describes the examples that the robot takes out works. The examples are a robot program which takes out works, a ladder which uses sequencer direct performance command plus hand control function, and an operation setting in the GOT screen.

8.4.1 Robot Program

```
'// Initial setting
HOPEN 1                 ' Opens hand                          (1)
'
'// Takes out a work
Mov P_DM(1),-200        ' Moves over the takeout position     (2)
Ovrd 70                 ' Override 70%                        (3)
Accel 50,50             ' Accelerate and decelerate 50%       (4)
Mvs P_DM(1)             ' Moves to the takeout position       (5)
Dly 0.2                 ' Waits for 0.2sec (arrival check)    (6)
HClose 1                ' Closes hand                         (7)
Wait M_IN(901)=1        ' Checks hand close                   (8)
Dly 0.1                 ' Waits for 0.1sec                    (9)
Ovrd 100                ' Override 100%s                      (10)
Accel 100,100           ' Accelerate and decelerate 100%      (11)
Mvs P_DM(1),-200        ' Moves over the takeout position     (12)
'
End
```

This program is assumed to be a vertical type robot. When a robot is horizontal type, because tool's Z axis is opposite to vertical type, change the code as follows:

Mov P_Dm(1), -200
to Mov P_Dm(1), +200

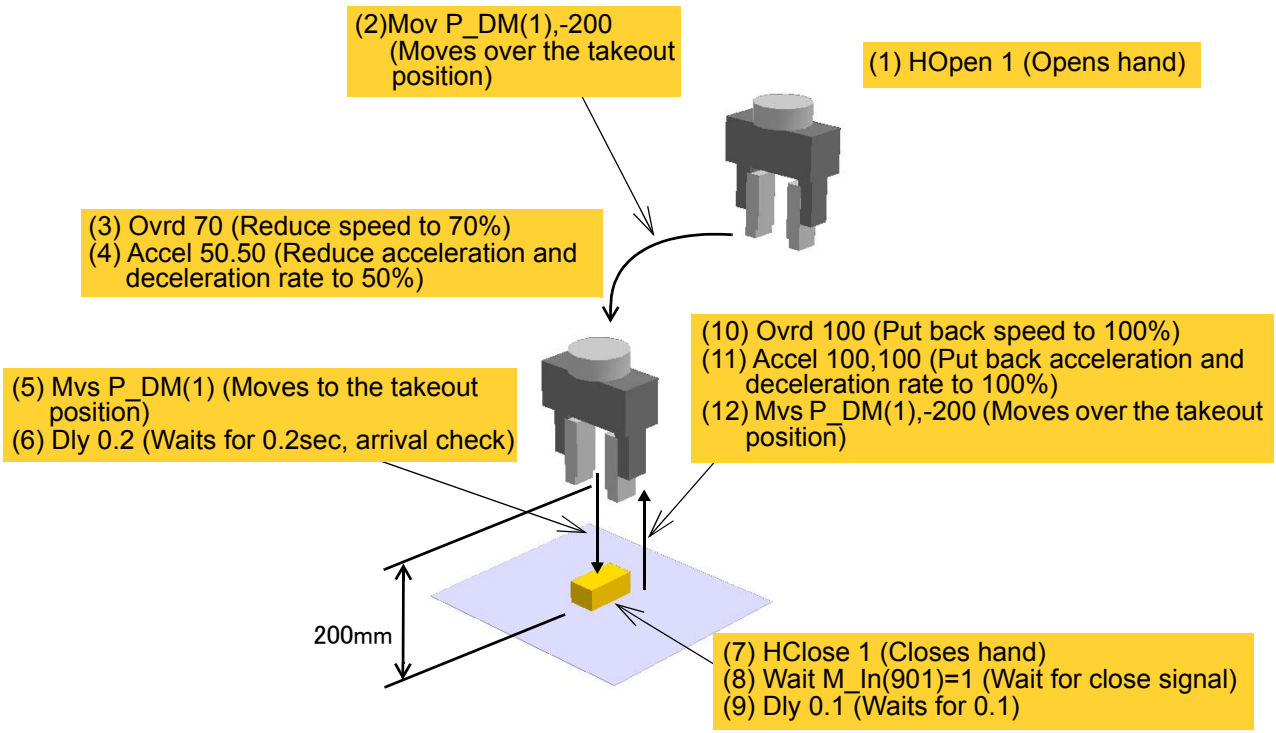


Fig.8-1:Robot operation

8.4.2 Sample Ladder Program

(1) Condition

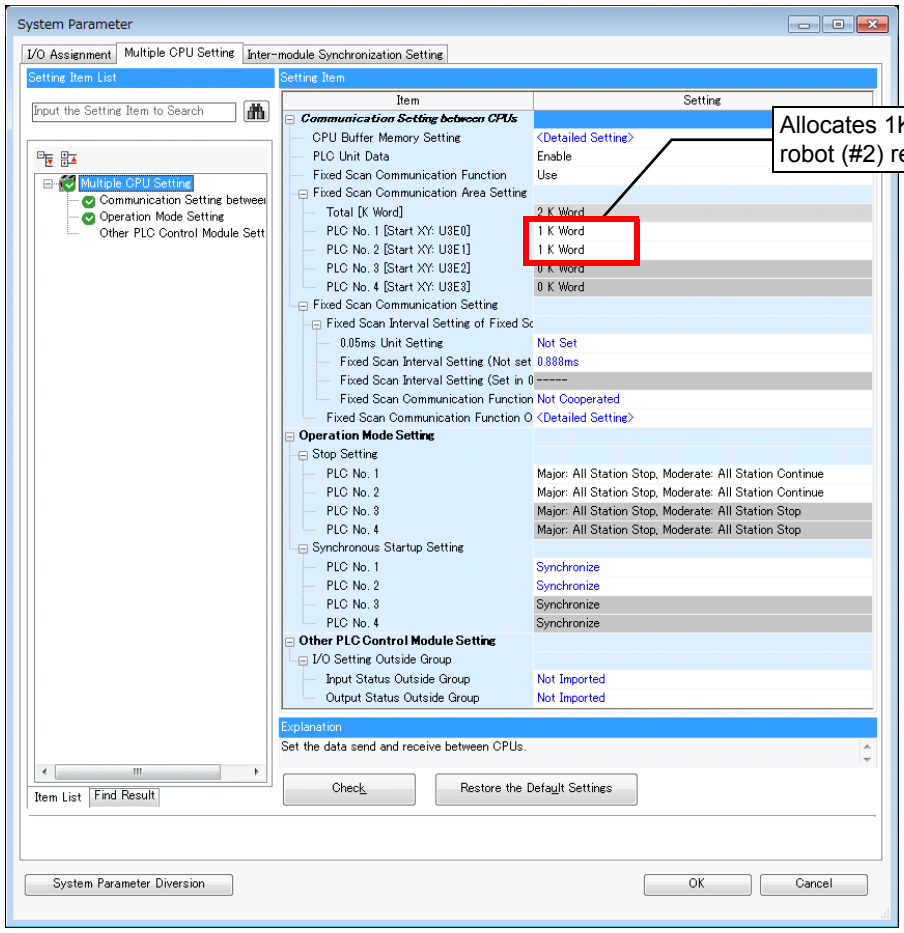
[CPU configuration]

Sequencer plus one robot

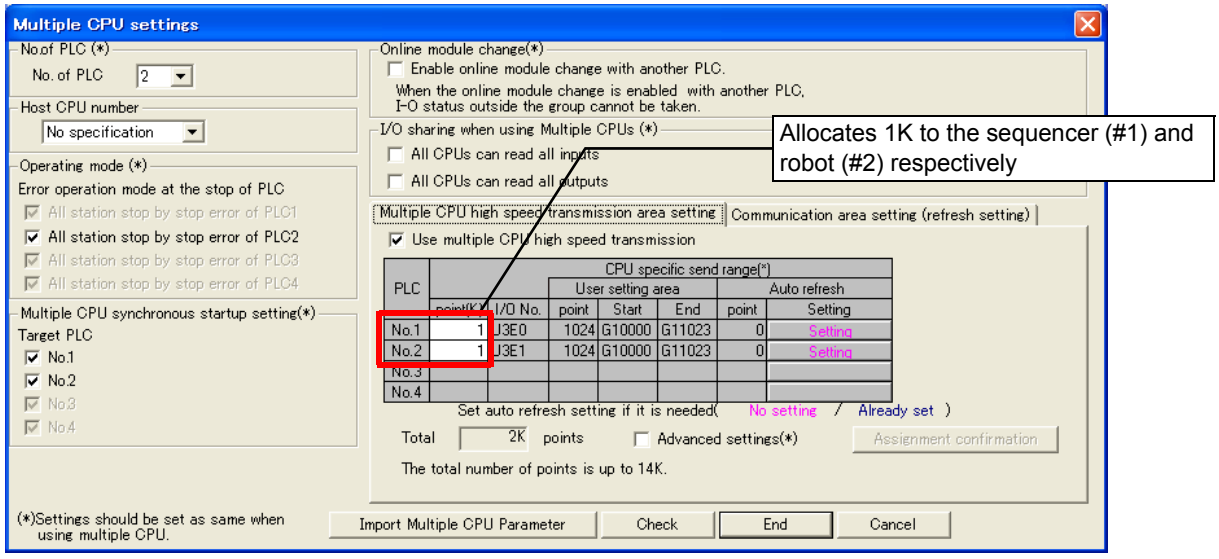
[Sequencer multiple CPUs setting]

The figure below shows the multiple CPU parameter setting of the sequencer:

<CR800-R series>



<CR800-Q series>



[Robot parameter setting]

● Selecting CPU buffer memory extended function

• IQMEM: Set both bits 0, 1 to one to enable the sequencer direct performance function

● Robot output signal control

• HANDENA (hand control permitted): 10079, 10079

• HANDOUT (hand output control signal): 10080, 10087

Use the robot input signals (10080 - 10087) to control the robot hand output signal (900 - 907)
Refer to Table 8-2.

[Robot hand]

Table 8-2:Hand output

Hand No	1		2		3		4	
Open/Close	Open	Close	Open	Close	Open	Close	Open	Close
Robot hand output sig	900	901	902	903	904	905	906	907
Hand output control signal	10080	10081	10082	10083	10084	10085	10086	10087

[Allocating robot dedicated I/O signals]

Allocate the signals HANDENA, HANDOUT as well as the dedicated signals allocated in initial setting. To handle the robot dedicated I/O signals in the sequencer, replace the robot dedicated I/O signals with device B.

Parameter Name	Robot Input Signal Name	Robot Output Signal Name	Robot Mapping		Sequencer Mapping (B)	
			Input	Output	Output	Input
STOP	Stop input	Pausing output	10000	10000	100	000
RCREADY	-	Controller power ON ready	-	10001	-	001
ATEXTMD	-	Remote mode output	-	10002	-	002
TEACHMD	-	Teaching mode output	-	10003	-	003
ATTOPMD	-	Automatic mode output	-	10004	-	004
IOENA	Operation rights input	Operation rights output	10005	10005	105	005
START	Start input	Operating output	10006	10006	106	006
STOPSTS	-	Stop signal input	-	10007	-	007
SLOTINIT	Program reset input	Program selection enabled output	10008	10008	108	008
ERRRESET	Error reset input	Error occurring output	10009	10009	109	009
SRVON	Servo ON input	In servo ON output	10010	10010	10A	00A
SRVOFF	Servo OFF input	Servo ON disable output	10011	10011	10B	00B
CYCLE	Cycle stop input	In cycle stop operation output	10012	10012	10C	00C
SAFEPOS	Safe point return input	In safe point return output	10013	10013	10D	00D
BATERR	-	Battery voltage drop	-	10014	-	00E
OUTRESET	General-purpose output signal reset	-	10015	-	10F	-
HLVLERR	-	High level error output	-	10016	-	010
LLVLERR	-	Low level error output	-	10017	-	011
CLVLERR	-	Warning level error output	-	10018	-	012
EMGERR	-	Emergency stop output	-	10019	-	013
PRGSEL	Program selection input	-	10020	-	114	-
OVRDSEL	Override selection input	-	10021	-	115	-
PRGOUT	Program No. output request	Program No. output	10022	10022	116	016
LINEOUT	Line No. output request	Line No. output	10023	10023	117	017
OVRDOUT	Override value request	Override value output	10024	10024	118	018
ERROUT	Error No. output request	Error No. output	10025	10025	119	019
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
IODATA	Numeric value input 0	Numeric value output 0	10032	10032	120	020
↑	Numeric value input 1	Numeric value output 1	10033	10033	121	021
↑	Numeric value input 2	Numeric value output 2	10034	10034	122	022
↑	Numeric value input 3	Numeric value output 3	10035	10035	123	023
↑	Numeric value input 4	Numeric value output 4	10036	10036	124	024
↑	Numeric value input 5	Numeric value output 5	10037	10037	125	025
↑	Numeric value input 6	Numeric value output 6	10038	10038	126	026
↑	Numeric value input 7	Numeric value output 7	10039	10039	127	027
↑	Numeric value input 8	Numeric value output 8	10040	10040	128	028
↑	Numeric value input 9	Numeric value output 9	10041	10041	129	029
↑	Numeric value input 10	Numeric value output 10	10042	10042	12A	02A

Parameter Name	Robot Input Signal Name	Robot Output Signal Name	Robot Mapping		Sequencer Mapping (B)		
			Input	Output	Output	Input	
↑	Numeric value input 11	Numeric value output 11	10043	10043	12B	02B	
↑	Numeric value input 12	Numeric value output 12	10044	10044	12C	02C	
↑	Numeric value input 13	Numeric value output 13	10045	10045	12D	02D	
↑	Numeric value input 14	Numeric value output 14	10046	10046	12E	02E	
↑	Numeric value input 15	Numeric value output 15	10047	10047	12F	02F	
HNDCNTL1	-	Hand output signal state 900	-	10048	-	030	
↑	-	Hand output signal state 901	-	10049	-	031	
↑	-	Hand output signal state 902	-	10050	-	032	
↑	-	Hand output signal state 903	-	10051	-	033	
↑	-	Hand output signal state 904	-	10052	-	034	
↑	-	Hand output signal state 905	-	10053	-	035	
↑	-	Hand output signal state 906	-	10054	-	036	
↑	-	Hand output signal state 907	-	10055	-	037	
HNDSTS1	-	Hand input signal state 900	-	10056	-	038	
↑	-	Hand input signal state 901	-	10057	-	039	
↑	-	Hand input signal state 902	-	10058	-	03A	
↑	-	Hand input signal state 903	-	10059	-	03B	
↑	-	Hand input signal state 904	-	10060	-	03C	
↑	-	Hand input signal state 905	-	10061	-	03D	
↑	-	Hand input signal state 906	-	10062	-	03E	
↑	-	Hand input signal state 907	-	10063	-	03F	
USRAREA	-	User defined area 1	-	10064	-	040	
↑	-	User defined area 2	-	10065	-	041	
↑	-	User defined area 3	-	10066	-	042	
↑	-	User defined area 4	-	10067	-	043	
↑	-	User defined area 5	-	10068	-	044	
↑	-	User defined area 6	-	10069	-	045	
↑	-	User defined area 7	-	10070	-	046	
↑	-	User defined area 8	-	10071	-	047	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
HANDENA	Hand control permission input	Hand control permission output	10079	10079	14F	04F	Allocation added with dedicated I/O parameters HANDENA, HANDOUT
HANDOUT	Hand output control 900	-	10080	-	150	-	
↑	Hand output control 901	-	10081	-	151	-	
↑	Hand output control 902	-	10082	-	152	-	
↑	Hand output control 903	-	10083	-	153	-	
↑	Hand output control 904	-	10084	-	154	-	
↑	Hand output control 905	-	10085	-	155	-	
↑	Hand output control 906	-	10086	-	156	-	
↑	Hand output control 907	-	10087	-	157	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	
-	-	-	-	-	-	-	

(2) Details

[Sequencer device mapping]

Device	Name	Description
M1	Robot transfer instruction	Turn it ON when carrying out an operation to take out works in the system.
M5	System running	Turn it ON when system is running. Turn it OFF when a suspension or error occurred.
M7	Error reset instruction	When instructing an error reset in the system, turn it ON.
M200 - M209	Work takeout operation	Sets up the flow of work takeout operation.
M240	Sequencer direct practicable condition	Turn it ON, when the sequencer direct is practicable.
M250	Sequencer direct performance trigger	When requesting for sequencer direct performance, turn it ON.
M251	Sequencer direct performed	When the sequencer direct performance is completed, turn it ON.
M252	Sequencer direct performed successfully	When the sequencer direct performance is successfully completed, turn it ON.
M253	Sequencer direct performance suspended	When the sequencer direct performance is suspended (paused), turn it ON. →It is necessary to issue this signal to suspend the system.
M254	Sequencer direct performance error exit	When the sequencer direct performance is unavailable, turn it ON. →It is necessary to issue this signal to admit the system error.
D20	Sequencer direct performance completion status	Stores the completion status of sequencer direct performance.
D104-D106	Command data value	Sets up the command data for sequencer direct performance.
D110-D118	Command condition data value	Sets up the command condition data for sequencer direct performance.
T10	Work takeout position arrival check	A timer to set up the delay time for arrival check after moving to the work takeout position.
T20	Hand close delay timer	A timer to set up the delay time after hand close.

[Description]

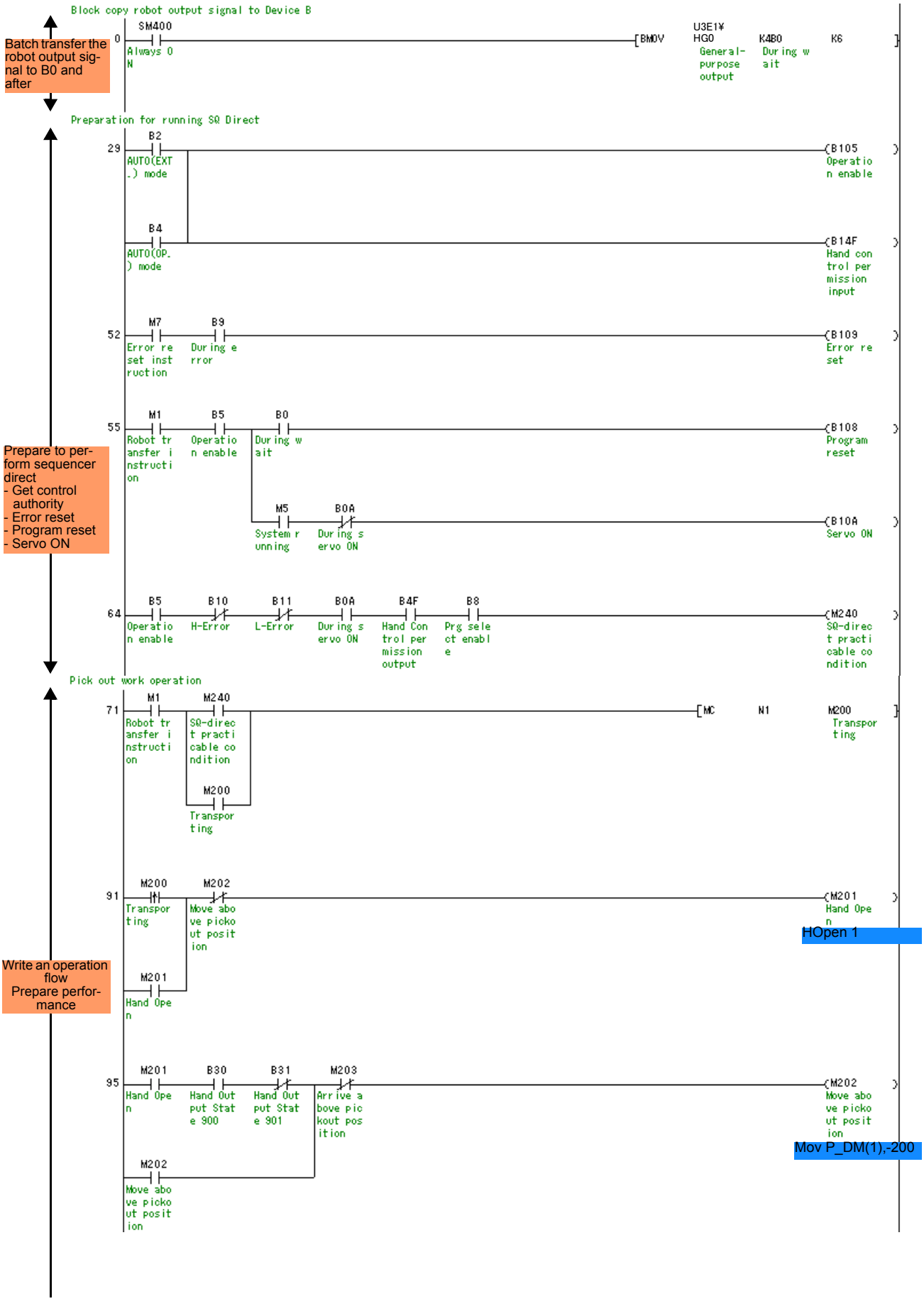
(1) When M1 (transfer instruction) is turned ON, carries out an operation to take out a work. When the operation is completed, Work transfer completed (M209) is turned ON.

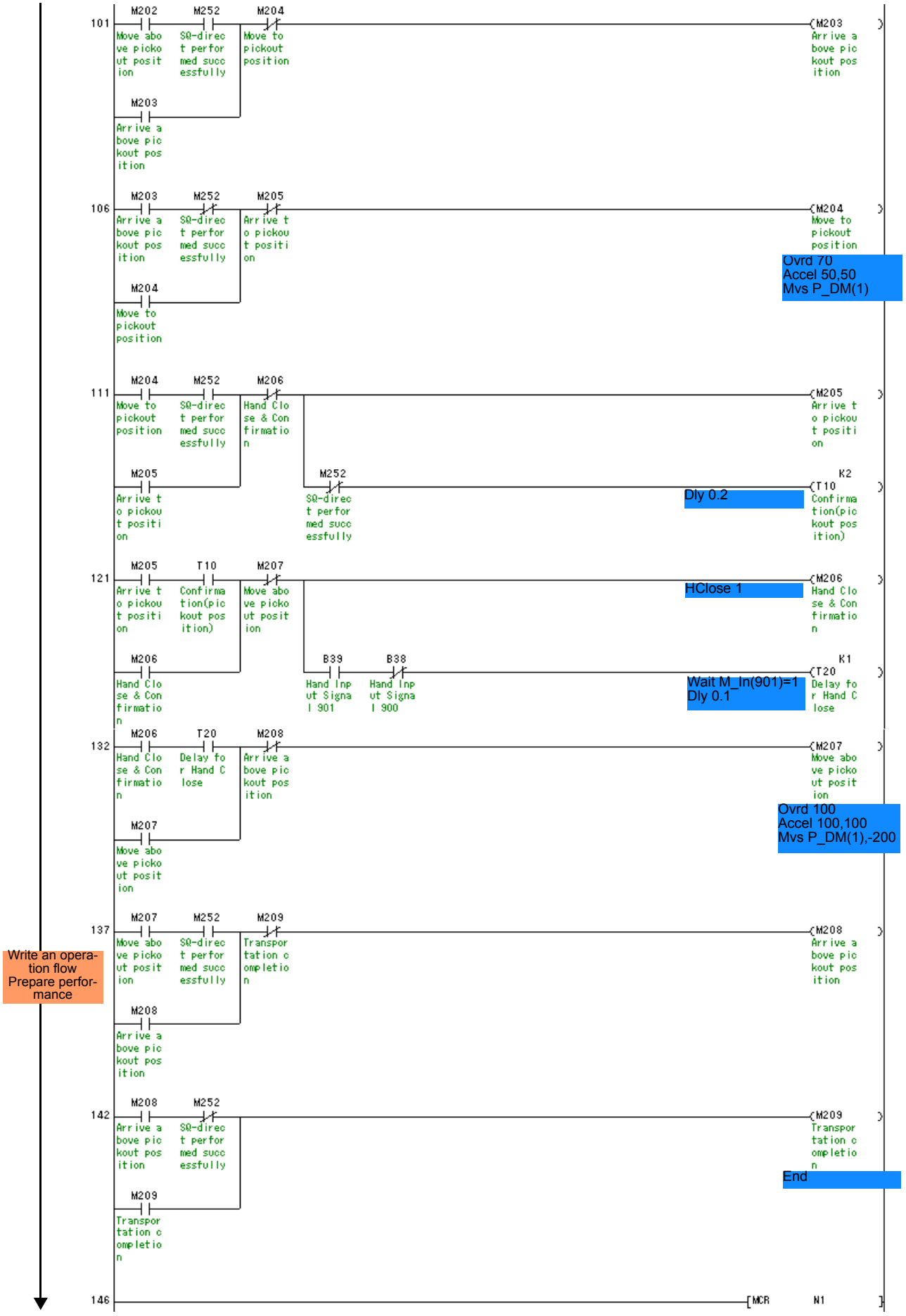
(2) The robot operates only when the System running (M5) is ON.
When the sequencer direct performance is suspended (M253 is turned ON) and becomes error (M254 is turned ON), carry out a system control (create it separately) to stop the system and turn OFF System running (M5). When System running (M5) is turned ON due to the resume, the robot operation resumes.

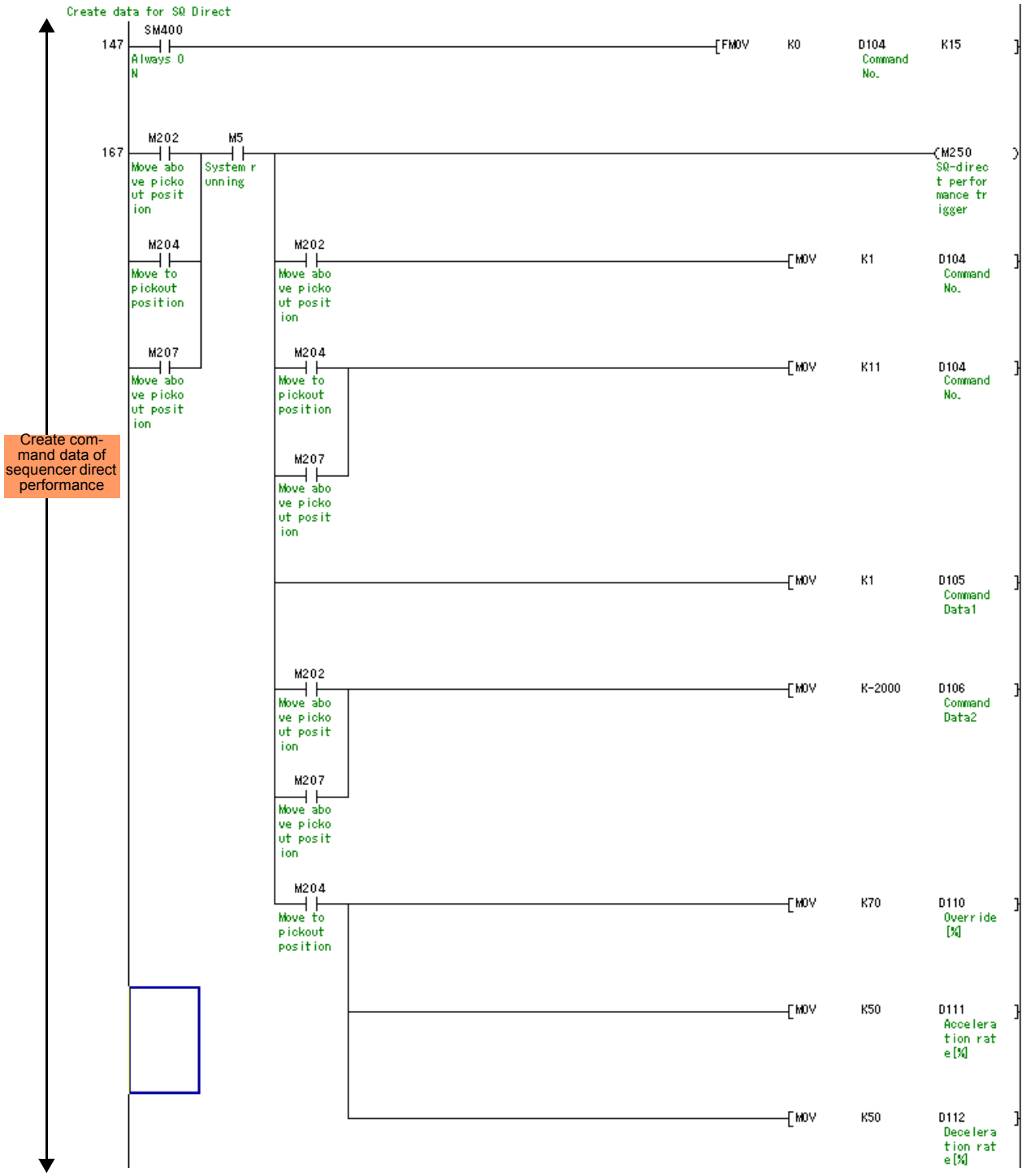
(3) The robot hand control in the sequencer is carried out only in AUTOMATIC mode.

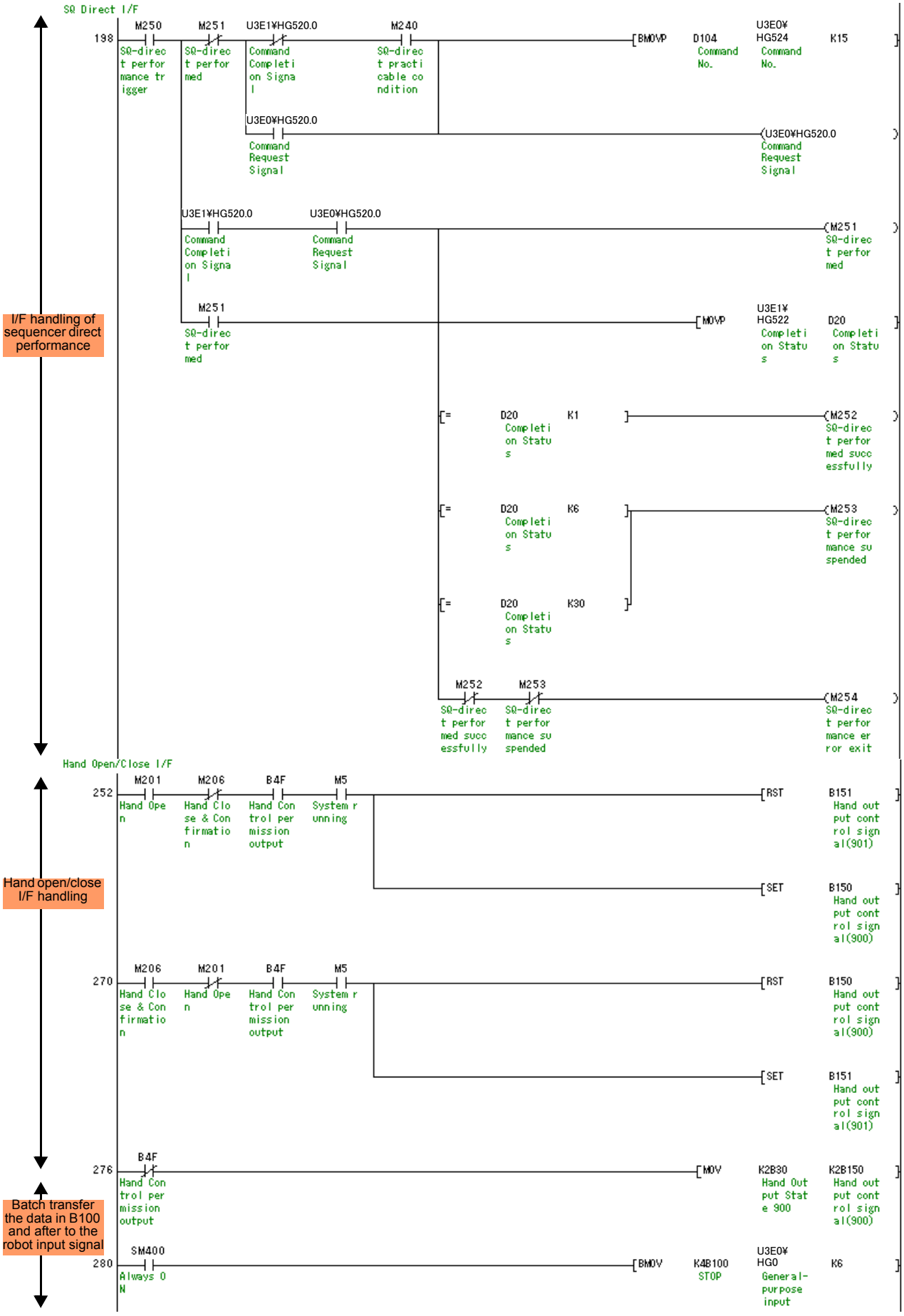
(4) When Error reset instruction (M7) is turned ON, the robot error is reset.

[Ladder program]



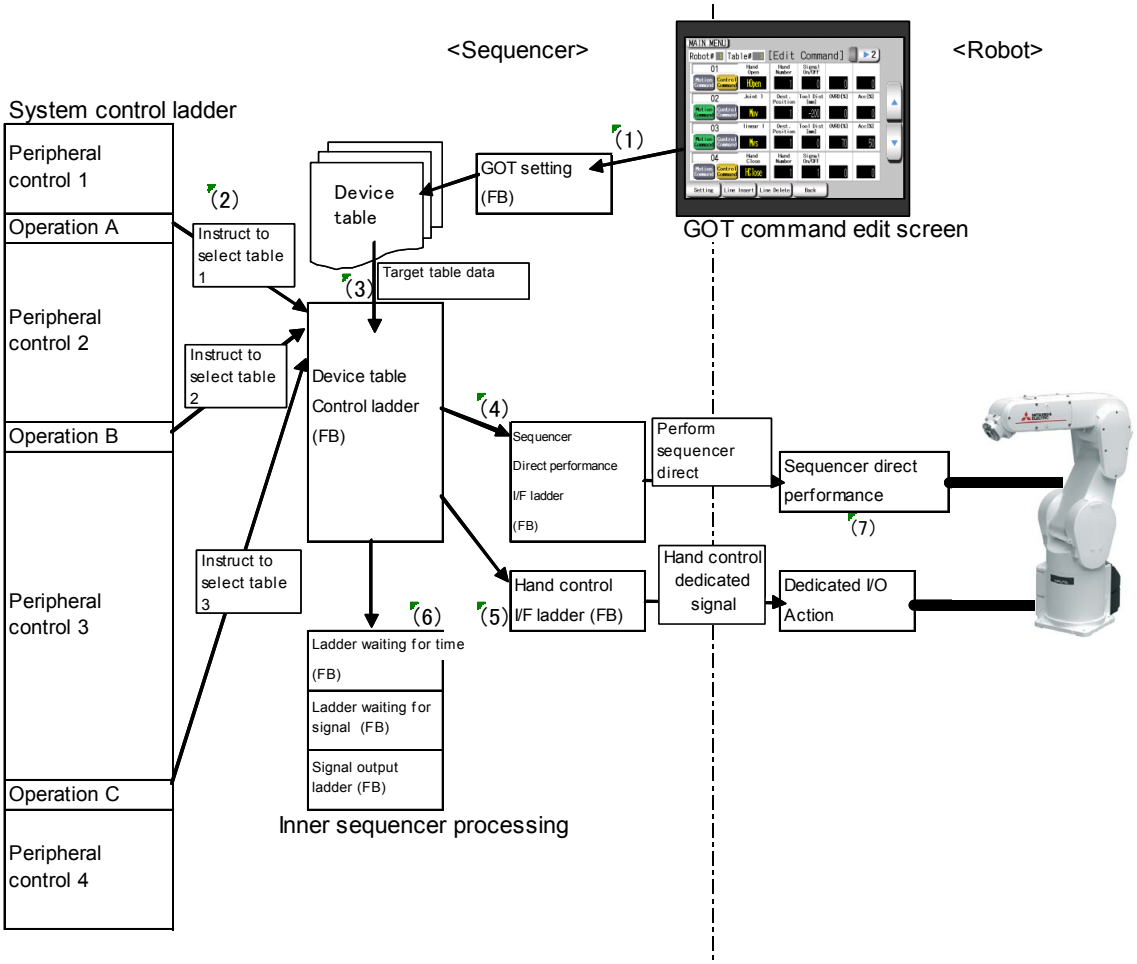

101
M202
Move above pickout position
M252
SQ-direct performed successfully
M204
Move to pickout position
M203
Arrive above pickout position
M203
106
M204
Move to pickout position
Ovrd 70
Accel 50,50
Mvs P_DM(1)
111
M205
Arrive to pickout position
M206
Hand Close & Confirmation
K2
T10
Confirmation(pickout position)
Dly 0.2
121
M206
Hand Close & Confirmation
HClose 1
K1
T20
Delay for Hand Close
B39
Hand Input Signal 901
B38
Hand Input Signal 900
Wait M_In(901)=1
Dly 0.1
132
M207
Move above pickout position
Ovrd 100
Accel 100,100
Mvs P_DM(1),-200
137
M208
Arrive above pickout position
M209
Transportation completion
Write an operation flow
Prepare performance
142
M209
Transportation completion
End
146
MCR
N1


Create data for SQ Direct
Create com-
mand data of
sequencer direct
performance
147
SM400
Always O
N
FMOV
K0
D104
Command
No.
K15
167
M202
Move abo
ve picko
ut posit
ion
M5
System r
unning
M250
SQ-direc
t perfor
mance tr
igger
M204
Move to
pickout
position
M202
Move abo
ve picko
ut posit
ion
MOV
K1
D104
Command
No.
M207
Move abo
ve picko
ut posit
ion
M204
Move to
pickout
position
MOV
K11
D104
Command
No.
M207
Move abo
ve picko
ut posit
ion
MOV
K1
D105
Command
Data1
M202
Move abo
ve picko
ut posit
ion
MOV
K-2000
D106
Command
Data2
M207
Move abo
ve picko
ut posit
ion
M204
Move to
pickout
position
MOV
K70
D110
Override
[%]
MOV
K50
D111
Accelera
tion rat
e[%]
MOV
K50
D112
Decelera
tion rat
e[%]


SQ Direct I/F
198
M250
SQ-direct performance trigger
M251
SQ-direct performed
U3E1¥HG520.0
Command Completion Signal
M240
SQ-direct practicable condition
BMOVP
D104
Command No.
U3E0¥HG524
Command No.
K15
U3E0¥HG520.0
Command Request Signal
U3E0¥HG520.0
Command Request Signal
U3E1¥HG520.0
Command Completion Signal
U3E0¥HG520.0
Command Request Signal
M251
SQ-direct performed
M251
SQ-direct performed
MOVP
U3E1¥HG522
Completion Status
D20
Completion Status
=
D20
Completion Status
K1
M252
SQ-direct performed successfully
=
D20
Completion Status
K6
M253
SQ-direct performance suspended
=
D20
Completion Status
K30
M252
SQ-direct performed successfully
M253
SQ-direct performance suspended
M254
SQ-direct performance error exit
I/F handling of sequencer direct performance
Hand Open/Close I/F
252
M201
Hand Open
M206
Hand Close & Confirmation
B4F
Hand Control permission output
M5
System running
RST
B151
Hand output control signal(901)
SET
B150
Hand output control signal(900)
270
M206
Hand Close & Confirmation
M201
Hand Open
B4F
Hand Control permission output
M5
System running
RST
B150
Hand output control signal(900)
SET
B151
Hand output control signal(901)
Hand open/close I/F handling
276
B4F
Hand Control permission output
MOV
K2B30
Hand Output State 900
K2B150
Hand output control signal(900)
Batch transfer the data in B100 and after to the robot input signal
280
SM400
Always ON
BMOV
K4B100
STOP
U3E0¥HG0
General-purpose input
K6

8.4.3 Sample Operation Setting in GOT Screen

Realizes a robot operation without user program by entering the robot operation into GOT.
Sequencer handling is provided by function block or ladder program. Also GOT screen is provided. (Refer to MITSUBISHI ELECTRIC FA Site.)

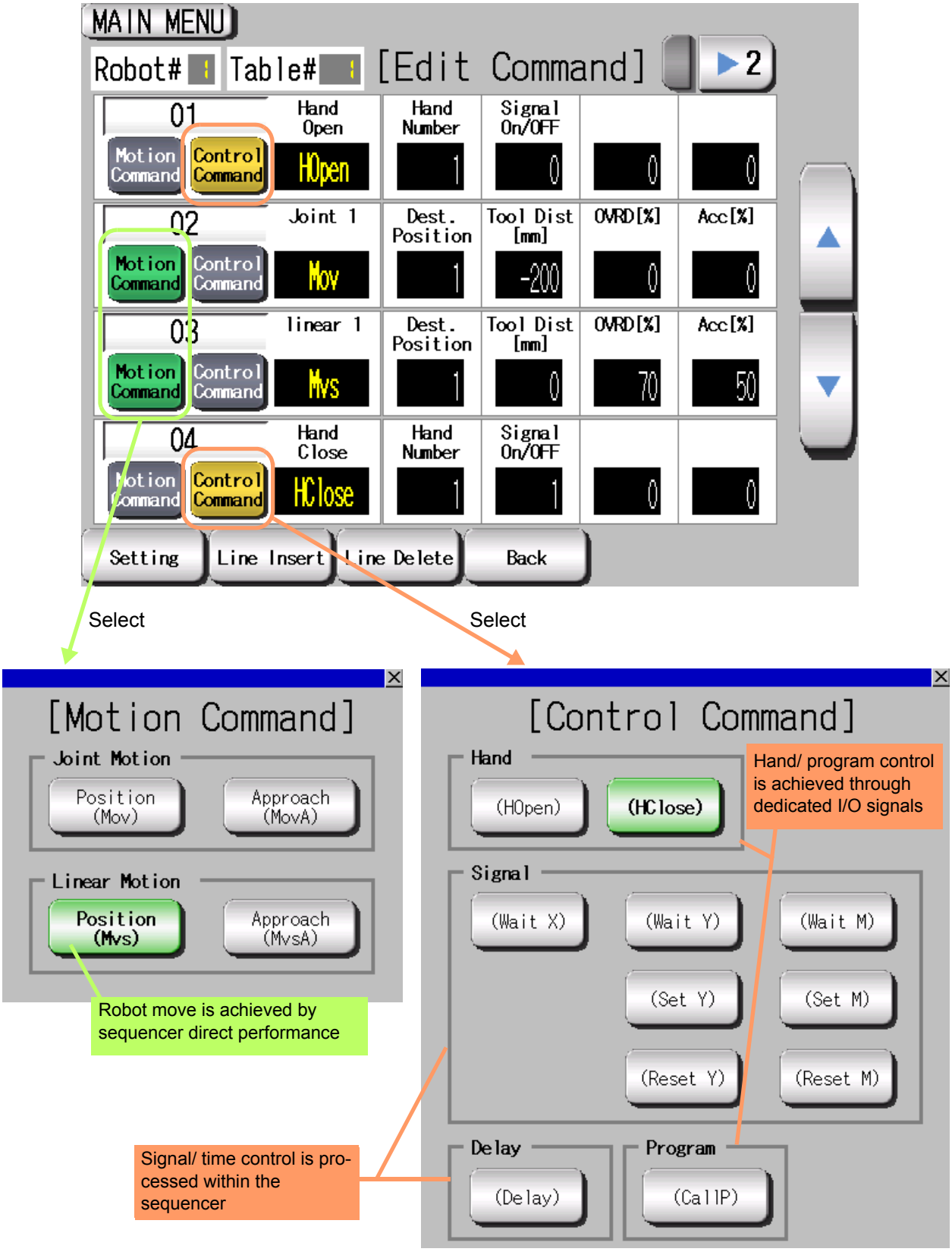


(1) Input robot control commands sequentially in GOT.
The input data is stored in the sequencer device table.
(2) When a robot operation is generated in the system control, the sequencer calls up the device table control by designating the table number.
(3) As for the device table control, the sequencer reads the setting values for target device table and carries out the control steps (4) - (6) in order according to the setting values.
(4) As for robot movement, the sequencer sends the instruction of sequencer direct performance to the robot based on the values in device table.
(5) As for hand control, the sequencer sends the dedicated signals for hand control.
(6) As for inner sequencer processing, the sequencer carries out the sequencer's target operation.
(7) The robot carries out operations instructed by sequencer direct performance and hand control.

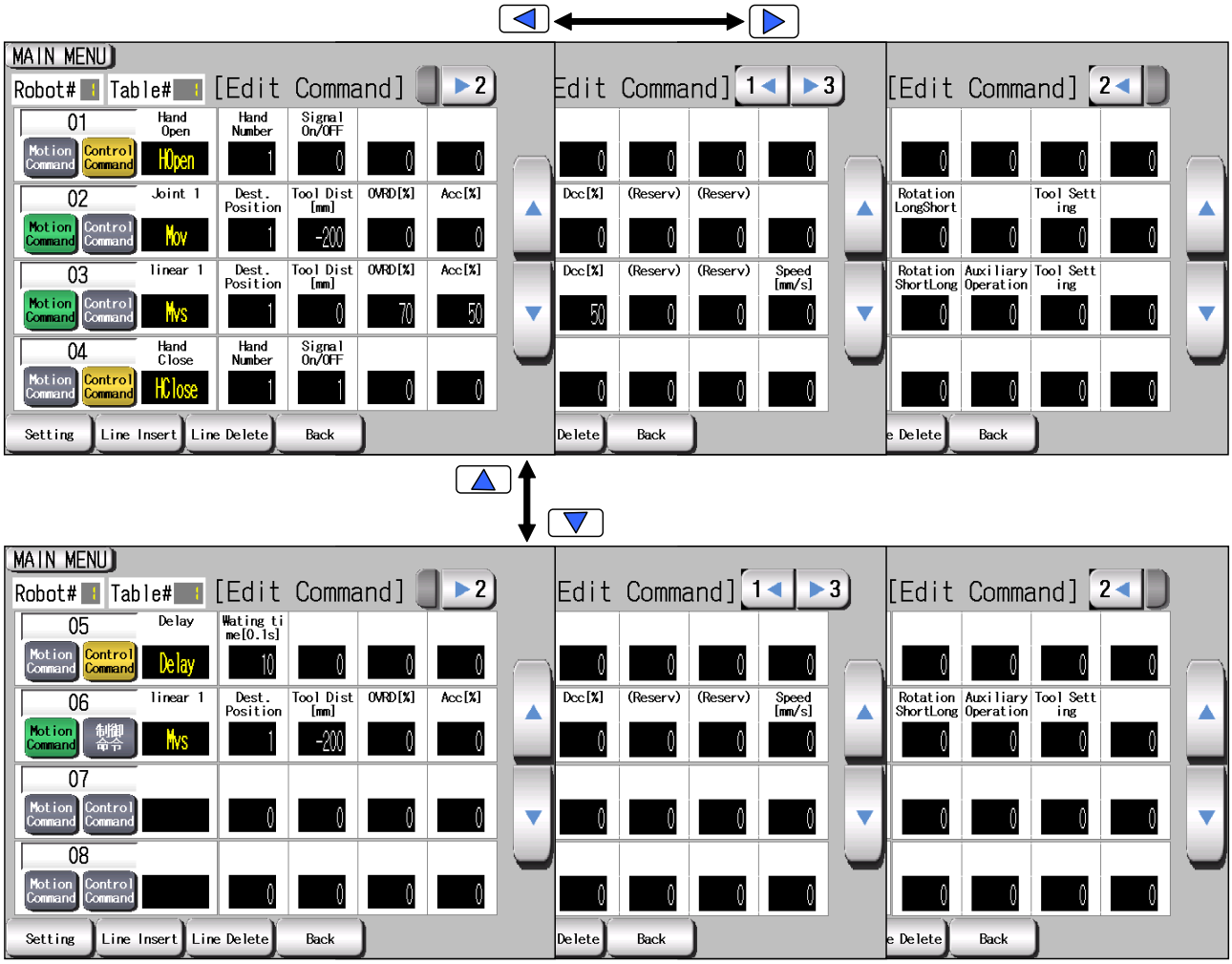
<S/W for robot operation>

Item	Target	Program to be prepared
Program created by the customer	Sequencer	**None**
	Robot	**None**
Function block and screen provided by us	Sequencer	● GOT setting ● Device table control ● Inner sequencer processing (waiting for time and signal, signal output, etc.) ● I/F handling of sequencer direct performance ● Hand control I/F handling
	GOT	● Command edit screen

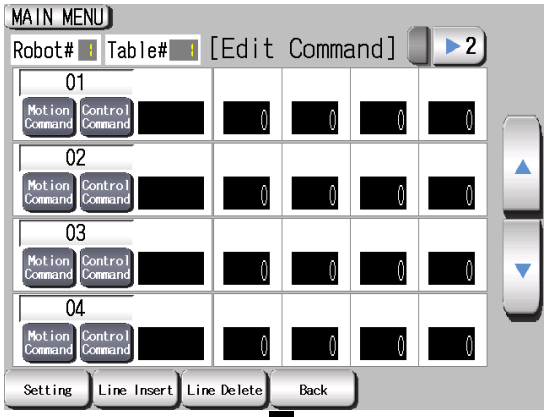
[GOT screen image (sample)]
In the GOT screen, enter setting values, such as operation command, position, speed, acceleration, to operate the robot based on the specified steps. The figure below shows an example of operation command input screen.



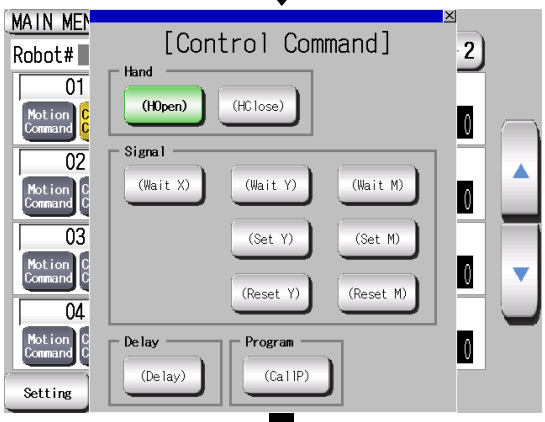
The figure below shows the change of pages when entering an operation command in the GOT screen.



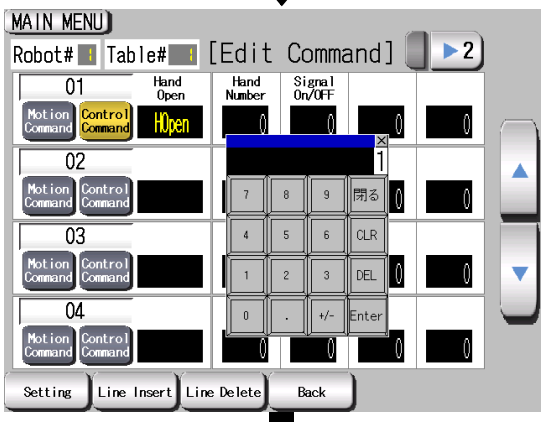
<Operating procedure>



1) Select target robot and table number to display the command edit screen.

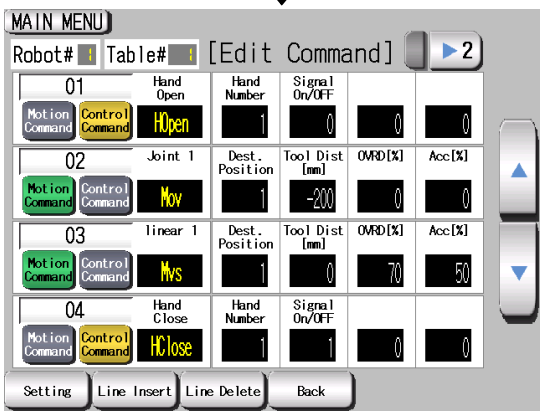


2) Push the [Action cmd]/[Control cmd] button to display the command select screen and select a command.



3) Display the numeric keypad by pushing a number display to enter the data necessary for the command.

* Necessary data may vary according to the command. Enter the data according to the guidance display at the top of number display.



4) Carry out the steps "2)" and "3)" as many as the number of commands to end the editing.

8.5 Precautions for Sequencer Direct Performance

8.5.1 Requirements

Sequencer direct performance can be carried out when all conditions below are met:

(1) Valid operation authority (robot output IOENA is ON)
(2) No H or L level error (robot signals HLVLERR, LLVLERR are OFF)
(3) Program is available (robot output SLOTINIT is ON)
(4) Robot servo is ON (robot output SRVON is ON)
(5) No stop input (robot output STOPSTS is OFF)
(6) Not returning to retracting point (robot output SAFEPOS is OFF)
(7) Remote JOG is not working (robot output JOGENA is OFF)
(8) Robot's origin has been set up
(9) Parameter ALWENA is set to zero (refer to Page 123, "8.5.3 Prohibit Program Startup with always Running Program")
(10) Robot language is set to "MELFA-BASIC V" (parameter RLNG=2) or "MELFA-BASIC VI" (parameter RLNG=3 (Only RT ToolBox3)) (Refer to Page 123, "8.5.4 Robot Language Setting")

8.5.2 Running together with Program

Even when sequencer direct performance function is valid, the program startup through external signal is possible. However, they cannot run simultaneously. The Table 8-3 lists whether it is possible to run each program simultaneously.

Table 8-3:Possibility to run each program as well as sequencer direct performance

Item	Decision	Description
Start up a program with startup condition START simultaneously	×	• START program is unavailable during performing sequencer direct • Sequencer direct is unavailable during running START program
Start up a program with startup condition ALWAYS simultaneously	○	• Sequencer direct is available during running ALWAYS program
Start up a program with startup condition ERROR simultaneously	△	• ERROR program is available during performing sequencer direct • Sequencer direct is unavailable during running ERROR program

The robot program and the position edit/variable monitor can the handle position data for sequencer direct performance.

The table below lists the handling of each position:

Positon No	Variable Name Used in Program	Function	Reference as a Command Note1)	Definition as a Command Note2)	Display Variable	Teach/ Edit Variable	Delete Variable
1 - 999	P_DM(1) - P_DM(999)	External program variable	○	×	○	○	×
5000	P_D5000	System state variable	○	×	○	×	×
5100 - 5102	P_D5100 - P_D5102						

Note1) Mov P_DM(1), etc.
Note2) Def Pos P_DM, etc.

8.5.3 Prohibit Program Startup with always Running Program

When the sequencer direct performance function is valid, ALWENA=0 is assumed regardless of the parameter ALWENA setting (X**, SERVO, RESET ERR commands in the always running program are prohibited). When the controller starts up with parameter ALWENA=1 while the sequencer direct performance function is valid, an **error "L3995 Unavailable together with the function (sequencer direct, ALWENA)"** occurs.

8.5.4 Robot Language Setting

The sequencer direct performance is enabled only when MELFA-BASIC V (parameter RLNG=2) or MELFA-BASIC VI (parameter RLNG=3 (only RT ToolBox3)) is selected for robot language . When MELFA-BASIC IV is selected (parameter RLNG=1), an **error "L3996 Sequencer direct function unavailable"** occurs on controller startup. This error cannot be reset.

9 Cooperative operation function

9.1 Outline

The cooperative operation function by two robots enables the transportation that two robots grasp the target workpiece at one end, respectively, together in synchronization.
A position-tracking control of robots enables this operation. After the common coordinates are set in a master robot (robot No.1) and a slave robot (robot No.2), robot No.2 obtains the current position data of robot No.1 every controller control time via a PLC, and tracks robot No.1 operation.
The controller control time is approx. 3.5 ms with the CR800-R, and approx. 7.1 ms with CR800-Q.

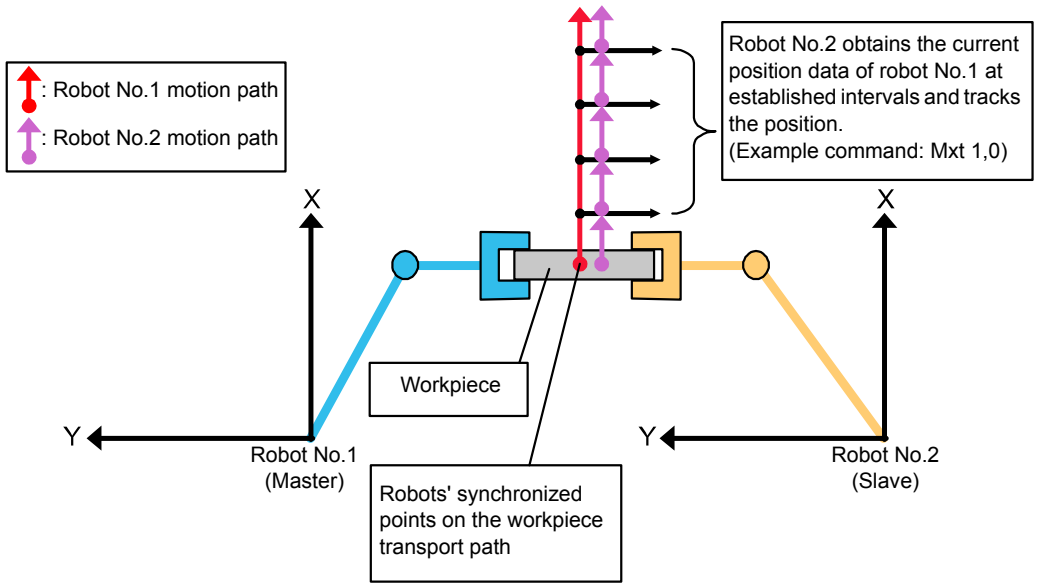


Fig.9-1:Cooperative operation

⚠ CAUTION The accuracy of the transport path by both robots operation and the tracking intervals of robot No.2 are not guaranteed.
When the operation speed of robot No.1 is set to high, the operation of robot No.2 may be delayed.

9.2 Parameter of Selecting CPU buffer Memory Extended Function

Parameter	Parameter Name	Array Qty Character Qty	Description	Factory Default
Select CPU buffer memory extended function	IQMEM	1 digit integer	Set validity (1)/ invalidity (0) for the function. Sets each bit by allocating a function to each bit. 0000000000000000 bit2-15: Not used |+- bit0: Use the extended function +-- bit1: Sequencer direct performance function	0000000000000000

9.3 System configuration

This is a multi-CPU system configured with three CPU modules; a PLC CPU module and two robot CPU modules.
A position data is transmitted between robots through the CPU buffer memory by the PLC.

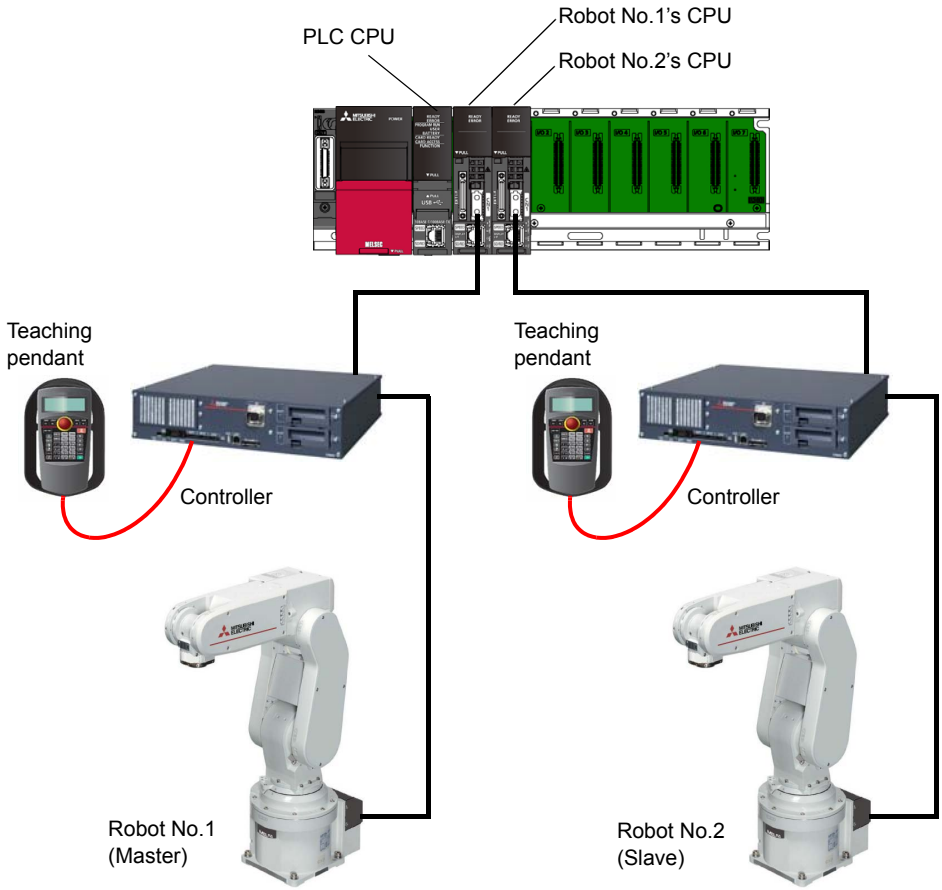


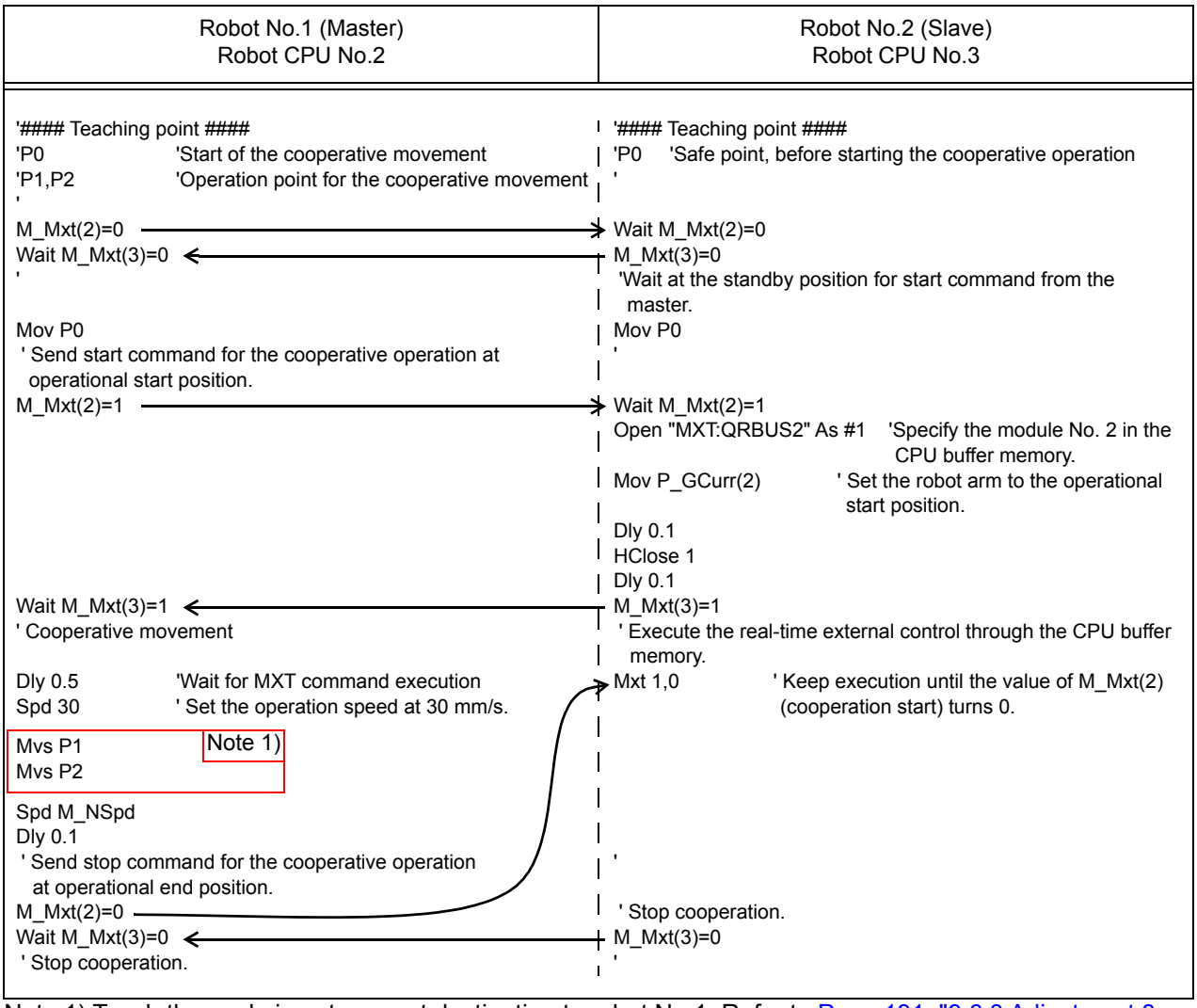
Fig.9-2:System configuration

9.4 Sample program

Sample programs are shown below.

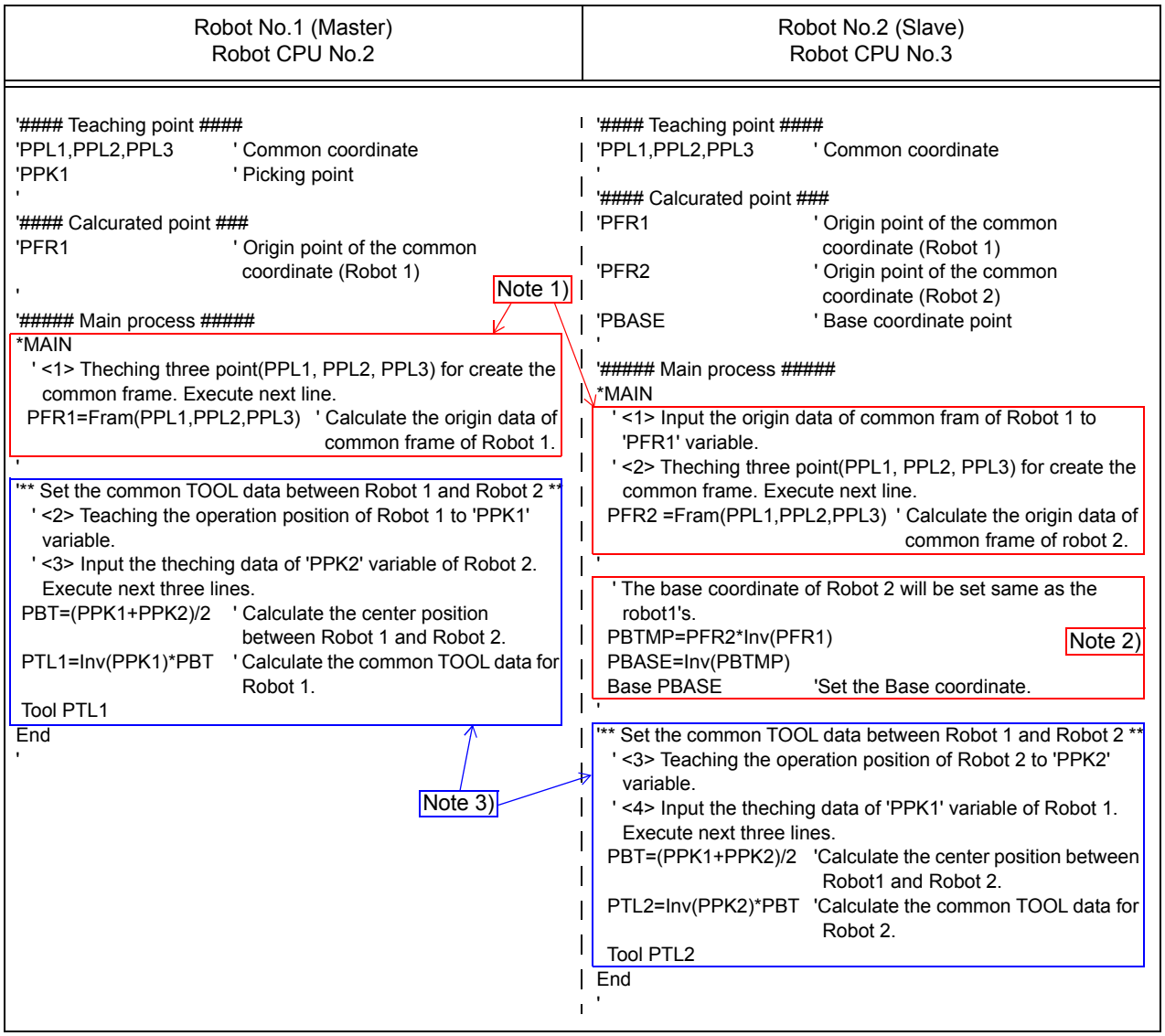
Robot	Program	Content
Robot 1 (Master)	1.prg	A program for automatic operation of the robot (actuation between two teaching positions)
	BFRM.prg	A program for adjustment of setting the common coordinates in robot No.1 and robot No.2
Robot 2 (Slave)	1.prg	A program for automatic operation of the robot (tracking robot No.1 to its destination)
	BFRM.prg	A program for adjustment of setting the common coordinates in robot No.1 and robot No.2

9.4.1 Automatic operation program (1.prg)



Note 1) Teach the workpiece transport destination to robot No.1. Refer to Page 131, "9.6.3 Adjustment 3: Teaching and parameter setting of the workpiece transport destination" for details.

9.4.2 Robot adjustment program (BFRM.prg)



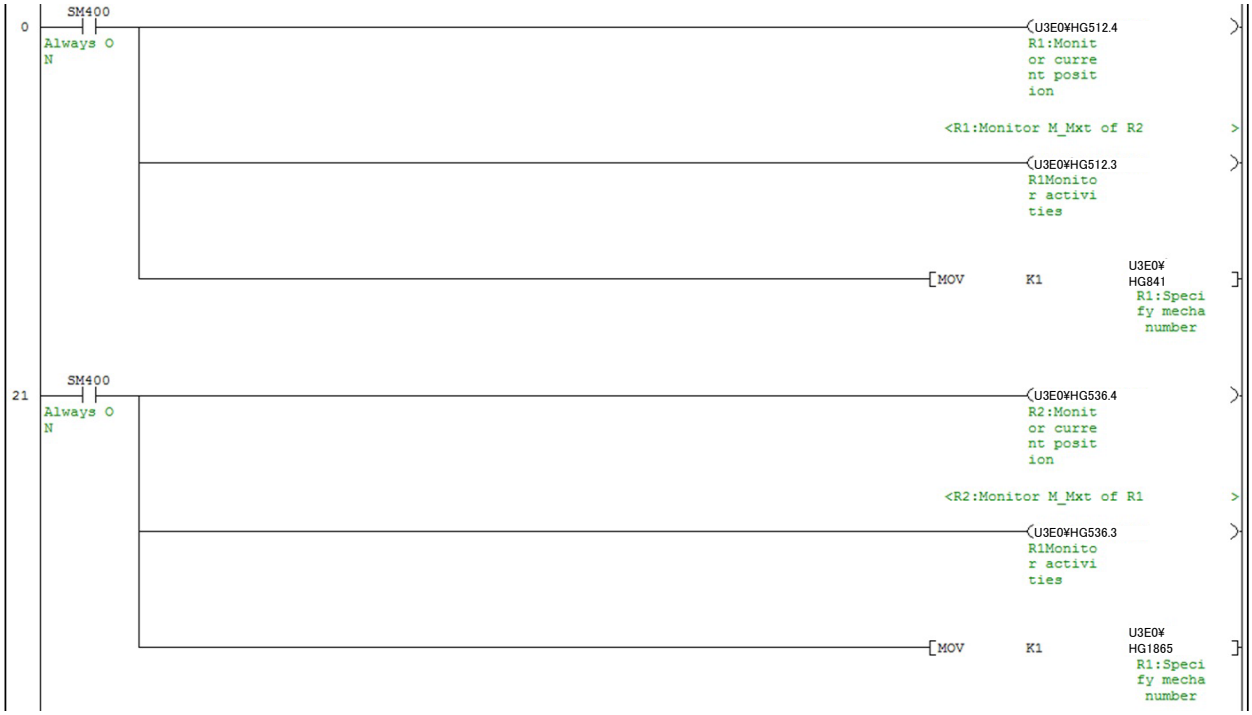
Note 1) Set the common frame coordinates to robot No.1 and robot No.2. Refer to Page 129, "9.6.1 Adjustment 1: Adjustment of the common base coordinates" for details.

Note 2) Use the setting of the robot No.1 base coordinates for robot No.2 base coordinates. Refer to Page 129, "9.6.1 Adjustment 1: Adjustment of the common base coordinates" for details.

Note 3) Set the common tool in the robots. Refer to Page 130, "9.6.2 Adjustment 2: Setting of the common tool" for details.

9.5 Sequence program

Use the PLC program to output the data of the robot No.1 current position to the CPU buffer memory.
(It is required to enable the extended function parameter IQMEM for robot No.1 and robot No.2.)



9.6 Adjustment

9.6.1 Adjustment 1: Adjustment of the common base coordinates

Use the robot program BFRM.prg to set the common coordinates in robot No.1 and robot No.2. (Use the setting of the robot No.1 base coordinates for robot No.2 base coordinates.)

■Outline of setting procedure

1) Set the position data (PPL1, PPL2, and PPL3) to define the common frame coordinates in robot No.1 and robot No.2. (Specify the common position data in the common frame coordinates of robot No.1 and robot No.2.)
2) Set the common frame coordinates in robot No.1 and robot No.2.
   <Program>
   PFR1=Fram(PPL1,PPL2,PPL3)　　'Calculate the origin data of common frame of robot No.1.
   PFR2=Fram(PPL1,PPL2,PPL3)　　'Calculate the origin data of common frame of robot No.2.
3) Use the setting of the robot No.1 base coordinates for robot No.2 base coordinates. (setting on robot No.2 only)
   <Program>
   Share the setting of the robot No.1 base coordinates with robot No.2.
   PBTMP=PFR2*Inv(PFR1)
   PBASE=Inv(PBTMP)
   Base PBASE　　'Set the base coordinate.

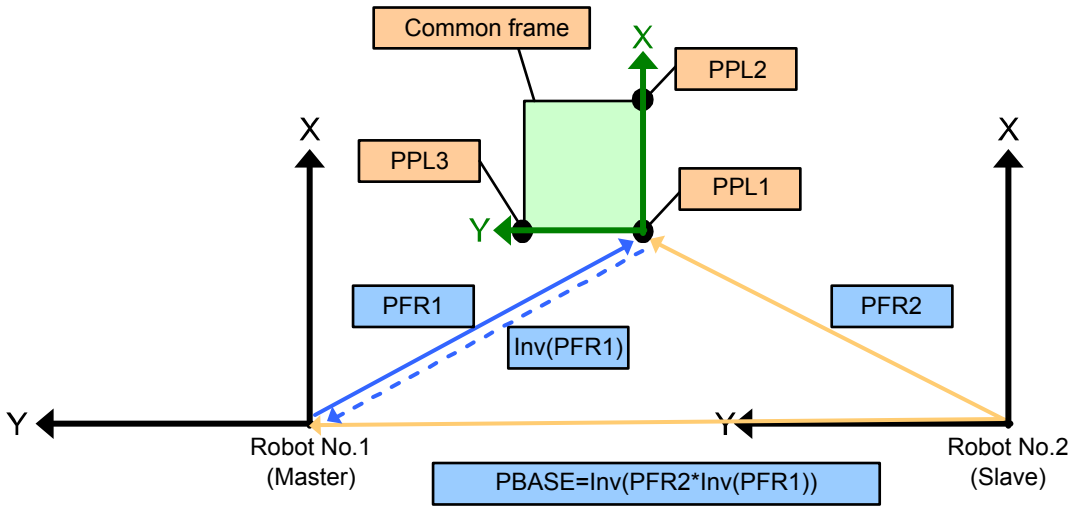


Fig.9-3:Adjustment of the common base coordinates

9.6.2 Adjustment 2: Setting of the common tool

Use the robot program BFRM.prg to set the common tool in robot No.1 and robot No.2. (Establish the tool position at midpoint between the workpiece grasp positions for robot No.1 and robot No.2.)

■Outline of setting procedure

1) Teach the workpiece grasp positions for robot No.1 and robot No.2. (PPK1, PPK2)
   (PPK1: Workpiece picking position of robot 1. PPK2: Workpiece picking position of robot 2.)
2) Set the tool coordinate of robot 1.
   <Program>

```
PBT=(PPK1+PPK2)/2          ' Determine the midpoint by calculation between the work-
                             piece removing positions for robot No.1 and robot No.2.
PTL=Inv(PPK1)*PBT          ' Determine the common tool by calculation in robot No.1 and
                             robot No.2.
Tool PTL
```

3) Set the tool coordinate of robot 2.
   <Program>

```
PBT=(PPK1+PPK2)/2          ' Determine the midpoint by calculation between the work-
                             piece removing positions for robot No.1 and robot No.2.
PTL=Inv(PPK2)*PBT          ' Determine the common tool by calculation in robot No.1 and
                             robot No.2.
Tool PTL
```

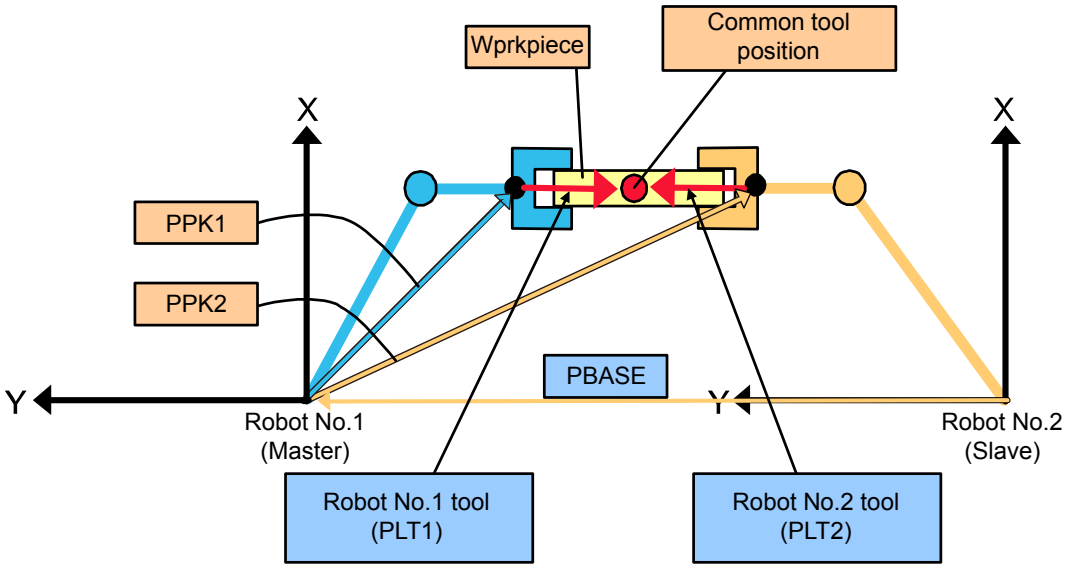


Fig.9-4:Setting of the common tool

9.6.3 Adjustment 3: Teaching and parameter setting of the workpiece transport destination

Use the robot program 1.prg for robot No.1 to teach the workpiece transport destination (to robot No.1 only).
Set the parameter to enable each extended function for robot No.1 and robot No.2.

■Outline of setting procedure

1) Teach the workpiece transport destination to robot No.1. (P1, P2)
2) Set the parameter to enable each extended function for robot No.1 and robot No.2.
   (Change [1: 0000000000000000] to [1: 0000000000000011].)

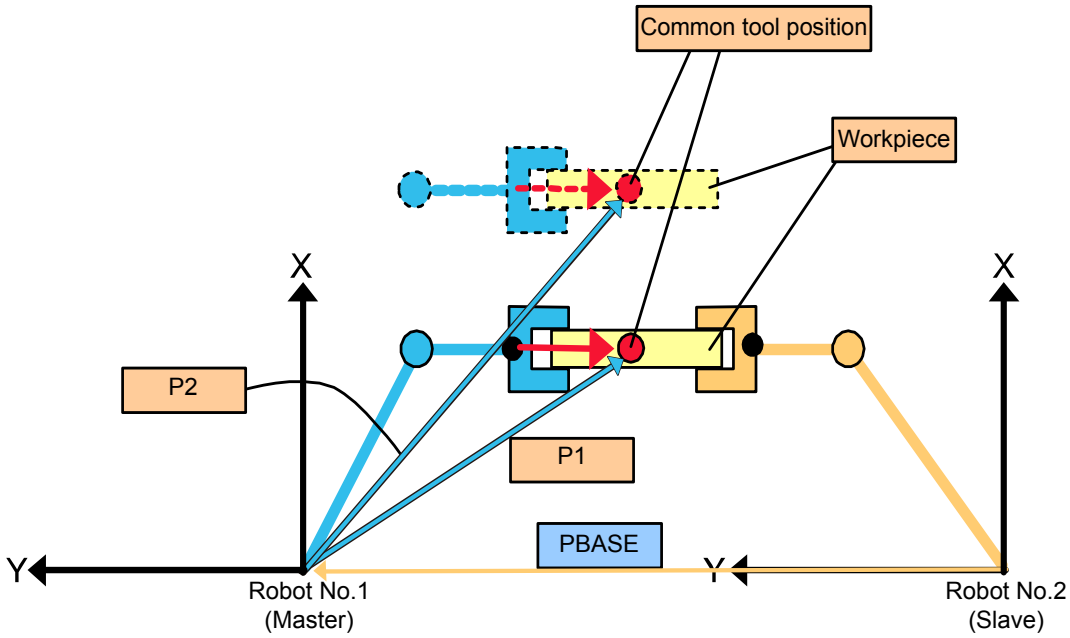


Fig.9-5:Teaching and parameter setting of the workpiece transport destination

9.7 Cooperated operation by robots

9.7.1 Parallel straight motion by robot No.1

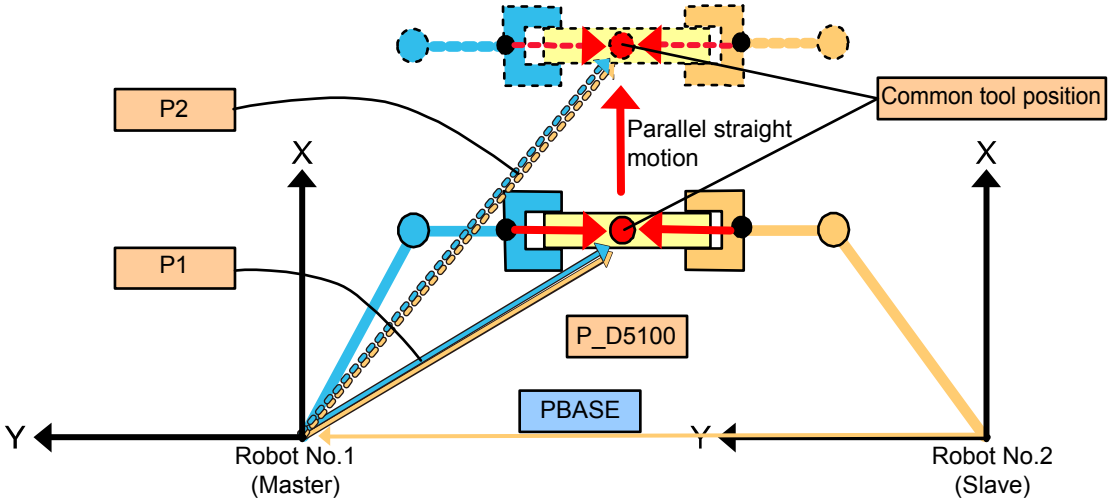


Fig.9-6:Parallel straight motion

9.7.2 Rotating motion by robot No.1

Robot No.1 and robot No.2 rotate the workpiece about the common tool position as a center of rotation.

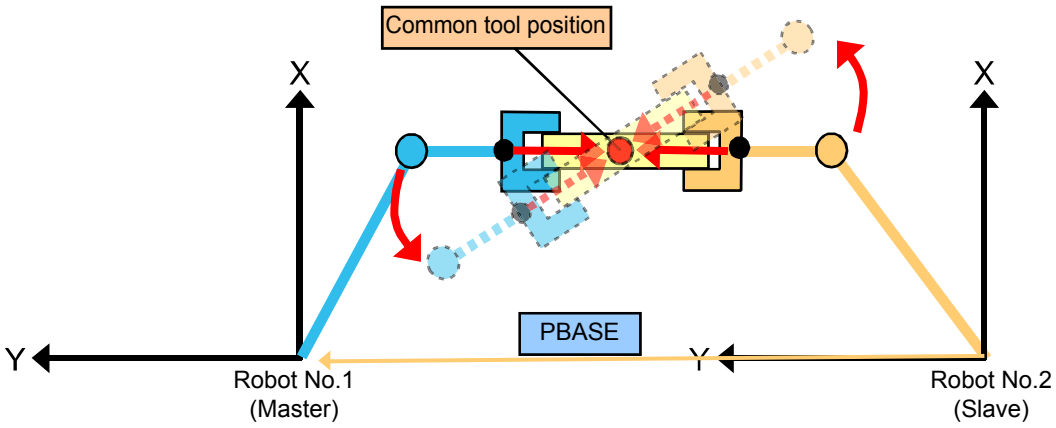


Fig.9-7:Rotating motion

10 CPU buffer Memory Extended Function Relevant Parameter

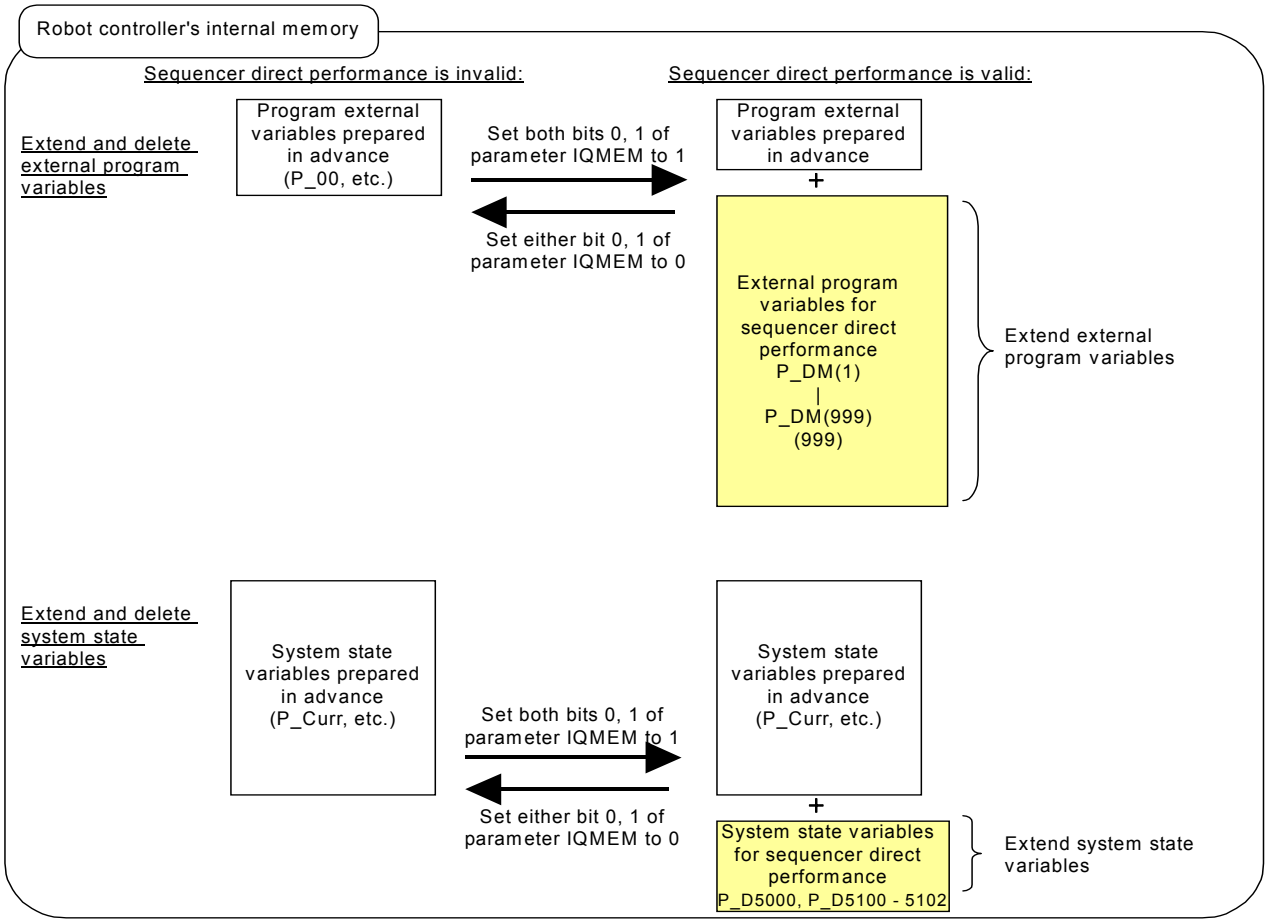
Here, describes a parameter relating to CPU buffer memory extended function.

When a parameter is changed, make sure to turn on again the robot controller' power supply (OFF to ON) or reset the sequencer. The parameter setting does not become effective only with the parameter change.

10.1 Parameter of Selecting CPU buffer Memory Extended Function

Parameter	Parameter Name	Array Qty Character Qty	Description	Factory Default
Select CPU buffer memory extended function	IQMEM	1 digit integer	Set validity (1)/ invalidity (0) for the function. Sets each bit by allocating a function to each bit. 0000000000000000 bit2-15: Not used |+- bit0: Use the extended function +-- bit1: Sequencer direct performance function	0000000000000000

When sequencer direct performance is valid, external program variables P_DM(1) - P_DM(999) (999 in total) in the robot controller and system state variables P_D5000, P_D5100 - P_D5102 are extended (extended variables are referred to as "extended variable", below).



(1) Variable extension timing
When the controller is started up while the sequencer direct performance function is valid (both bits 0, 1 of parameter IQMEM for selecting CPU buffer memory extended function are set to one), external program variables and system state variables are extended.

User defined external variable with the same name is used:
When the user defined external variable with the same name is in use, an **error "L4811 User defined external variable redefinition error"** occurs, but the variables for sequencer direct performance are extended. However, because already created user defined external variable still remains, there are two variables with same name. In this case, because the extended variables for sequencer direct performance is preferred for variable reference, the values of extended variables for sequencer direct performance are displayed in the variable monitor.
When the sequencer direct performance gets invalid, the user defined external variable becomes available as before.

Out of memory:
In case of out of memory, **errors "C7010 Out of memory"**, **"L4800 System based program unavailable"** occur. Here, because the system based program itself is not created, the variables are not extended.

Change of parameter PRGGBL (external variable extension)
When the system starts up while sequencer direct performance function is valid, this parameter PRGGBL is changed to one (valid) even if PRGGBL was set to zero (invalid). The factory default of PRGGBL is one, but when its value was changed by the user, the valid setting of sequencer direct performance changes the value.

(2) Deletion timing of extended variables
When the controller is started up while the sequencer direct performance function is invalid (either bits 0, 1 of parameter IQMEM for selecting CPU buffer memory extended function is set to zero), extended external program variables and extended system state variables are deleted. The user available memory space increases.

(3) Add controller program check
When the definition (Def Pos P_DM, etc.) of extended variable is tried to be entered into the controller while sequencer direct performance function is valid, an error "4350 Duplicated variable definition tried" occurs.

(4) Program syntax check
The definition (Def Pos P_DM, etc.) error of extended variable is not checked by syntax check in the RT ToolBox3 (because the syntax check of RT ToolBox3 cannot determine whether the sequencer direct performance function is valid or not). Its error is checked by the controller's program check during transferring the program to the controller.

(5) Treatment when sequencer direct is performed while the variables were not extended
The sequencer direct performance is not received without variable extension.
The completion status is set to the number indicating an impracticable reason (refer to Page 91, "(2) Operation Command Is Impracticable:").

10.2 Function Definition Parameter

Parameter	Parameter Name	Array Qty Character Qty	Description	Factory Default
Define function	IQSPEC	1 digit integer	Set up function for robots. Set each function allocated by each bit. 0000000000000000 bit1-15: Not used +--- bit0: Direction to write into CPU buffer memory 0: Reads/writes in order from first to last address (until Ver. N7) 1: Reads in order from first to last address, writes in order from last to first address (communication specification among iQ Platform multiple CPUs)	0000000000000001

The access sequence of the CPU buffer memory before the software version N7 of the robot controller is direction to the final address from the top address for both of reading and writing. However, the sequencer's communication specification among iQ Platform multiple CPUs is direction from last to start address for writing. Thus, when a system is designed according to the CPU buffer memory map specification, the interlock of dataset may be impossible. (For more information, refer to the Fig. 10-1.)
Therefore, when utilizing CPU buffer memory expanded function, it is necessary to make the CPU buffer memory access order the same as the specification of the sequencer. We provide the parameter (IQSPEC) to solve it. The initial value is set to the same specification as the sequencer, so its change by customer is not necessary at all. But, in order to assure the compatibility with previous models, the behavior based on the previous specification is possible.

Prevention of separation of data over 32 bits

When user's free area is used

The program reads in order from start of user's free area. In write command, the transmission data is written in order from last to start address of user's free area.
Consequently, the interlock device at the start of data for communication can prevent separation of data for communication

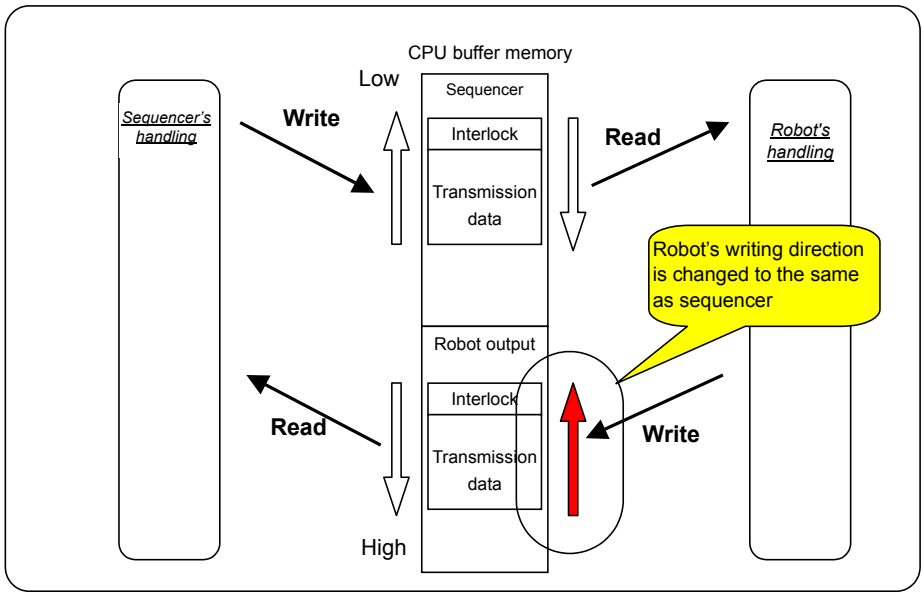


Fig.10-1:Change the writing order of CPU buffer memory data

11 Extended Function Relevant Error List

(1) Error occurred when MELFA-BASIC Ⅳ is selected while CPU buffer memory extended function is valid

Error No	Error Cause and Measure	
L3994	Error message	Shared memory extended function unavailable (MB4)
	Cause	CPU buffer memory extended function is unavailable in MELFA-BASIC IV. The parameter RLNG=1 (MELFA-BASIC IV) is selected while CPU buffer memory extended function is valid. Make sure to set the parameter RLNG to 2 (MELFA-BASIC V) or 3 (MELFA-BASIC VI (only RT ToolBox3)).
	Measure	Set the parameter RLNG to 2 (MELFA-BASIC V) or 3 (MELFA-BASIC VI (only RT ToolBox3)).

(2) Error occurred when the parameter ALWENA is set to one (enabled) while the sequencer direct performance is valid

Error No	Error Cause and Measure	
L3995	Error message	Unavailable together with the function (sequencer direct, ALWENA)
	Cause	Unavailable together with the function (sequencer direct, ALWENA) The parameter ALWENA is set to one (enabled) while the sequencer direct performance function is valid. During performing sequencer direct, X** commands are unavailable in the always running program. Make sure to set the parameter ALWENA to zero (disabled).
	Measure	Set the parameter ALWENA to zero (disabled).

(3) Error occurred when MELFA-BASIC Ⅳ is selected while the sequencer direct performance is valid

Error No	Error Cause and Measure	
L3996	Error message	Sequencer direct function unavailable (MB4)
	Cause	Sequencer direct function is unavailable in MELFA-BASIC IV. The parameter RLNG=1 (MELFA-BASIC IV) is selected while the sequencer direct performance function is valid. Make sure to set the parameter RLNG to 2 (MELFA-BASIC V) or 3 (MELFA-BASIC VI (only RT ToolBox3))
	Measure	Set the parameter RLNG to 2 (MELFA-BASIC V) or 3 (MELFA-BASIC VI (only RT ToolBox3)).

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: TOKYO BUILDING, 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 5-1-14, YADA-MINAMI, HIGASHI-KU NAGOYA 461-8670, JAPAN
Authorised representative:
Mitsubishi Electric Europe B.V. FA - European Business Group
Mitsubishi-Electric-Platz 1, D-40882 Ratingen, Germany
Tel: +49(0)2102-4860
Dec., 2018 MEE Printed in Japan on recycled paper.