EPSON

EPSON RC+ 7.0

Remote Control Reference

Rev.3

EM145S2711F

EPSON RC+ 7.0   Remore Contorol Reference Rev.3

EPSON RC+ 7.0

Remote Control Reference

Rev.3

FOREWORD

Thank you for purchasing our robot products. This manual contains the information necessary for the correct use of the EPSON RC+ software.
Please carefully read this manual and other related manuals when using this software.
Keep this manual in a handy location for easy access at all times.

WARRANTY

The robot and its optional parts are shipped to our customers only after being subjected to the strictest quality controls, tests and inspections to certify its compliance with our high performance standards.
Product malfunctions resulting from normal handling or operation will be repaired free of charge during the normal warranty period. (Please ask your Regional Sales Office for warranty period information.)
However, customers will be charged for repairs in the following cases (even if they occur during the warranty period):

1. Damage or malfunction caused by improper use which is not described in the manual, or careless use.
2. Malfunctions caused by customers' unauthorized disassembly.
3. Damage due to improper adjustments or unauthorized repair attempts.
4. Damage caused by natural disasters such as earthquake, flood, etc.

Warnings, Cautions, Usage:

1. If the robot or associated equipment is used outside of the usage conditions and product specifications described in the manuals, this warranty is void.
2. If you do not follow the WARNINGS and CAUTIONS in this manual, we cannot be responsible for any malfunction or accident, even if the result is injury or death.
3. We cannot foresee all possible dangers and consequences. Therefore, this manual cannot warn the user of all possible hazards.

TRADEMARKS

TRADEMARK NOTATION IN THIS MANUAL

Microsoft® Windows® XP Operating system
Microsoft® Windows® Vista Operating system
Microsoft® Windows® 7 Operating system
Microsoft® Windows® 8 Operating system
Microsoft® Windows® 10 Operating system
Throughout this manual, Windows XP, Windows Vista, Windows 7, Windows 8, and Windows 10 refer to above respective operating systems. In some cases, Windows refers generically to Windows XP, Windows Vista, Windows 7, Windows 8, and Windows 10.

NOTICE

The contents of this manual are subject to change without notice.
Please notify us if you should find any errors in this manual or if you have any comments regarding its contents.

MANUFACTURER

SEIKO EPSON CORPORATION

SAFETY PRECAUTIONS

Installation of robots and robotic equipment should only be performed by qualified personnel in accordance with national and local codes. Please carefully read this manual and other related manuals when using this software.

Keep this manual in a handy location for easy access at all times.

WARNING	This symbol indicates that a danger of possible serious injury or death exists if the associated instructions are not followed properly.
CAUTION	This symbol indicates that a danger of possible harm to people or physical damage to equipment and facilities exists if the associated instructions are not followed properly.

1. Before Reading This Manual

This manual contains information on how to use the remote I/O control extended function. This manual assumes that users have sufficient knowledge about our Robot Controllers. Before using this feature, be sure to read the contents of related manuals for the robot systems, and understand their function.

2. Main Features

- This function allows you to execute commands in the controller similar to SPEL commands using inputs and outputs. By selecting "Remote I/O" as a control device for EPSON RC+ 7.0 and configuring the appropriate I/O settings, the function can be used in addition to the standard Remote I/O.
- This function can be used with the Controller's standard inputs and outputs, and also with optional Fieldbus inputs and outputs (DeviceNet, PROFIBUS-DP, PROFINET, CC-Link, and EtherNet/IP, EtherCAT, Modbus).
- The following resources are provided for command execution:

  Handshake signals: Input/Output 7 bit port
  Command/response data signals: Up to 8 words (16 bits per word)
- Commands are categorized.
- Some data can be stored in tables and lists for more efficient command execution.

3. Overview

This function enables direct control of the robot system from external equipment by using discrete I/O or Fieldbus, without running any SPEL programs. The external equipment controls the robot system by setting commands in the selected Remote I/O space. Results of the commands can be acquired in the Remote I/O space selected for the response data.

CAUTION	■ A command is completed when you receive a response after sending a request. A new command cannot be requested until the response of the previous command is received. ■ A received command is executed even if the Ethernet is disconnected.

The provided command functions are based on EPSON RC+ 7.0 SPEL+. To use this function, also refer to *EPSON RC+ 7.0 SPEL+ Language Reference* manual. Please note that names of SPEL commands may be used in the descriptions of each command.

CAUTION	■ Although this function is based on EPSON RC+ 7.0 SPEL+ functions, it does not provide all of the EPSON RC+ 7.0 SPEL+ functions.

4. Remote I/O to Be Used

This function exchanges commands with external equipment using I/O described below.

4.1 Control signals
Handshake data
4.2 Data signals
Data (command, response) signals for exchanging control signals and information

4.1 Control signals

4.1.1 External Equipment Control Signals

Control signals output from the external equipment consist of the following three signals.

Name	Label	Description
Command set	ExtCmdSet	Requests the command execution Requests by setting the signal to High. Be sure to execute the command after setting the command data to the data field for preventing errors. This signal should be cleared after the Controller receives the command.
Response acquisition	ExtRespGet	Set this signal to High to notify the Controller that the response from the Controller is acquired. This signal should be cleared once the response set signal is cleared.
Function reset	ExtRESET	This signal initializes the interface function. Keep this signal to High while using the function. Function can not work in Low state. This signal also can be used to reset in case of interface function error or to abort the motion command halfway.

4.1.2 Controller Control Signals

Control signals output from the Controller consist of the following four signals.

Name	Label	Description
Command acquisition	ExtCmdGet	This signal outputs the command acquisition state of the Controller. (High=acquired) The signal can be cleared when the command set signal is cleared.
Response set signal	ExtRespSet	This signal is output when the response is set (High=Set)
Command result	ExtCmdResult	This signal outputs the command execution result. (High = error, Low = normal) Contents output to the response data vary according to the result of this signal.
Function error	ExtError	“High” will be output in case the this function cannot continue. (Normal = Low) At this point, an error code is output to the response data. The external equipment needs to judge the error code whether to reset the function or the controller. The function remains in halt state until either reset operation is done.

4.2 Data signals

4.2.1 Command Signals (ExtCmd0 to ExtCmd127)

This is the data used to specify a command and its associated parameters. Commands consist of several words.

(1) Word

A word consists of 16-bit port (16 bit).

(2) Command syntax

Commands consist of up to eight words. The number of words varies according to the command to be used. The minimum command consists of one word.

Command number	Parameter 1	Parameter 2	………	Parameter 7

4.2.2. Response Signals (ExtResp0 to ExtResp127)

This is the data for the command response. The response data consists of several words.

(1) Word

A word consists of a 16-bit port.

(2) Response syntax

Commands consist of up to eight words. The number of words varies according to the command to be used. The minimum command consists of one word. For error response, all commands use three words.

Command number	Response 1	Response 2	………	Response 7

CAUTION	■ Fields described as "Reserved" in descriptions of each command may be used in the future.

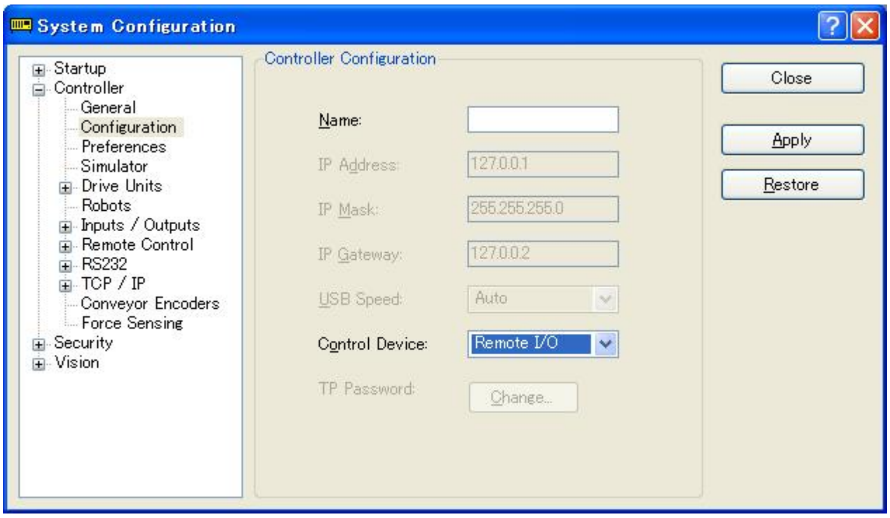
5. Configuration

To enable this function, you must configure the Controller beforehand. Set each signal described in *4. Remote I/O to be Used* in EPSON RC+ 7.0.

5.1 Selecting the Control Device

This function operates as one of the Remote I/O functions. To use this function, first select Remote I/O as the control device.

EPSON RC+ 7.0-[Setup]-[Controller]-[Configuration]-[Control device]

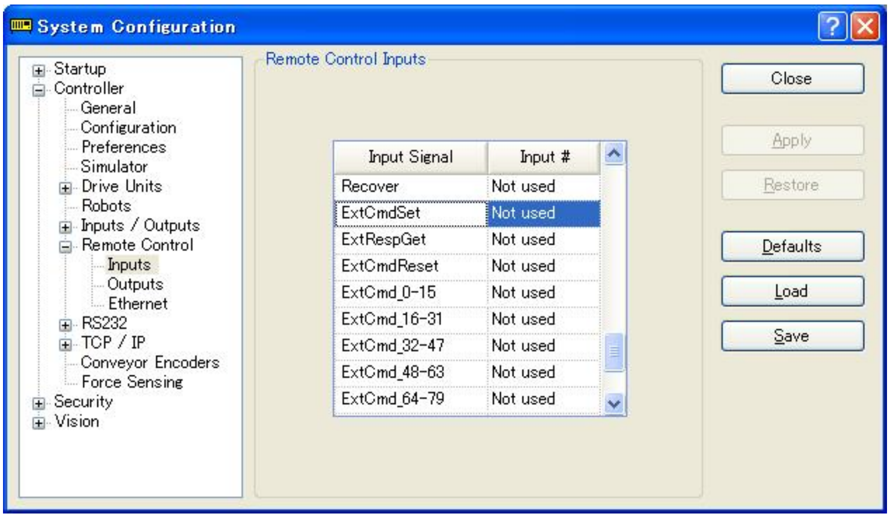


5.2 Setting the Control Signals

5.2.1 Setting the Input Signals

Set each signal controlled by the external equipment as an input signal of the Remote I/O.

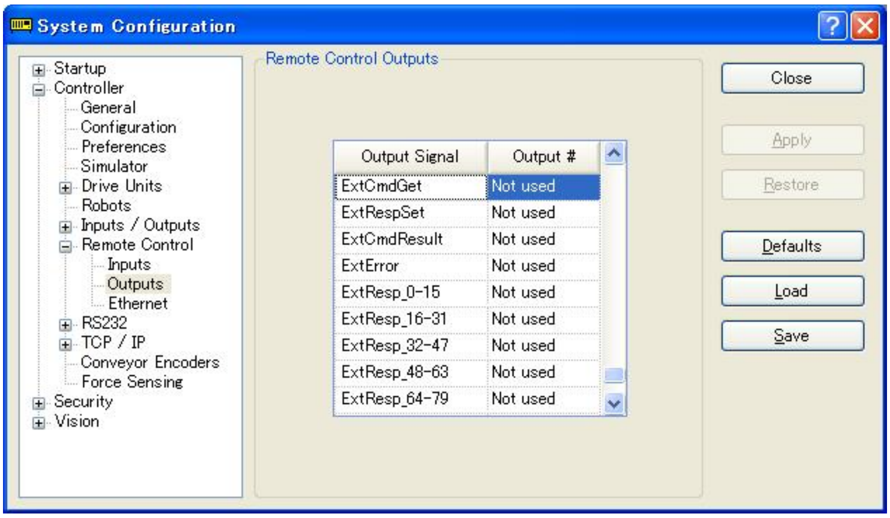
CAUTION	■ This function will not become effective unless all signals are set.



5.2.2. Setting the Output Signals

Set each signal output by the external equipment as output signals of the Remote I/O.

CAUTION	■ This function will not become effective unless all signals are set.



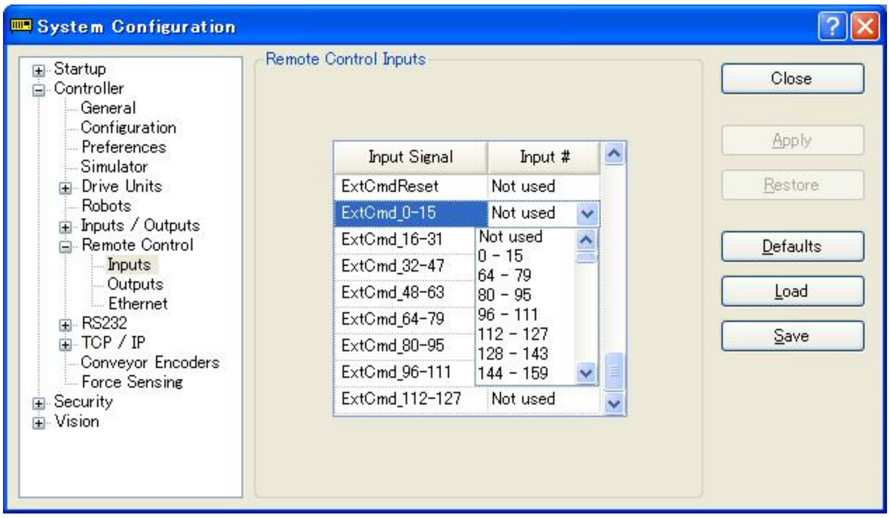
5.3 Setting the Data Signals

5.3.1 Setting the Command Signals

Set the command data signals in units of words. Check the size of the command and set it with the largest size.

Also, be careful of the following:

- For data signals, make sure to set the number of words you need successively from "ExtCmd0".



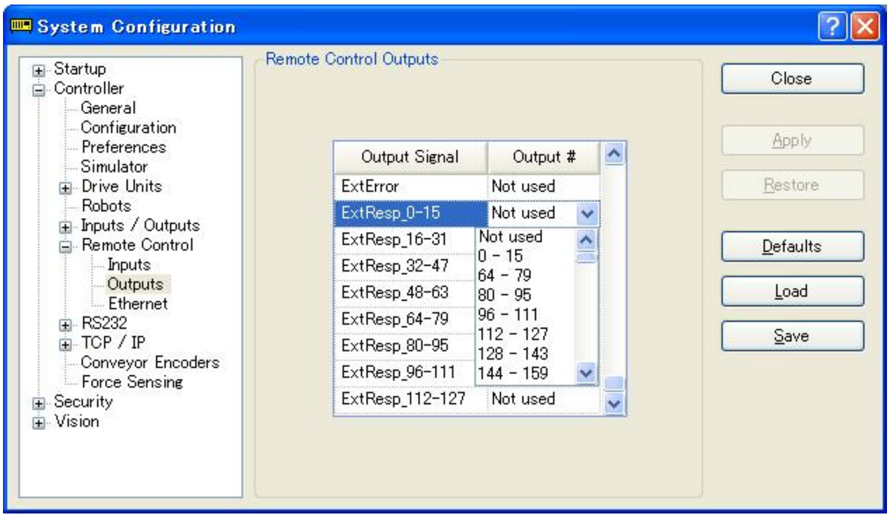
5.3.2 Setting the Response Signals

Set the outputs for acquiring the response data from the Controller in units of words.

Check the size of the command response and set it with the largest size.

Also, be careful of the following:

- For data signals, make sure to set the number of words you need successivelyfrom "ExtCmd0".
- Be sure to set 3 or more words for the response signal.



6. Control Method

6.1 Initial External Equipment Output Signal States

When connecting to the Controller or resetting this function, set the outputs from the external equipment as shown below.

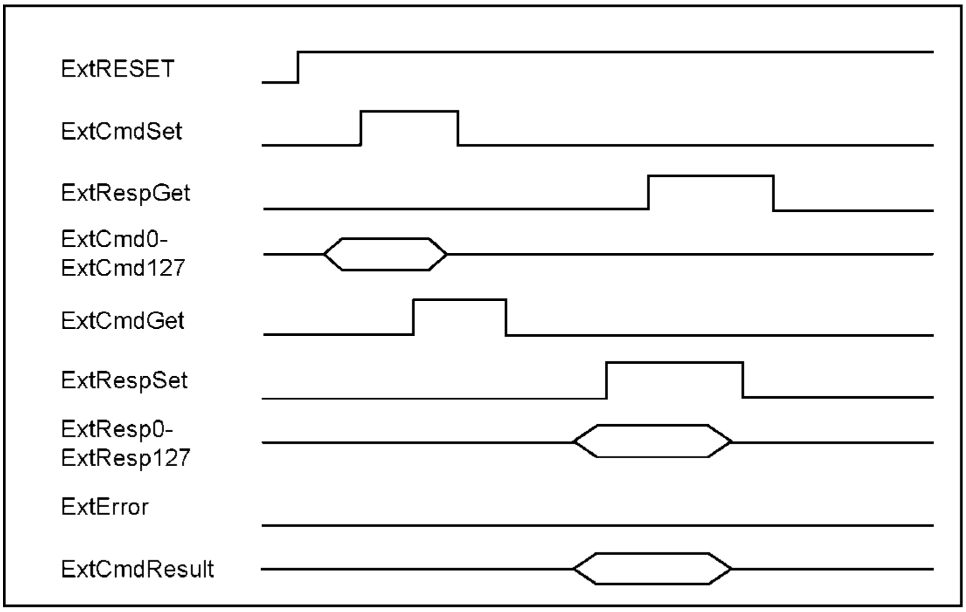
Name	Label	Output
Command set	ExtCmdSet	Low
Response acquisition	ExtRespGet	Low
Reset	ExtRESET	Low

6.2 Starting a Function

- Before a function can be started, the reset signal (ExtRESET) must be High.
- A command request can be received when the command set input (ExtCmdSet) is changed from Low to High while the reset input (ExtRESET) is set to High.
- When the command set input (ExtCmdSet) is set to High while the reset input is in Low state, the request will be ignored.
- When the reset input (ExtRESET) is changed to High while the command set input (ExtCmdSet) is set to High, the Controller cannot recognize the signal as a command request.

CAUTION	■ Release of the reset input should be executed after the Controller becomes operable. Also, initialize each input when the Controller is reset.

6.3 Command Execution



This section describes the command execution sequence for one command.

(1) The ExtRESET input is set to High to allow the command to be executed.

(2) The input data for the command to be executed is set in the command data area (ExtCmd0 - ExtCmd127).

(3) The command execution is requested by setting the ExtCmdSet input to High.

(4) The command acquisition of the Controller is confirmed when the command acquired output (ExtCmdGet) is set to High.

(5) After confirmation of the command acquisition, the command request input (ExtCmdSet) is set to Low.

(6) Command completion is indicated when the response set output (ExtRespSet) is set to High.

(7) The command result output (ExtCmdResult) indicates the command execution result.

(8) After the command result is checked, the response acquisition input (ExtRespGet) is set to High.

(9) The response set output (ExtRespSet) is set to Low.

(10) The response acquisition input (ExtRespGet) is set to Low.

6.4 Response Acquisition

This section describes the response acquisition procedure.
A command response is one of two types: Normal response and Error response.

Normal response : This indicates that the proper command was requested and execution was also completed normally. For settings commands and control commands, command number and normal response codes are returned in the response data outputs. For acquisition commands, acquired data is returned in the response data outputs.

Error response : This indicates that the requested command or the execution result was not correct. For response data outputs, command number and response codes (error codes) are returned.

The external equipment acquires whether the requested command is either one of the above response by the command result output (ExtCmdResult).
Be sure to check this output when acquiring the response data signal.

Low : The result is normal. Execute the acquisition process for the requested command.

High : The result is abnormal. Check the abnormality from the response codes and deal with the error as necessary.

Set the response acquisition input (ExtRespGet) after the acquisition of the response result and the response data is completed. If the response acquisition input is set before the acquisition completion, the Controller may rewrite the information.

6.5 Malfunction

A malfunction is a situation where the robot control using this function can not continue. Controller aborts the command execution if there is a running command. Also, a response for the executing command can not be returned. The function remains in halt state and commands can not be accepted until “High” is output to the malfunction signal (ExtError), the error code is set to the response data signal, and the function or the controller is reset.

6.5.1 Malfunction Factors

Malfunction occurs due to two main factors:

Controller factor : Controller needs to be reset

External equipment factor : Operation can be resumed after a function reset.
This occurs when a new command execution request is sent while another commad is still executing. Command processing of this function is under the premise that one command is complete by a set of request and response. If a new command request is executed while the other command is being executed, phases of the external equipment and the Controller do not match. In this case, stop the operation for safety.

6.5.2 How to Reset a Malfunction

This section describes how to reset a malfunction condition.

(1) Set the function reset input (ExtRESET) to the reset state (Low).

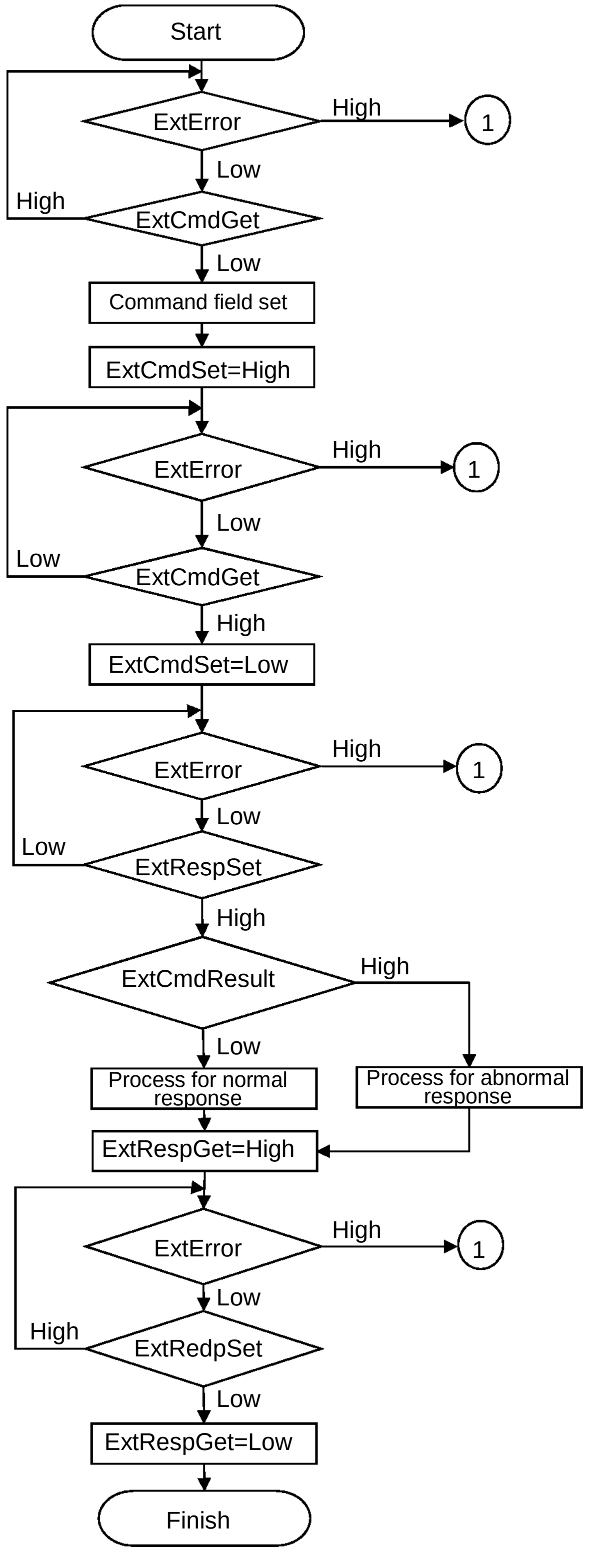
(2) Set the command set input (ExtCmdSet) to the release state (Low).

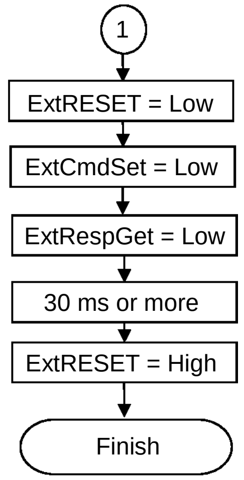
(3) Set the response acquisition input (ExtRespGet) to the release state (Low).

(4) Wait at least 30 ms.

(5) Set the function reset input (ExtRESET) to the release state (High).

Now, the malfunction reset is completed and a new commad can be requested.


Start
ExtError
High
1
Low
High
ExtCmdGet
Low
Command field set
ExtCmdSet=High
ExtError
High
1
Low
Low
ExtCmdGet
High
ExtCmdSet=Low
ExtError
High
1
Low
Low
ExtRespSet
High
ExtCmdResult
High
Low
Process for normal response
Process for abnormal response
ExtRespGet=High
ExtError
High
1
Low
High
ExtRedpSet
Low
ExtRespGet=Low
Finish


1
ExtRESET = Low
ExtCmdSet = Low
ExtRespGet = Low
30 ms or more
ExtRESET = High
Finish

7. Response Codes

Normal responses for commands other than data acquisition commands and error response are in the following format:

Command number	Response 1	Response 2

Note that some error codes do not have Response 2. In such cases, “0000H” will be returned.

7.1 Response 1 Codes

This section outlines the codes for response 1.

Outline	Description	Remedial measure	Code (HEX)
Normal	Command is completed normally.	−	0000
Command number error	Unsupported command number is requested.	*3	1000
Command sequence error	The order of the command requests is not proper.	*3	1002
Execution error	Requested command cannot be executed.	*3	2000
Command/Response Word number setting error	Word settings for both command and response necessary to execute the requested command are not proper.	*3	2001
Command word number setting error	Word setting for command necessary to execute the requested command is not proper.	*3	2002
Response word number setting error	Word setting for response necessary to execute the requested command is not proper.	*3	2003
Parameter error	Command parameter is not correct.	*3	2004
Table number specification error	Table number specification, such as speed table, exceeds the range.	*3	2005
Table registration error	Motion command option and the table specified for status acquisition are not registered.	*3	2006
Pallet undifined	Pallet specified by the pallet acquisition command is not registered.	*3	2007
Pallet point number discrepancy	Point number of registred pallet does not match that of the specified pallet.	*3	2008
Box undifined	Box specified by the Box acquisition command is not registered.	*3	2009
Command execution error	Error occurred as a result of command execution.	*3	200A
Command not accepted	Command execution cannot be accepted due to the system status.	*3	200B
RC error	Controller error occured.	*1	3000
Function error	The function has abnormality.	*2	9999

*1: Check the error code in Response 2, and refer to the Controller manual.

*2: Reset the function using the function reset signal.

*3: Next command can be accepted without any change. Command in issue is not completed.
Review the control method. (except 200B.)

7.2 Response 2 Codes

This section describes the details of the codes for response 2.
If there is no description in this section, “0000H” will be set.

7.2.1 Command Execution Error (Response 1: 200A)

Refer to the error code list in:

Robot Controller manual, or
EPSON RC+ 7.0 SPEL+ Language Reference.

7.2.2 RC Error (Response 1: 3000)

Refer to the error code list in:

Robot Controller manual.

7.2.3 Function Error (Response 1: 9999)

Codes 9901 and 9902 are errors caused by the Controller.

Outline	Description	Code (HEX)
Command acceptance status error	Command request is sent while the other command is being executed.	0001
WBMPostMessageExtra failure	Message notification failed.	9901
PRINT message error	Content of the PRINT message is not proper.	9902

8. Command List

8.1 Setting Commands

Group	Command number	Description	Number of Words	
			Command	Response
Accel/Decel setting of PTP motion (Accel)	0	Sets acceleration and deceleration for PTP motion	3	3
	1	Registers acceleration and deceleration to the accel/decel table	4	3
	2	Acquires the currently set acceleration and deceleration values	1	3
	3	Acquires acceleration and deceleration values from the accel/decel table for PTP motion	2	4
Accel/ Decel settings for Linear and CP motion (AccelS)	50	Sets acceleration and deceleration	5	3
	51	Sets acceleration setting value	3	3
	52	Informs the Controller of deceleration setting value and sets acceleration and deceleration	3	3
	53	Registers acceleration and deceleration to the accel/decel table	6	3
	54	Registers acceleration setting value to the accel/decel table	4	3
	55	Registers deceleration setting value to the accel/decel table	4	3
	56	Acquires current acceleration and deceleration setting values	1	5
	57	Acquires current acceleration value	1	3
	58	Acquires current deceleration value	1	3
	59	Acquires the registered values from the accel/ decel table	2	6
	60	Acquires the registered acceleration value from the accel/decel table	2	4
	61	Acquires the registered deceleration value from the accel/decel table	2	4
Accel/decel setting of Tool orientation change in CP motion (AccelR)	100	Sets the acceleration and deceleration	5	3
	101	Sets the acceleration value of the acceleration and deceleration	3	3
	102	Sets the deceleration value of the acceleration and deceleration	3	3
	103	Registers to the accel/decel table	6	3
	104	Registers acceleration value to the accel/decel table	4	3
	105	Registers deceleration value to the accel/decel table	4	3
	106	Acquires the currently set accel/decel values	1	5
	107	Acquires current acceleration value	1	3
	108	Acquires current deceleration value	1	3
	109	Acquires the registered value from the accel/decel table	2	6
	110	Acquires the registered acceleration value from the accel/decel table	2	4
	111	Acquires the registered deceleration value from the accel / decel table	2	4

Group	Command number	Description	Number of Words	
			Command	Response
Speed setting of PTP motion (Speed)	150	Sets the speed	4	3
	151	Registers to the speed table	5	3
	152	Acquires current speed setting value	1	4
	153	Acquires the setting value from the speed table	2	5
Arm speed setting of CP motion (SpeedS)	200	Sets the speed, depart speed, and approach speed settings	7	3
	201	Sets the setting values of speed and depart speed to the Controller.	5	3
	202	Notifies and sets the approach speed setting value	3	3
	203	Registers to the speed table	8	3
	204	Registers speed and depart speed to the speed table	6	3
	205	Registers approach speed to the speed table	4	3
	206	Acquires current speed, depart speed, and approach speed	1	7
	207	Acquires the setting values of current speed and depart speed	1	5
	208	Acquires the set value of current approach speed	1	3
	209	Acquires the setting value from the speed table	2	8
	210	Registers speed and depart speed to the speed table	2	6
	211	Acquires the approach speed from the speed table	2	4
Speed setting of Tool orientation change in CP motion when using ROT (SpeedR)	250	Sets the speed	3	3
	251	Registers to the speed table	4	3
	252	Acquires current speed setting value	1	3
	253	Acquires the setting value from the speed table	2	4
Parameter setting to offset speed and accel/decel in PTP motion (Weight)	300	Sets the parameter for offsetting the speed and accel/decel in PTP motion with an arm length specified	5	3
	301	Sets the parameter for offsetting the speed and accel/decel in PTP motion without an arm length specified	3	3
	302	Acquires the parameter setting value for offsetting the speed and accel/decel in PTP motion	1	5
Load inertia and eccentricity setting (Inertia)	350	Sets the load inertia and eccentricity	3	3
	351	Sets the load inertia	3	3
	352	Sets the eccentricity	3	3
	353	Acquires the setting values of load inertia and eccentricity	1	5
	354	Acquires the setting value of the load inertia	1	3
	355	Acquires the setting value of the eccentricity	1	3

Group	Command number	Description	Number of Words	
			Command	Response
Arch parameter setting (Arch)	400	Sets the arch parameter	6	3
	401	Sets the depart distance of the arch parameter	4	3
	402	Sets the approach distance of the arch parameter	4	3
	403	Acquires the arch parameter	2	6
	404	Acquires the depart distance setting value	2	4
	405	Acquires the approach distance setting value	2	4
Setting of positioning end judgement range (Fine)	450	Executes the settings of all joints	7	3
	451	Sets the setting values of Joint #1, #2, and #3	4	3
	452	Notifies and sets the setting values of Joint #4, #5, and #6	4	3
	453	Acquires the setting values of all joints	1	7
	454	Acquires the setting values of Joint #1, #2, and #3	1	4
	455	Acquires the setting values of Joint #4, #5, and #6	1	4
Tool selection (Tool)	500	Selects the tool	2	3
	501	Acquires the tool selection status	1	3
Pallet definition (Pallet)	550	Defines the pallet by specifying 4 points	8	3
	551	Defines the pallet by specifying 3 points	7	3
	552	Limits the numbers of points and divisions to define the pallet	4	3
	553	Selects the data type and defines the pallet by split	5	3
	554	Acquires the content of 4-point pallet definition	2	8
	555	Acquires the content of 3-point pallet definition	2	7
	556	Limits the number of points and division and acquires the pallet definition	2	5
	557	Selects the data type and acquires the details of pallet definition	3	5
	558	Acquires the point number set to the specified pallet	2	3
Approach check area setting (Box)	600	Specifies the lower and upper positions to define the approach check area	7	3
	601	Sets the lower limit position	5	3
	602	Sets the upper limit position	5	3
	603	Specifies the lower and upper limit positions and acquires the setting values of the approach check area	3	7
	604	Specifies the lower limit position and acquires the setting value of the approach check area	3	5
	605	Specifies the upper limit position and acquires the setting value of the approach check area	3	5
Approach check plane setting (Plane)	650	Set the approach check plane	5	3
	651	Acquires the setting value of the approach check plane	3	5

Group	Command number	Description	Number of Words	
			Command	Response
Local coordinate definition (Local)	700	Sets the definition of Local coordinate system	5	3
	701	Acquires the definition of Local coordinate system	3	5
Allowable motion area setting (XYLim)	750	Set the allowable motion area by specifying the lower and upper limit positions	6	3
	751	Sets the lower limit position	4	3
	752	Sets the upper limit position	4	3
	753	Acquires the setting value of the allowable motion area by specifying the lower and upper limit positions	2	6
	754	Acquires the setting value of the allowable motion area by specifying the lower limit position	2	4
	755	Acquires the setting value of the allowable motion area by specifying the upper limit position	2	4
Pulse value setting for the allowable motion area of the specified joint (Jrange)	800	Sets the allowable motion area pulse value by specifying the upper and lower limit pulses	6	3
	801	Sets the lower limit pulse value	4	3
	802	Sets the upper limit pulse value	4	3
	803	Acquires the allowable motion area pulse setting value by specifying the lower and upper limit pulses	2	6
	804	Acquires the allowable motion area pulse setting value by specifying the lower limit pulse	2	4
	805	Acquires the allowable motion area pulse setting value by specifying the upper limit pulse	2	4
Base coordinate definition (Base)	850	Defines the Base coordinate system	4	3
	851	Acquires the Base coordinate definition	2	4
Local number setting	900	Sets the Local coordinate number	2	3
	901	Acquires the setting status of the Local coordinate system number	1	3
Sense condition setting (Sense)	950	Sets the condition for using Sense with command 2002 and 2003	3	3
	951	Acquires the condition for using Sense with command 2002 and 2003	1	4
	952	Acquires the status of condition satisfaction	1	3
Find condition setting (Find)	1000	Sets the condition for using Find with command 2001, 2002, and 2003	3	3
	1001	Acquires the condition for using Find with command 2001, 2002, and 2003	1	4
	1002	Acquires the status of condition satisfaction	1	2
Till condition setting (Till)	1050	Sets the condition for using Till with motion commands	3	3
	1051	Acquires the condition for using Till with motion commands	1	4
	1052	Acquires the status of condition satisfaction	1	3
CP control (CP)	1100	Control the CP	2	3
	1101	Acquires the CP control state	1	3

Group	Command number	Description	Number of Words	
			Command	Response
Power control (Power)	1150	Controls the Power	2	3
	1151	Acquires the Power control state	1	3
Point editing	1200	Sets the current manipulator position to the specified point	2	3
	1201	Adjusts two coordinates to the specified point	7	3
	1202	Adjusts the specified coordinate to the specified point	5	3
	1203	Sets two coordinates to the specified point	7	3
	1204	Sets the specified coordinate to the specified point	5	3
	1205	Adjusts the coordinate to the specified point	3	3
	1206	Sets the hand orientation of the specified point to Righty	2	3
	1207	Sets the hand orientation of the specified point to Lefty	2	3
	1208	Sets the elbow orientation of the specified point to ABOVE	2	3
	1209	Sets the elbow orientation of the specified point to BELOW	2	3
	1210	Sets the wrist orientation of the specified point to FLIP	2	3
	1211	Sets the wrist orientation of the specified point to NOFLIP	2	3
	1212	Sets the j4flag value of the specified point	3	3
	1213	Sets the j6flag value of the specified point	3	3
	1214	Sets the Local number to the specified point	3	3
	1215	Acquires the hand orientation of the specified point	2	3
	1216	Acquires the elbow orientation of the specified point	2	3
	1217	Acquires the wrist orientation of the specified point	2	3
	1218	Acquires the j4flag value of the specified point	2	3
	1219	Acquires the j6flag value of the specified point	2	3
	1220	Acquires the Local number of the specified point	2	3
	1221	Sets the coordinate recorded by Find to the specified point	2	3
	1222	Acquires the coordinate of the specified point	3	3
	1223	Sets the J1flag	3	3
	1224	Acquires the status of J1flag	2	3
	1225	Sets the J2flag	3	3
	1226	Acquires the status of J2flag	2	3
	1227	Sets the J1angle attribute of the point	5	3
	1228	Acquire the J1angle attribute of the point	2	3

Group	Command number	Description	Number of Words	
			Command	Response
Initial Joint #3 height (Z coordinate value) in Jump command (Limz)	1250	Sets the initial Joint #3 height (Z coordinate value) in Jump command	3	3
	1251	Acquires the initial Joint #3 height (Z coordinate value) in Jump command	1	3
Parallel processing	1300	Registers the parallel processing list to be used in motion command execution	5	3
	1301	Acquires the setting state of the parallel processing list used in motion command execution	3	5
	1302	Initializes the specified list	2	3
	1303	Sets the parallel processing list to be used in the motion commands	2	3
	1304	Acquires the selective condition of the parallel processing list to be used in the motion commands	1	3
Singularity avoidance	1350	Specifies whether to use LJM automatically in order to avoid singularity (AutoLJM)	2	3
	1352	Sets the singularity avoiding function	2	3
Motor control	1400	Controls ON/OFF of the motor.	2	3
	1401	Acquires the status of the motor.	1	3
Reset	1450	Resets the controller to an initial status.	1	3

8.2 Motion Commands

<table>
<tr><th rowspan="2">Group</th><th rowspan="2">Command number</th><th rowspan="2">Description</th><th colspan="2">Number of Words</th></tr>
<tr><th>Command</th><th>Response</th></tr>
<tr><td rowspan="7">PTP motion from the current arm position to the specified position (Go)</td><td rowspan="7">2000</td><td colspan="3">Moves from the current position to the specified position in PTP motion</td></tr>
<tr><td>Destination specification= 0 Speed and Accel not specified</td><td>3</td><td rowspan="6">3</td></tr>
<tr><td>Destination specification= 1 Speed and Accel not specified</td><td>4</td></tr>
<tr><td>Destination specification= 2 Speed and Accel not specified</td><td>5</td></tr>
<tr><td>Destination specification= 0 Speed and Accel specified</td><td>4</td></tr>
<tr><td>Destination specification= 1 Speed and Accel specified</td><td>5</td></tr>
<tr><td>Destination specification= 2 Speed and Accel specified</td><td>6</td></tr>
<tr><td rowspan="7">Gate motion PTP motion (Jump)</td><td rowspan="7">2001</td><td colspan="3">Moves in PTP motion with gate motion</td></tr>
<tr><td>Destination specification= 0 Speed and Accel not specified</td><td>3</td><td rowspan="6">3</td></tr>
<tr><td>Destination specification= 1 Speed and Accel not specified</td><td>4</td></tr>
<tr><td>Destination specification= 2 Speed and Accel not specified</td><td>5</td></tr>
<tr><td>Destination specification= 0 Speed and Accel specified</td><td>4</td></tr>
<tr><td>Destination specification= 1 Speed and Accel specified</td><td>5</td></tr>
<tr><td>Destination specification= 2 Speed and Accel specified</td><td>6</td></tr>
<tr><td rowspan="3">3D gate motion (2 CP motion and 1 PTP motion) (Jump3)</td><td rowspan="3">2002</td><td colspan="3">Moves the arm with 3D gate motion
This is a combination of two CP motion and one PTP motion.</td></tr>
<tr><td>Speed and Accel not specified</td><td>5</td><td rowspan="2">3</td></tr>
<tr><td>Speed and Accel specified</td><td>6</td></tr>
<tr><td rowspan="3">3D gate motion 3 CP motion (Jump3CP)</td><td rowspan="3">2003</td><td colspan="3">Moves the arm with 3D gate motion
This is a combination of three CP motion.</td></tr>
<tr><td>Speed and Accel not specified</td><td>5</td><td rowspan="2">3</td></tr>
<tr><td>Speed and Accel specified</td><td>6</td></tr>
</table>

<table>
<tr><th rowspan="2">Group</th><th rowspan="2">Command number</th><th rowspan="2">Description</th><th colspan="2">Number of Words</th></tr>
<tr><th>Command</th><th>Response</th></tr>
<tr><td rowspan="7">Linear interpolation motion (Move)</td><td rowspan="7">2005</td><td colspan="3">Moves the arm from the current position to the specified position in a linear interpolation motion</td></tr>
<tr><td>Destination specification= 0 Speed and Accel not specified</td><td>3</td><td rowspan="6">3</td></tr>
<tr><td>Destination specification= 1 Speed and Accel not specified</td><td>4</td></tr>
<tr><td>Destination specification= 2 Speed and Accel not specified</td><td>5</td></tr>
<tr><td>Destination specification= 0 Speed and Accel specified</td><td>4</td></tr>
<tr><td>Destination specification= 1 Speed and Accel specified</td><td>5</td></tr>
<tr><td>Destination specification= 2 Speed and Accel specified</td><td>6</td></tr>
<tr><td rowspan="6">Arc interpolation motion (Arc) (Arc3)</td><td rowspan="3">2006</td><td colspan="3">Moves the arm from the current position to the specified position in Arc interpolation motion on XY plane face</td></tr>
<tr><td>Speed and Accel not specified</td><td>4</td><td rowspan="2">3</td></tr>
<tr><td>Speed and Accel specified</td><td>5</td></tr>
<tr><td rowspan="3">2007</td><td colspan="3">Moves the arm from the current position to the specified position in Arc interpolation motion in 3D</td></tr>
<tr><td>Speed and Accel not specified</td><td>4</td><td rowspan="2">3</td></tr>
<tr><td>Speed and Accel specified</td><td>5</td></tr>
</table>

8.3 Jog & Teach Commands

Group	Command number	Description	Number of Words	
			Command	Response
JOG & Teach	2050	Jog motion	5	3
	2051	Teach the current position to the specified point	3	3
	2052	Save the current point setting to the point file	2	3
	2053	Controls the temporary halt and resume of the motor excitation	3	3
	2054	Acquires the motor excitation status	1	3

8.4 Input / Output Commands

Group	Command number	Description	Number of Words	
			Command	Response
I/O control	2100	Acquires the status of the specified input port in bytes	2	3
	2101	Acquires the status of the specified input port in words	2	3
	2102	Outputs the byte data to the specified byte output port	3	3
	2103	Outputs the word data to the specified word output port	3	3
	2104	Acquires the bit status of the specified input bit port	2	3
	2105	Turns ON the output of the specified bit port	2	3
	2106	Turns OFF the output of the specified bit port	2	3
Memory I/O control	2107	Acquires the status of the specified memory I/O port in bytes	2	3
	2108	Acquires the status of the specified memory I/O port in words	2	3
	2109	Sets the specified memory I/O port in bytes	3	3
	2110	Sets the specified memory I/O port in words	3	3
	2111	Acquires the status of the specified memory I/O bit	2	3
	2112	Turns OFF the specified bit of the memory I/O	2	3
	2113	Turns ON the specified bit of the memory I/O	2	3

8.5 Reference Commands

Group	Command number	Description	Number of Words	
			Command	Response
Current position information	2150	Acquires the current position of the manipulator	2	5
Acquisition of the distance between 2 manipulator coordinates	2151	Acquires the distance between 2 manipulator coordinates	3	3
PTP move check	2152	Acquires whether the PTP (point to point) motion from the current position to the target position is possible	2	3
Manipulator type acquisition	2153	Acquires the manipulator type	1	3
Manipulator model name acquisition	2154	Acquires the manipulator model name	2	3
Controller error	2155	Acquires the Controller error information	1	3

9. Basic Command Usage

This section describes the basic usage of commands.

This function has the following types of commands: (Refer to *8. Command List*)

Setting Commands

Motion Commands

Jog & Teach Commands

Iuput and Output Commands

Reference Commands

Some commands execute the same operation and have different resource sizes. Also, some commands execute multiple functions.

You can select the commands and build a robot control system suitable for system configuration.

9.1 Using Speed and Acceleration Tables

You can set the speed and acceleration at one time by registering the parameters beforehand in a table and specifying a table index at the time of motion command execution. This can save the number of commands to be issued and make the motion faster.

The following items can be set:

Acceleration and deceleration settings of PTP motion

Acceleration and deceleration settings of Linear and CP motion

Acceleration and deceleration settings for Tool orientation change in CP motion

Speed setting of PTP motion

Arm speed setting in CP motion

Tool orientation change speed in CP motion when using ROT

Each item has a table structure. You can register 16 pattern settings. To set parameters, specify positions of the settings in the table using the available options of the motion command and operate the Manipulator.

CAUTION	■ Registered data in tables will be cleared when the Controller is turned OFF or reset.

9.2 Command Execution Procedure

The command execution procedure has several patterns. For details on execution procesedure, refer to the descriptions of each command.

(1) Functions by single command issue.

(2) Functions by issuing the same command several times.

(3) Functions by issuing several commands.

(4) Functions by issuing the same command several times and issuing the other commands.

9.3 Parallel Processing Lists

Some motion commands can control ON/OFF of the specified I/O in parallel with motion based on the specified progress rate. To enable this function, register the parameter lists for parallel processing beforehand, and set the list numbers to be used. There are 16 lists available, and one of them is used to register the processes against progress of 16 lists.

Registration and selection of the lists can be done with commands from No. 1300 to 1304.

CAUTION	■ Registered data in lists will be cleared when the Controller is turned OFF or reset.

10. Command Reference

10.1 Acceleration and Deceleration Settings of PTP Motion

These commands are used to set acceleration and deceleration of all PTP motion. Available acceleration/deceleration parameter is an integer equal to or greater than 1. This value indicates the ratio of acceleration to the maximum acceleration (or deceleration).

CAUTION	■ Setteings will be initialized in the following cases: Controller's power is turned ON Motor ON is executed Excitation control is executed Reset is executed Halt button or Ctrl+C are pressed ■ When executing the setting commands in Low Power mode (Power Low) In Low Power mode (Power Low), new values will be saved while the current values will be restrained at low.

Command 0: Set PTP Accel, Decel

Sets acceleration and deceleration for PTP motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*accel*	Specifies the ratio (%) of the maximum acceleration using an integer equals to or greater than 1.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*decel*	Specifies the ratio (%) to the maximum deceleration using an integer equals to or greater than 1.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code*.

Description

Acceleration and deceleration for PTP motion are set by issuing this command.

Example

Set "100" for acceleration and "80" for deceleration.

Command	Response
0000H 0064H 0050H	0000H 0000H 0000H

Command 1: Set PTP Accel, Decel In Table

Sets the acceleration and deceleration values in the acceleration/deceleration table for PTP motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel*	Specifies the ratio (%) to the maximum acceleration using an integer equals to or greater than 1.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*decel*	Specifies the ratio (%) to the maximum deceleration using an integer equals to or greater than 1.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets PTP motion acceleration and deceleration in a specified table. The execution of this command does not affect actual settings. The settings are reflected when they are specified as options at execution of the target motion command.

This command is used in combination with the motion commands.

Example

Sets “100” for acceleration and “80” for deceleration to the table 5.

Command	Response
0001H 0005H 0064H 0050H	0000H 0000H 0000H *1
07D0H 0100H 0000H 0005H	0000H 0000H 0000H *2

*1: Registeration to the table

*2: Specifies the table number and executes PTP motion

Command 2: Get PTP Accel, Decel

Acquires the current PTP motion acceleration and deceleration values.

Command Syntax

No parameters.

Response Syntax

	bit	Name	Description
Response 1	15	*accel*	Returns the current value as an integer that is equal to or greater than 1.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*decel*	Returns the current value as an integer that is equal to or greater than 1.
	14		
	|		
	1		
	0		

Description

Acquires the current PTP motion acceleration and deceleration.

Example

When acceleration is set to “100” and deceleration is “80”.

Command	Response
0002H	0002H 0064H 0050H

Command 3: Get PTP Accel, Decel From Table

Acquires the acceleration and deceleration values from the acceleration/deceleration table for PTP motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	｜		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel*	Returns the current value as an integer equal to or greater than 1.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*decel*	Returns the current value as an integer equal to or greater than 1.
	14		
	｜		
	1		
	0		

Description

Acquires acceleration and deceleration values from the acceleration/deceleration table for PTP motion. If the specified table number is out of range or not registered, an error response will be returned.

Example

When acceleration is “100” and deceleration is “80” in the table 1.

Command	Response
0003H 0001H	0003H 0001H 0064H 0050H

10.2 Acceleration and Deceleration Settings for Linear and CP Motion

These commands are used to specify acceleration and deceleration for linear and circular interpolation motion. This includes the linear motion and circular interpolation motion from the current arm position to the specified position in an X-Y plane.

CAUTION	■ Values will be initialized in following cases: Controller’s power is turned ON Motor ON is executed When excitation control is executed Reset is executed Halt button or Ctrl+C are pressed ■ When executing the setting commands in Low Power mode (Power Low) In Low Power mode (Power Low), new values will be saved while the current values will be restrained at low.

Command 50: Set Linear Accel, Decel

Sets linear motion acceleration and deceleration.

Command Syntax

Parameter 1	bit	Name	Description
	15	*accel* High-order word	Specifies the value which is the actual acceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*accel* Low-order word	Specifies the value which is the actual acceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*decel* High-order word	Specifies the value which is the actual deceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*decel* Low-order word	Specifies the value which is the actual deceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer.Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. *Response code.*

Description

Acceleration and deceleration are set by issuing this command. Setting values should be specified as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.0002.

Command	Response
0032H 0001H 871BH 0003H 0D40H	0032H 0000H 0000H

Command 51: Set Linear Accel

Informs the Controller of acceleration setting value.

Command Syntax

Parameter 1	bit	Name	Description
	15	*accel* High-order word	Specifies the value which is the actual acceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*accel* Low-order word	Specifies the value which is the actual acceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. *Response code.*

Description

This command sets acceleration value when setting acceleration and deceleration separately.
This command does not function by itself, but functions in combination with the Command 52.
Setting will be executed by issuing the Command 52 after this command. Acceleration value will be canceled if commands other than the Command 52 are issued.

Setting values should be specified as fixed-point data which validates to three decimal places.

Example

When acceleration is set to "100.123".

Command	Response
0033H 0001H 871BH	0033H 0000H 0000H

Command 52: Set Linear Decel

Informs the Controller of deceleration setting value and sets acceleration and deceleration.

Command Syntax

	bit	Name	Description
Parameter 1	15	*decel* High-order word	Specifies the value which is the actual deceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*decel* Low-order word	Specifies the value which is the actual deceleration in linear or CP motion (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command sets deceleration value when setting acceleration and deceleration separately.
This command does not function by itself. Settings of acceleration and deceleration will be executed when acceleration is set by the Command 51 right before this command. If the previous command is not the Command 51, an error response will be returned.

Setting values should be specified as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
0033H 0001H 871BH	0033H 0000H 0000H *1
0034H 0003H 0D40H	0034H 0000H 0000H *2

*1 Notifies the acceleration value by the command No. 51.

*2 Notifies the deceleration value by the command No.52. Command sequence is completed and settings of acceleration and deceleration will be executed.

Command 53: Set Linear Accel, Decel In Table

Registers acceleration and deceleration for linear and CP motion to the acceleration/deceleration table.

Command Syntax

	Bit	Name	Description
Parameter 1	15 14 | 1 0	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.

	bit	Name	Description
Parameter 2	15 14 | 1 0	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 | 1 0	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 | 1 0	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 5	15 14 | 1 0	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to 7. *Response code.*

Description

Registers acceleration and deceleration to specified table numbers. The issue of this command does not affect actual settings. The settings will be reflected when they are specified as options at execution of the target motion command.
This command is used in combination with the motion commands.

Setting values should be specified as fixed-point data which validates to three decimal places.

Example

When registering "100.123" for acceleration and "200.000" for deceleration in the table 15.

Command	Response
0035H 000FH 0001H 871BH 0003H 0D40H	0035H 0000H 0000H

Command 54: Set Linear Accel In Table

Sets acceleration for linear and CP motion to the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. *Response code.*

Description

This command registers acceleration to the acceleration/deceleration table when registering acceleration and deceleration separately.
This command does not function by it self. This is used in combination with the Command 55.
Setting will be executed by issuing the Command 55 after this command.
Separate registration will be canceled if commands other than the Command 55 are issued.

The rest is same as the command No.53.

Example

When registering “100.123” for acceleration and “200.000” for deceleration in the table 15.

Command	Response
0036H 000FH 0001H 871BH	0036H 0000H 0000H

Command 55: Set Linear Decel In Table

Sets deceleration for linear and CP motion to the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.

	bit	Name	Description
Parameter 2	15 14 │ 1 0	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 │ 1 0	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

This command registers deceleration to the acceleration/deceleration table when registering acceleration and deceleration separately.
This command does not function by it self. This is used in combination with command No. 54.
If the previous command is the command No.54, registeration to the acceleration/deceleration table in combination with the previously specified acceleration will be completed.
If the previous command is not the command No. 54, an error response will be returned.

The rest is same as the command No.53.

Example

When registering “100.123” for acceleration and “200.000” for deceleration in the table 15.

Command	Response
0036H 000FH 0001H 871BH	0036H 0000H 0000H
0037H 000FH 0003H 0D40H	0037H 0000H 0000H

Command 56: Get Linear Accel, Decel

Reads the current acceleration and deceleration settings for linear and CP motion.

Command Syntax

No parameters.

Response Syntax

	bit	Name	Description
Response 1	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 3	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^{2}2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 4	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

Description

Acquires current acceleration and deceleration settings.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
0038H	0038H 0001H 871BH 0003H 0D40H

Command 57: Get Linear Accel

Acquires current acceleration and deceleration settings for linear and CP motion separately.
This command acquires the acceleration value.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires current acceleration setting value.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
0039H	0039H 0001H 871BH

Command 58: Get Linear Decel

Acquires current acceleration and deceleration settings for linear and CP motion separately.
This command acquires deceleration value.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 2	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Description

Acquires current deceleration setting value.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to 100.123 and deceleration is 200.000:

Command	Response
003AH	003AH 0003H 0D40H

Command 59: Get Linear Accel, Decel From Table

Acquires current acceleration and deceleration settings for linear and CP motion separately.
Acquires both acceleration and deceleration.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	⏐		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	⏐		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	⏐		
	1		
	0		

	bit	Name	Description
Response 3	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	⏐		
	1		
	0		

	bit	Name	Description
Response 4	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	⏐		
	1		
	0		

	bit	Name	Description
Response 5	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires acceleration and deceleration values from the acceleration/deceleration table for linear and CP motion.
Acceleration and deceleration can be acquired at one time by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When “100.123” is registered for acceleration and “200.000” is registered for deceleration in the table 15.

Command	Response
003BH 000FH	003BH 000FH 0001H 871BH 0003H 0D40H

Command 60: Get Linear Accel From Table

Reads the acceleration value for linear and CP motion from the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires setting values from the acceleration/deceleration table for linear and CP motion. Acceleration value can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When “100.123” is registered for acceleration and “200.000” is registered for deceleration in the table 15.

Command	Response
003CH 000FH	003CH 000FH 0001H 871BH

Command 61: Get Linear Decel From Table

Reads the deceleration value for linear and CP motion from the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: mm/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires setting values from the acceleration/deceleration table for linear and CP motion. Deceleration can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When “100.123” is registered for acceleration and “200.000” is registered for deceleration in the table 15.

Command	Response
003CH 000FH	003DH 000FH 0003H 0D40H

10.3 Acceleration and Deceleration Settings for Tool Orientation Change in CP Motion

The following commands are used to set and display acceleration and deceleration settings for Tool orientation change in CP motion.

The commands are enabled when ROT option is used in commands No. 2003 (Jump3CP), 2005 (Move), 2006 (Arc), and 2007 (Arc3).

⚠ CAUTION	■ Setteing will be initialized in following cases: Controller's power is turned ON Motor ON is executed When excitation control is executed Reset is executed Halt button or Ctrl+C are pressed

Command 100: Set Accel, Decel For Tool Orientation

Sets acceleration and deceleration setting for Tool orientation change in CP motion.

Command Syntax

Parameter 1	bit	Name	Description
	15	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 3	Bit	Name	Description
	15	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 3	Bit	Name	Description
	15	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets acceleration and deceleration for Tool orientation change in CP motion.
Values should be specified as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
0064H 0001H 871BH 0003H 0D40H	0064H 0000H 0000H

Command 101: Set Accel For Tool Orientation

Sets acceleration for Tool orientation in CP motion separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command sets the acceleration value when setting acceleration and deceleration separately. This command is used in combination with Command 102.

To set acceleration and deceleration, execute Commands 101 and 102, in that order.
The actual values will be set when Command 102 is executed.
If commands other than Command 102 are issued after this command, setting will be canceled.

Example

When acceleration is set to "100.123" and deceleration is "200.000".

Command	Response
0065H 0001H 871BH	0065H 0000H 0000H
0066H 0003H 0D40H	0066H 0000H 0000H

Command 102: Set Decel For Tool Orientation

Sets deceleration for Tool orientation in CP motion separately.

Command Syntax

Parameter 1	bit	Name	Description
	15	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command sets the deceleration value when setting acceleration and deceleration separately.
This command is used in combination with Command 101.
To set acceleration and deceleration, execute Command 101 and 102, in that order.
The actual values will be set when Command 102 is executed.
If the previous command is not Command 101, an error response will be returned.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
0065H 0001H 871BH	0065H 0000H 0000H
0066H 0003H 0D40H	0066H 0000H 0000H

Command 103: Set Accel, Decel For Tool Orientation In Table

Sets the acceleration and deceleration settings for Tool orientation change in CP motion in the acceleration/deceleration table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

Response Syntax

Refer to 7. Response code.

Description

Registers both acceleration and deceleration to the specified table.

The issue of this command does not affect actual settings. The settings will be reflected when they are specified as options at execution of the target motion command.

This command is used in combination with the motion commands.

Example

When registering “100.123” for acceleration and “200.000” for deceleration in the table 1.

Command	Response
0067H 0001H 0001H 871BH 0003H 0D40H	0067H 0000H 0000H

Command 104: Set Accel For Tool Orientation In Table

Sets the acceleration for Tool orientation change in CP motion to the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel* High-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*accel* Low-order word	Specifies the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command registers acceleration to the acceleration/deceleration table when registering acceleration and deceleration separately.
This command does not function by itself. This command functions in combination with command No.105.
To set acceleration, execute the commands No. 104 and No.105, in that order.
Registration will be completed by executing the command No.105.

If commands other than No.105 are issued after this command, setting will be canceled.

Example

When registering “100.123” for acceleration and “200.000” for deceleration in the table 1.

Command	Response
0068H 0001H 0001H 871BH	0068H 0000H 0000H
0069H 0001H 0003H 0D40H	0069H 0000H 0000H

Command 105: Set Decel For Tool Orientation In Table

Sets deceleration for Tool orientation change in CP motion in the acceleration/deceleration table separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*decel* High-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) $\times$ 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*decel* Low-order word	Specifies the deceleration value which is the actual deceleration (Unit: deg/sec^2) $\times$ 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. *Response code.*

Description

This command registers deceleration to the acceleration/deceleration table when registering acceleration and deceleration separately.
This command is used in combination with Command 104.
To set deceleration, execute the Commands 104 and 105, in that order.
Registration will be completed by executing Command 105.

If the previous command is not Command 104, an error response will be returned.

Example

When registering "100.123" for acceleration and "200.000" for deceleration in the table 1.

Command	Response
0068H 0001H 0001H 871BH	0068H 0000H 0000H
0069H 0001H 0003H 0D40H	0069H 0000H 0000H

Command 106: Get Accel, Decel for Tool Orientation

Acquires current acceleration and deceleration values for Tool orientation change in CP motion.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires current acceleration and deceleration settings.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
006AH	006AH 0001H 871BH 0003H 0D40H

Command 107: Get Accel For Tool Orientation

Acquires current acceleration value for Tool orientation change in CP motion separately.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command acquires the acceleration value when acquiring acceleration and deceleration for Tool orientation change in CP motion separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
006BH	006BH 0001H 871BH

Command 108: Get Decel For Tool Orientation

Acquires current deceleration value for Tool orientation change in CP motion separately.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command acquires deceleration value when acquiring acceleration and deceleration for Tool orientation change in CP motion separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is set to “100.123” and deceleration is “200.000”.

Command	Response
006CH	006CH 0003H 0D40H

Command 109: Get Accel, Decel For Tool Orientation From Table

Acquires the values for Tool orientation change in CP motion from the acceleration/deceleration table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 5	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command acquires registered values for Tool orientation change in CP motion from the specified position in the acceleration/deceleration table.
Acceleration and deceleration values can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is “100.123” and deceleration is “200.000” in the table 1.

Command	Response
006DH 0001H	006DH 0001H 0001H 871BH 0003H 0D40H

Command 110: Get Accel For Tool Orientation From Table

Acquires acceleration for Tool orientation change in CP motion from the acceleration/deceleration table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	│		
	1		
	0		

Response Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*accel* High-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*accel* Low-order word	Returns the acceleration value which is the actual acceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Description

Acquires registered values for Tool orientation change in CP motion from the specified position in the acceleration/deceleration table.
Acceleration value can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is “100.123” and deceleration is “200.000” in the table 1.

Command	Response
006EH 0001H	006EH 0001H 0001H 871BH

Command 111: Get Decel For Tool Orientation From Table

Acquires deceleration for Tool orientation change in CP motion from the acceleration/deceleration table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*decel* High-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*decel* Low-order word	Returns the deceleration value which is the actual deceleration (Unit: deg/sec^2) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires registered values for Tool orientation change in CP motion from the specified position in the acceleration/deceleration table.
Deceleration value can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.
The value will be returned as fixed-point data which validates to three decimal places.

Example

When acceleration is “100.123” and deceleration is “200.000” in the table 1.

Command	Response
006FH 0001H	006FH 0001H 0003H 0D40H

10.4 Speed Setting of PTP Motion

These commands are used to specify the speed to all PTP motion commands. Speed setting should be specified by the percentage (%) of the maximum speed indicated by an integer from 1 to 100. The manipulator moves at the maximum speed when “100” is specified.

Depart speed and approach speed are only applicable for the gate-motion PTP motion (command No.2001, Jump).

CAUTION	■ Setteing will be initialized in following cases: Controller’s power is turned ON Motor ON is executed When excitation control is executed Reset is executed Halt button or Ctrl+C are pressed

Setting value becomes lower than the defalt value in Low Power mode. Even when the value greater than the default is input by commands, the default value will be set. Setting value is set as motion speed in High Power mode. If greater motion speed is required, set the mode to High Power by Power High command and close the safety door. The value will be changed to the defalt if the safety door is open.

Command 150: Set PTP Speed

Sets speed for PTP motion.

Command Syntax

Parameter 1	bit	Name	Description
	15	*speed*	Specifies the percentage (%) of the maximum speed indicated by an integer from 1 to 100.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*departSpeed*	Specifies the departing motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*approachSpeed*	Specifies the approaching motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets speed, depart speed, and approach speed for PTP motion.

Example

When setting speed as 100, depart speed as 80, and approach speed as 50.

Command	Response
0096H 0064H 0050H 0032H	0096H 0000H 0000H

Command 151: Set PTP Speed In Table

Registers a speed value in the speed table for PTP motion.

Command Syntax

Parameter 1	bit	Name	Description
	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*speed*	Specifies the percentage (%) of the maximum speed indicated by an integer from 1 to 100.
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*departSpeed*	Specifies the departing motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*approachSpeed*	Specifies the approaching motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. *Response code.*

Description

Registers speed, depart speed, and approach speed for PTP motion to the specified table.
If the specified table number is out of range, an error response will be returned.

Example

When registering speed as “100”, depart speed as “80”, and approach speed as “50” to the table 1.

Command	Response
0097H 0001H 0064H 0050H 0032H	0097H 0000H 0000H

Command 152: Get PTP Speed

Acquires the current speed setting values for PTP motion.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*speed*	Returns the percentage (%) of the maximum speed indicated by an integer from 1 to 100.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*departSpeed*	Returns the departing motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*approachSpeed*	Returns the approaching motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	|		
	1		
	0		

Description

Acquires current values of speed, depart speed, and approach speed.

Example

When speed is “100”, depart speed is “80”, and approach speed is “50”.

Command	Response
0098H	0098H 0064H 0050H 0032H

Command 153: Get PTP Speed From Table

Acquires speed values from the speed table for PTP motion.

Command Syntax

Parameter 1	bit	Name	Description
	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	│		
	1		
	0		

Response Syntax

Parameter 1	bit	Name	Description
	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	│		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*speed*	Returns the percentage (%) of the maximum speed indicated by an integer from 1 to 100.
	14		
	│		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*departSpeed*	Returns the departing motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	│		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*approachSpeed*	Returns the approaching motion speed in gate-motion PTP motion by an integer from 1 to 100. (Unit: %)
	14		
	│		
	1		
	0		

Description

Acquires speed values from the speed table for PTP motion.
If the specified table number is out of range or not registered, an error response will be returned.

Example

When speed is “100”, depart speed is “80”, and approach speed is “50” in the table 1.

Command	Response
0099H 0001H	0099H 0001H 0064H 0050H 0032H

10.5 Arm Speed Setting of CP Motion

These commands are used to set Arm speed setting in CP motion such as Move, Arc, Arc3, Jump3, and Jump3CP.

SpeedS specifies speed in CP motion (Move and Arc) execution.

SpeedS value specifies the manipulator speed. The unit is mm/sec. Default values vary according to the manipulator types. For default values of SpeedS, refer to each model's manipulator manual. The default values are automatically set when the Controller's power is turned ON.

CAUTION	■ Setteing will be initialized in following cases: Controller's power is turned ON Motor ON is executed When excitation control is executed Reset is executed Halt button or Ctrl+C are pressed

In Low Power mode, the lower speed between the default value and the setting value will be effective for SpeedS. If the greater speed setting is specified in the command window or during the program, default speed will be set.

SpeedS setting value will be set as motion speed in High Power mode. If greater motion speed is required, set the mode to High Power by Power High command and close the safety door. SpeedS value will be changed to the defalt if the safety door is open.

Command 200: Set CP Speeds

Sets Arm speed for CP motion.

Command Syntax

Parameter	bit	Name	Description
Parameter 1	15	*speed* High-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter	bit	Name	Description
Parameter 2	15	*speed* Low-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter	bit	Name	Description
Parameter 3	15	*departSpeed* High-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter	bit	Name	Description
Parameter 4	15	*departSpeed* Low-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter	bit	Name	Description
Parameter 5	15	*approachSpeed* High-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter	bit	Name	Description
Parameter 6	15	*approachSpeed* Low-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets Arm speed for CP motion.

Example

When setting speed as “100.001”, depart speed as “50.002”, and approach speed as “60.003”.

Command	Response
00C8H 0001H 86A1H 0000H C352H 0000H EA63H	00C8H 0000H 0000H

Command 201: Set CP Speed and Depart Speed

Sets Arm speeds for CP motion separately.
Sets speed and depart speed setting values.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*speed* High-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 2	15 14 │ 1 0	*speed* Low-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 │ 1 0	*departSpeed* High-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 │ 1 0	*departSpeed* Low-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

This command sets speed and depart speed when setting arm speeds for CP motion separately.
This command functions in combination with Command 202.

To set setting values, execute the Command 201 and 202, in that order.
The settings will be effective after issuing Command 202.
If commands other than Command 202 are issued after this command, setting will be canceled.

The rest is same as the Command 200.

Example

When setting speed as “100.001”, depart speed as “50.002”, and approach speed as “60.003”.

Command	Response
00C9H 0001H 86A1H 0000H C352H	00C9H 0000H 0000H
00CAH 0000H EA63H	00CAH 0000H 0000H

Command 202: Set CP Approach Speed

Sets Arm speeds for CP motion separately.
Sets the approach speed.

Command Syntax

	bit	Name	Description
Parameter 1	15	*approachSpeed* High-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*approachSpeed* Low-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command sets approach speed when setting arm speeds for CP motion separately.
This command functions in combination with Command 201.

To set the setting value, execute the Command 201 and 202, in that order.
The settings will be effective after issuing Command 202.
If the last command is not Command 201, an error response will be returned.

The rest is same as Command 200.

Example

When setting speed as "100.001", depart speed as "50.002", and approach speed as "60.003".

Command	Response
00C9H 0001H 86A1H 0000H C352H	00C9H 0000H 0000H
00CAH 0000H EA63H	00CAH 0000H 0000H

Command 203: Set CP Speeds In Table

Registers speeds for CP motion to the Arm speed table.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 | 1 0	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.

	bit	Name	Description
Parameter 2	15 14 | 1 0	*speed* High-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 | 1 0	*speed* Low-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 | 1 0	*departSpeed* High-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 5	15 14 | 1 0	*departSpeed* Low-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 6	15 14 | 1 0	*approachSpeed* High-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 7	15	*approachSpeed* Low-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Registers speed, depart speed, and approach speed to the specified table.

The issue of this command does not affect actual settings. The settings will be reflected when they are specified as options at execution of the target motion command.

This command is used in combination with the motion commands.

Example

When registering speed as “100.001”, depart speed as “50.002”, and approach speed as “60.003” to the table 1.

Command	Response
00CBH 0001H 0001H 86A1H 0000H C352H 0000H EA63H	00CBH 0000H 0000H

Command 204: Set CP Speed, Depart Speed In Table

Registers speeds for CP motion to the Arm speed table separately.
Registers speed setting value and depart speed value.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*speed* High-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*speed* Low-order word	Specifies the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*departSpeed* High-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*departSpeed* Low-order word	Specifies the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command registers speed and depart speed to the speed table when registering arm speeds for CP motion separately.

This command functions in combination with Command 205.

To set speeds, execute Commands 204 and 205, in that order.
Registration will be completed by executing Command 205.

If commands other than Command 205 are issued after this command, setting will be canceled.

Example

When registering speed as “100.001”, depart speed as “50.002”, and approach speed as “60.003” to the table 1.

Command	Response
00CCH 0001H 0001H 86A1H 0000H C352H	00CCH 0000H 0000H
00CDH 0001H 0000H EA63H	00CDH 0000H 0000H

Command 205: Set CP Approach Speed In Table

Registers speeds for CP motion to the Arm speed table separately.
Registers approach speed.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*approachSpeed* High-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*approachSpeed* Low-order word	Specifies the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command registers speed and approach speed to the speed table when registering arm speeds for CP motion separately.

This command functions in combination with Command 204.

To set speeds, execute Commands 204 and 205, in that order.
Registration will be completed by executing Command 205.

If the previous command is not Command 204, an error response will be returned.

Example

When registering speed as “100.001”, depart speed as “50.002”, and approach speed as “60.003” to the table 1.

Command	Response
00CCH 0001H 0001H 86A1H 0000H C352H	00CCH 0000H 0000H
00CDH 0001H 0000H EA63H	00CDH 0000H 0000H

Command 206: Get CP Speeds

Acquires Arm speed settings for CP motion.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*speed* High-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*speed* Low-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*departSpeed* High-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*departSpeed* Low-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 5	15	*approachSpeed* High-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 6	15	*approachSpeed* Low-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires current Arm speed settings for CP motion.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003”.

Command	Response
00CEH	00CEH 0001H 86A1H 0000H C352H 0000H EA63H

Command 207: Get CP Speed, Depart Speed From Table

Acquires Arm speed settings for CP motion separately.
Acquires speed setting value and depart speed value.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*speed* High-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*speed* Low-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*departSpeed* High-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*departSpeed* Low-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command acquires speed and depart speed when acquiring arm speeds for CP motion separately.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003”.

Command	Response
00CFH	00CFH 0001H 86A1H 0000H C352H

Command 208: Get CP Approach Speed From Table

Acquires Arm speed settings for CP motion separately.
Acquires the approach speed value.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*approachSpeed* High-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 2	15	*approachSpeed* Low-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Description

This command acquires approach speed when acquiring arm speeds for CP motion separately.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003”.

Command	Response
00CFH	00D0H 0000H EA63H

Command 209: Get CP Speeds From Table

Acquires registered values from the Arm speed table for CP motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*speed* High-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*speed* Low-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*departSpeed* High-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*departSpeed* Low-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 6	bit	Name	Description
	15	*approachSpeed* High-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 7	bit	Name	Description
	15	*approachSpeed* Low-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires registered values from the specified position of the arm speed table for CP motion.
Speed, depart speed, and approach speed can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003” in the table 1.

Command	Response
00D1H 0001H	00D1H 00001H 0001H 86A1H 0000H C352H 0000H EA63H

Command 210: Get CP Speed, Depart Speed From Table

Acquires registered values from the Arm speed table for CP motion separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*speed* High-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*speed* Low-order word	Returns the value which increased the speed (integer, unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*departSpeed* High-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*departSpeed* Low-order word	Returns the value indicating the depart speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires registered values from the specified position of the arm speed table for CP motion.
Speed and depart speed can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003” in the table 1.

Command	Response
00D2H 0001H	00D2H 00001H 0001H 86A1H 0000H C352H

Command 211: Get CP Approach Speed From Table

Acquires registered values from the Arm speed table for CP motion separately.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*approachSpeed* High-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*approachSpeed* Low-order word	Returns the value indicating the approach speed for Jump3 (Unit: mm/sec) × 1000 and converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires registered values from the specified position of the arm speed table for CP motion.
Registered approach speed can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When speed is “100.001”, depart speed is “50.002”, and approach speed is “60.003” in the table 1.

Command	Response
00D3H 0001H	00D3H 00001H 0000H EA63H

10.6 Speed Setting for Tool Orientation Change in CP Motion When Using ROT

These commands are used to set and display the speed setting for Tool orientation change in CP motion when using ROT.

The commands are available when ROT decoration parameter is specified in motion commands Move, Arc, Arc3, and Jump3CP.

CAUTION	■ Setteing will be initialized in following cases: Controller’s power is turned ON Motor ON is executed When excitation control is executed Reset is executed Halt button or Ctrl+C are pressed

Command 250: Set ROT Speed

Sets the speed setting for Tool orientation change in CP motion when using ROT.

Command Syntax

Parameter 1	bit	Name	Description
	15 14 │ 1 0	*speed* High-order word	Specifies the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.

Parameter 2	bit	Name	Description
	15 14 │ 1 0	*speed* Low-order word	Specifies the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the speed setting for Tool orientation change in CP motion when using ROT.

Setting value should be specified as fixed-point data which validates to three decimal places.

Example

When setting “1000” as the speed setting value.

Command	Response
00FAH 000FH 4240H	00FAH 0000H 0000H

Command 251: Set ROT Speed In Table

Registers the speed setting to the Tool orientation change speed table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the registration position in the table using an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*speed* High-order word	Specifies the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*speed* Low-order word	Specifies the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Registers the speed setting to the specified table.

The issue of this command does not affect actual settings. The settings will be reflected when they are specified as options at execution of the target motion command.

This command is used in combination with the motion commands.

Example

When registering “1000” for the speed setting value in the table 1.

Command	Response
00FBH 00001H 000FH 4240H	00FBH 0000H 0000H

Command 252: Get ROT Speed

Acquires the speed setting for Tool orientation change in CP motion when using ROT.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*speed* High-order word	Returns the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*speed* Low-order word	Returns the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the speed setting for Tool orientation change in CP motion when using ROT.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When the speed setting value is “1000”.

Command	Response
00FCH	00FCH 000FH 4240H

Command 253: Get ROT Speed From Table

Acquires the speed setting from the Tool orientation change speed table.

Command Syntax

	bit	Name	Description
Parameter 1	15	*tableNumber*	Specifies the specified table number in an integer.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*tableNumber*	Returns the specified table number in an integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*speed* High-order word	Returns the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*speed* Low-order word	Returns the Tool orientation change speed in CP motion (Integer larger than 0.1. Unit: deg/sec) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the speed setting from the specified position in the Tool orientation change speed table.
Registered speed setting value can be acquired by using this command.
If the specified table number is out of range or not registered, an error response will be returned.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When the speed setting value is “1000” in the table 1.

Command	Response
00FDH 0001H	00FDH 00001H 000FH 4240H

10.7 Parameter Setting for Speed and Accel/Decel Offset in PTP Motion

These commands are used to set the parameters for offsetting the speed and accel/decel in PTP motion.

Specify the parameter to operate the maximum acceleration and deceleration in PTP motion. For Weight command, set the weight of the Hand and a work piece.

Arm length designation is only necessary for the SCARA robots (including RC series). The length is a distance from the center of the Joint #2 to the center of the Joint #3. This is invalid for the models other than the SCARA robots (including RS series).

If the equivalent transfer weight calculated by the setting value exceeds the maximum weight capacity, an error will occur.

Robot parameter data is stored to the compact flash in the Controller. Therefore, execution of the command writes to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Potential errors

If the value exceeds the maximum allowable load

When the equivalent load weight calculated from the value entered exceeds the maximum load weight, an error will occur.

Potential Damage to the Manipulator Arm

Note that if the hand weight for Weight is significantly less than the actual weight, exesive acceleration and deceleration values will be set and may cause damage to the manipulator.

CAUTION	■ Weight values are not changed by turning main power Off

Command 300: Set Weight And Length

Sets the parameter setting for offsetting the speed and accel/decel in PTP motion.
Sets the hand weight and the arm length.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 | 1 0	*handWeight* High-order word	Specifies the Hand weight to be added to the Arm as the value × 1000 and convering it to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 2	15 14 | 1 0	*handWeight* Low-order word	Specifies the Hand weight to be added to the Arm as the value × 1000 and convering it to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 | 1 0	*armLength* High-order word	Specifies the length from the rotational center of Arm #2 to the center of the gravity of Arm #3 (Unit: mm) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 | 1 0	*armLength* Low-order word	Specifies the length from the rotational center of Arm #2 to the center of the gravity of Arm #3 (Unit: mm) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the parameter setting for speed and accel/decel offset in PTP motion by specifying the hand weight and the arm length.

Example

When setting 5.12 Kg for the hand weight, and 120.001 for the arm length.

Command	Response
012CH 0000H 1400H 0001H D4C1H	012CH 0000H 0000H

Command 301: Set Weight

Sets the parameter setting for offsetting the speed and accel/decel in PTP motion.
Sets the Hand weight.

Command Syntax

Parameter 1	bit	Name	Description
	15	*handWeight* High-order word	Specifies the Hand weight to be added to the Arm as the value × 1000 and convering it to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*handWeight* Low-order word	Specifies the Hand weight to be added to the Arm as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets only the hand weight of the parameter setting for speed and accel/decel offset in PTP motion.

Example

When setting 5.12Kg for the hand weight.

Command	Response
012DH 0000H 1400H	012DH 0000H 0000H

Command 302: Get Weight And Length

Acquires the parameter setting for offsetting the speed and accel/decel in PTP motion.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*handWeight* High-order word	Returns the Hand weight to be added to the Arm as the value × 1000 and convering it to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*handWeight* Low-order word	Returns the Hand weight to be added to the Arm as the value × 1000 and convering it to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*armLength* High-order word	Returns the length from the center of Joint #2 to the center of Joint #3 (Unit: mm) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*armLength* Low-order word	Returns the length from the center of Joint #2 to the center of Joint #3 (Unit: mm) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the parameter setting for offsetting the speed and accel/decel in PTP motion.
This command acquires the hand weight and the arm length.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When the hand weight is 5.12 Kg, and the arm length is 120.001.

Command	Response
012EH	012EH 0000H 1400H 0001H D4C1H

10.8 Load Inertia and Eccentricity Setting

These commands are used to specify the moment of inertia around the end effector. By using the commads, acceleration/deceleration and servo gain of the end effector can be offset properly. Also, you can specify the distacnce from the center of the end effector to the gravity center of the hand and the work piece by using the eccentricity parameter.

Robot parameter data is stored to the compact flash in the Controller. Therefore, execution of the command writes to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Command 350: Set Inertia And Eccentricity

Sets the load inertia and eccentricity.

Command Syntax

	bit	Name	Description
Parameter 1	15	*loadInertia* High-order word	Specifies the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*loadInertia* Low-order word	Specifies the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*eccentricity* High-order word	Specifies the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*eccentricity* Low-order word	Specifies the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Setting value should be specified as fixed-point data which validates to three decimal places.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
015EH 0000H 1400H 0001H D4C1H	015EH 0000H 0000H

Command 351: Set Inertia

Sets the load inertia.

Command Syntax

Parameter 1	bit	Name	Description
	15	*loadInertia* High-order word	Specifies the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*loadInertia* Low-order word	Specifies the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is used to set the load inertia when setting the load inertia and the eccentricity separately.
This command must be used in combination with Command 352.

To set the load inertia and the eccentricity, execute Command 351 and 352, in that order. The settings will be effective after issuing Command 352.
If commands other than Command 352 are issued after this command, the new values will be canceled.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
015FH 0000H 1400H	015FH 0000H 0000H
0160H 0001H D4C1H	0160H 0000H 0000H

Command 352: Set Eccentricity

Sets the eccentricity.

Command Syntax

Parameter 1	bit	Name	Description
	15	*eccentricity* High-order word	Specifies the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*eccentricity* Low-order word	Specifies the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is used to set the eccentricity when setting the load inertia and the eccentricity separately.
This command must be used in combination with Command 351.

To set the load inertia and the eccentricity, execute Command 351 and 352, in that order. The settings will be effective after issuing Command 352.
If the last command is not Command 351, an error response will be returned.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
015FH 0000H 1400H	015FH 0000H 0000H
0160H 0001H D4C1H	0160H 0000H 0000H

Command 353: Get Inertia And Eccentricity

Acquires the load inertia and the eccentricity.

Command Syntax

No parameter.

Response Syntax

<table>
<tr><th rowspan="6">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">loadInertia
High-order word</td><td rowspan="5">Returns the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm²) as the value × 1000 converted to a 32-bit integer.
High-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">loadInertia
Low-order word</td><td rowspan="5">Returns the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm²) as the value × 1000 converted to a 32-bit integer.
Low-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 3</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">eccentricity
High-order word</td><td rowspan="5">Returns the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer.
High-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 4</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">eccentricity
Low-order word</td><td rowspan="5">Returns the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer.
Low-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

Description

Acquires the current load inertia and the eccentricity.

The values will be returned as fixed-point data which validates to three decimal places.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
0161H	0161H 0000H 1400H 0001H D4C1H

Command 354: Get Inertia

Acquires the load inertia.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*loadInertia* High-order word	Returns the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*loadInertia* Low-order word	Returns the moment of inertia around the center of the end effector including the hand and the work piece (real number, Unit: Kgm^2) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command is used to acquire the load inertia when acquiring the current load inertia and the eccentricity separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
0162H	0162H 0000H 1400H

Command 355: Get Eccentricity

Acquires the eccentricity.

Command Syntax

No parameter.

Response Syntax

Response 1	bit	Name	Description
	15	*eccentricity* High-order word	Returns the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Response 2	bit	Name	Description
	15	*eccentricity* Low-order word	Returns the distance from the center of the end effector to the gravity center of the hand and the work piece by specifying the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command is used to acquire the eccentricity when acquiring the current load inertia and the eccentricity separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting 5.12 Kg^2 for the load inertia and 120.001 mm for the eccentricity.

Command	Response
0163H	0163H 0001H D4C1H

10.9 Arch Parameter Setting

These commands are used to set arch parameters for Jump, Jump3, and Jump3CP commands. They define values in the Arch table which are necessary for Jump motion commands (2001, 2002, and 2003).

For details of Arch, refer to *Arch* in the *SPEL+ Language Reference*.

Command 400: Set Arch

Sets Arch parameters (depart distance and approach distance).

Command Syntax

	bit	Name	Description
Parameter 1	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*departDistance* High-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*departDistance* Low-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*approachDistance* High-order word	Specifies the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*approachDistance* Low-order word	Specifies the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets Arch parameters (depart distance and approach distance).

Setting value should be specified as fixed-point data which validates to three decimal places.

* Approach distance for Jump command (2001): approach distance (vertical distance from the target position) after the completion of horizontal movement.
* Approach distance for Jump3 command (2002) and Jump3CP command (2003): approach distance after the completion of span motion.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
0190H 0003H 0000H 278BH 0000H 4E9BH	0190H 0000H 0000H

Command 401: Set Arch Depart Distance

Sets Arch depart distance.

Command Syntax

	bit	Name	Description
Parameter 1	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*deparDdistance* High-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*deparDdistance* Low-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is used to set the depart distance when setting Arch parameters separately.
This command must be used in combination with Command 402.

To set the parameters, execute Command 401 and 402, in that order.
The settings will be effective after issuing Command 402.
If commands other than Command 402 are issued after this command, the new setting will be canceled.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
0191H 0003H 0000H 278BH	0191H 0000H 0000H

Command 402: Set Arch Approach Distance

Sets Arch approach distance.

Command Syntax

Parameter 1	bit	Name	Description
	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*approachDistance* High-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*approachDistance* Low-order word	Specifies the depart distance as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is used to set the approach distance when setting Arch parameters separately.
This command must be used in combination with command No. 402.

To set the parameters, execute commands No. 401 and No. 402, in that order.
The settings will be effective after issuing command No. 402.
If the last command is not the command No. 401, an error response will be returned.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
0192H 0003H 0000H 4E9BH	0192H 0000H 0000H

Command 403: Get Arch

Acquires Arch parameter setting values.
Acquires the depart distance and the approach distance.

Command Syntax

	bit	Name	Description
Parameter 1	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*archNumber*	Returns the arch number by an integer from 0 to 6. Returns the specified table number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*departDistance* High-order word	Specifies the depart distance (vertical distance from the start point) as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*departDistance* Low-order word	Returns the depart distance (vertical distance from the start point) as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*approachDistance* High-order word	Returns the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Response 5	bit	Name	Description
	15	*approachDistance* Low-order word	Returns the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Description

Acquires the Arch parameter setting values (depart distance and approach distance) from th specified arch table.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
0193H 0003H	0190H 0003H 0000H 278BH 0000H 4E9BH

Command 404: Get Arch Depart Distance

Acquires Arch parameter setting values separately.
Acquires the depart distance.

Command Syntax

	bit	Name	Description
Parameter 1	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*archNumber*	Returns the arch number by an integer from 0 to 6. Returns the specified table number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*departDistance* High-order word	Returns the depart distance (vertical distance from the start point) as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*departDistance* Low-order word	Returns the depart distance (vertical distance from the start point) as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command is used to set the depart distance when setting Arch parameters separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
0194H 0003H	0194H 0003H 0000H 278BH

Command 405: Get Arch Approach Distance

Acquires Arch parameter setting values separately.
Acquires the approach distance.

Command Syntax

	bit	Name	Description
Parameter 1	15	*archNumber*	Specifies the arch number by an integer from 0 to 6. Valid values are integers from 0 to 6.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*archNumber*	Returns the arch number by an integer from 0 to 6. Returns the specified table number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*approachDistance* High-order word	Returns the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*approachDistance* Low-order word	Returns the approach distance after the completion of horizontal movement as the value (Unit: mm) × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

This command is used to acquire the approach distance when acquiring Arch parameters separately.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting 10.123 mm for the depart distance and 20.123 mm for the approach distance in the Arch number 3.

Command	Response
01954H 0003H	0195H 0003H 0000H 4E9BH

10.10 Setting of Positioning Error Ranges

These commands are used to specify the allowable positioning error for detecting completion of any given move for each joint.

This positioning completion check begins after the CPU has completed sending the target position pulse to the servo system. Due to servo delay, the manipulator will not yet have reached the target position. This check continues to be executed every few milliseconds until each joint has arrived within the specified range setting. Positioning is considered complete when all axes have arrived within the specified ranges. Once positioning is complete program control is passed to the next statement, however, servo system keeps the control of the manipulator target position.

When relatively large ranges are used with command, the positioning will be confirmed relatively early in the move, and executes the next statement.

The default settings depend on the robot type. The default settings depend on the manipulator type. Refer to your manipulator manual for details.

Cycle Times and the Fine Command

The Fine value does not affect the acceleration or deceleration control of the manipulator arm. However, smaller Fine values can cause the system to run slower because it may take the servo system extra time (a few milliseconds) to get within the acceptable position range. Once the arm is located within the acceptable position range (defined by the Fine instruction), the CPU executes the next user instruction. (Note that all activated axes must be in position before the CPU executes the next user instruction.)

Initialization (by Motor On, SLock, SFree)

Any time the following commands are used the Fine value is initialized to default values:

SLock, SFree, Motor instructions

Make sure that you reset Fine values after one of the above commands execute.

Command 450: Set Axis 1 - 6 Positioning Error Ranges

This command specifies the allowable positioning error for detecting completion of any given move for each joint.

Command Syntax

* Axis 5 and 6 are not necessary for the 4-axis robots.

	bit	Name	Description
Parameter 1	15	*axis1*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axis2*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*axis3*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*axis4*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*axis5*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 6	15	*axis6*	Integer ranging from (0-65535) which represents the allowable positioning error
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command specifies the allowable positioning error for detecting completion of any given move for each joint.
4-axis manipulator: specify the parameters from *axis1* to *axis4*.
6-axis manipulator: specify the parameters from *axis1* to *axis6*.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C2H 0064H 00C8H 012CH 0190H 01F4H 0258H	01C2H 0000H 0000H

Command 451: Set Axis 1 – 3 Positioning Error Ranges

Specifies the allowable positioning error for detecting completion of any given move for each joint separately. Sets *axis1* to *axis3*.

Command Syntax

	bit	Name	Description
Parameter1	15	*axis1*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axis2*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*axis3*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command specifies Axis #1, #2, and #3 to specify the allowable positioning error for detecting completion of any given move.
This command functions in combination with Command 451.

To set parameters, execute Command 451 and 452, in that order.
The settings will be effective after issuing Command 452.
If commands other than Command 452 are issued after this command, setting will be canceled.

The rest is same as Command 450.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C3H 0064H 00C8H 012CH	01C3H 0000H 0000H

Command 452: Set Axis 4 – 6 Positioning Error Ranges

Specifies the allowable positioning error for detecting completion of any given move for each joint separately. Sets *axis4* to *axis6*.

Command Syntax

* For 4-axis manipulators, Parameter 2 and 3 are not necessary.

	bit	Name	Description
Response 1	15	*axis4*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*axis5*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*axis6*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command specifies Axis #4, #5, and #6 to specify the allowable positioning error for detecting completion of any given move. For 4-axis manipulators, *axis5* and *axis6* are not necessary.
This command functions in combination with Cmmand 451.

To set parameters, execute Command 451 and 452, in that order.
The settings will be effective after issuing Command 452.
If the last command is not Command 451, an error response will be returned.

The rest is same as Command 450.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C4H 0190H 01F4H 0258H	01C4H 0000H 0000H

Command 453: Get Axis 1 – 6 Positioning Error Ranges

Acquires setting values of the allowable positioning error for detecting completion of any given move for each joint.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*axis1*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*axis2*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*axis3*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*axis4*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 5	15	*axis5*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 6	15	*axis6*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

* For 4-axis manipulators, Parameter 2 and 3 are not returned.

Description

This command acquires setting values of all axes for specifying the allowable positioning error for detecting completion collectively.
However, values for *axis5* and *axis6* are not returned if the manipulator is the 4-axis manipulator.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C5H	01C5H 0064H 00C8H 012CH 0190H 01F4H 0258H

Command 454: Get Axis 1 – 3 Positioning Error Ranges

Acquires values for specifying the allowable positioning error for detecting motion completion separately. Acquires Axis #1, #2, and #3.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*axis1*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*axis2*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*axis3*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	|		
	1		
	0		

Description

This command acquires setting values of Axis #1, #2, and #3 for specifying the allowable positioning error for detecting completion when acquiring the setting values separately.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C6H	01C6H 0064H 00C8H 012CH

Command 455: Get Axis 4 - 6 Positioning Error Ranges

Acquires values for specifying the allowable positioning error for detecting motion completion separately.
Acquires Axis #4, #5, and #6.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*axis4*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 2	15	*axis5*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 3	15	*axis6*	Integer ranging from (0-65535) which represents the allowable positioning error.
	14		
	｜		
	1		
	0		

Description

This command acquires setting values of Axis #4, #5, and #6 for specifying the allowable positioning error for detecting completion when acquiring the setting values separately.
However, values for *axis5* and *axis6* are not returned if the manipulator is a 4-axis manipulator.

Example

6-axis manipulator
Axis #1: 100
Axis #2: 200
Axis #3: 300
Axis #4: 400
Axis #5: 500
Axis #6: 600

Command	Response
01C7H	01C7H 0190H 01F4H 0258H

10.11 Tool Selection

This command selects the tool specified by the tool number (*toolNum*). When the tool number is 0, no tool is selected and all motion are done with respect to the center of the end effector joint. However, when Tool entry 1, 2, or 3 is selected, motion is done with respect to the end of the tool as defined with the tool definition.

Power Off and Its Effect on the Tool Selection

Power Off and Its Effect on the Tool Selection.

Life of Compact Flash

Robot parameter data is stored to the compact flash in the Controller. Therefore, execution of the command writes to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Command 500: Set Tool

Selects the current tool.

Command Syntax

	bit	Name	Description
Parameter 1	15	*toolNum*	Integer from 0-15 representing which of 16 tool definitions to use.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Selects the tool.

Example

When selecting the tool 5.

Command	Response
01F4H 0005H	01F4H 0000H 0000H

Command 501: Get Tool

Acquires the current tool.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*toolNumber*	Integer from 0-15 representing the current tool number among 16 tool definitions.
	14		
	｜		
	1		
	0		

Description

Returns the current tool number.

Example

When the tool 5 is selected.

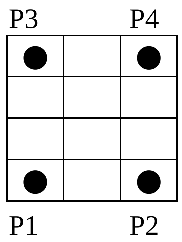
Command	Response
01F5H	01F5H 0005H

10.12 Pallet Definition

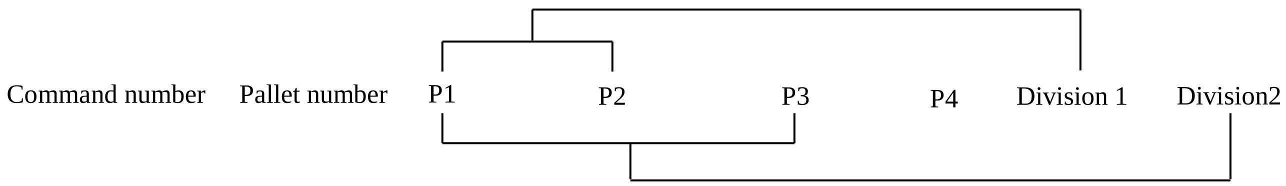
A pallet is defined by teaching points P1, P2 and P3 as a minimum to the manipulator, and by specifying the number of points from P1 to P2 and from P2 to P3. If the pallet is a well ordered rectangular shape, only 3 of the 4 corner points need to be specified. However, in most situations, it is recommended to use 4 corner points to define the pallet.

To define a pallet, first teach either 3 or 4 corner points.
Then, define the pallet as follows:

4-point definition: P1, P2, P3 (and P4) are shown below. There are 3 positions from P1-P2 and 4 positions from P1-P3. This makes a pallet which has 12 positions total. To define this pallet the syntax is as follows:



10	11	12
7	8	9
4	5	6
1	2	3



Command 550: Define 4 Point Pallet

Defines the pallet by specifying 4 points.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*P1*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*P2*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*P3*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*P4*	Use this parameter with the Point number 1 through 3 when defining the pallet by 4-point definition.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 6	15	*columns*	Integer representing the number of points on the P1-to-P2 side of the pallet. Range is from 1 to 32767. (columns×rows <32767)
	14		
	｜		
	1		
	0		

Parameter 7	bit	Name	Description
	15	*rows*	Integer representing the number of points on the P1-to-P3 side of the pallet. Range is from 1 to 32767. (columns×rows <32767)
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Defines a pallet by specifying 4 points.

Example

When defining the pallet 3 with 4 points (P1, P2, P3, and P4) and deviding P1-to-P2 side into 10 and P1-to-P3 side into 15.

Command	Response
0226H 0003H 0001H 0002H 0003H 0004H 000AH 000FH	0226H 0000H 0000H

Command 551: Define 3 Point Pallet

Defines a pallet by specifying 3 points.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*P1*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*P2*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*P3*	Specifies the point variable to use to define the pallet (standard 3-point definition).
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*columns*	Integer representing the number of points on the P1-to-P2 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 6	15	*rows*	Integer representing the number of points on the P1-to-P3 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	｜		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Defines a pallet by specifying 3 points.

Example

When defining the pallet 3 by three points (P1, P2, and P3) and dividing P1-to-P2 side into 10 and P1-to-P3 side into 15.

Command	Response
0227H 0003H 0001H 0002H 0003H 000AH 000FH	0227H 0000H 0000H

Command 552: Define 4 Point Pallet (Restricted)

Defines a pallet by putting the restrictions on the number of points and divisions.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 ｜ 1 0	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.

	bit	Name	Description
Parameter 2	15 ｜ 8	*P1*	Specifies the point number by an integer from 0 to 255.
	7 ｜ 0	*P2*	Specifies the point number by an integer from 0 to 255.

	bit	Name	Description
Parameter 3	15 ｜ 8	*P3*	Specifies the point number by an integer from 0 to 255.
	7 ｜ 0	*P4*	Specifies the point number by an integer from 0 to 255. *When P4 is not used, set the same number as P3.

	bit	Name	Description
Parameter 4	15 ｜ 8	*columns*	Integer representing the number of points on the P1-to-P2 side of the pallet. Range is from 1 to 255.
	7 ｜ 0	*rows*	Integer representing the number of points on the P1-to-P3 side of the pallet. Range is from 1 to 255.

Response Syntax

Refer to *7. Response code.*

Description

This command restrains the setting range of point numbers and division numbers to save the number of words to be used.
Setting range of each parameter is restricted to the numbers from 1 to 255.
When defining the pallet by 3 points, give P4 the same number as P3.

Example

When the pallet 3 is defined by 4-point definition:

P1=255, P2=254, P3=253, P4=252
columns=252, rows=251

Command	Response
0228H 0003H 00FFH 00FEH 00FDH 00FCH 00FBH 00FAH	0228H 0000H 0000H

Command 553: Define Pallet Using Data Type

Selects the data type and defines the pallet by dividing it.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*data*	Select the types of data 1 and data 2.
	14		0: Data 1= P1 / Data 2 = P2
	|		1: Data 1= P3 / Data 2 = P4
	1		2: Data 1 = columns 1 / Data 2 = rows 2
	0		3: Data 1= P3 / Data 2 = No data

	bit	Name	Description
Parameter 3	15	*data1*	Information selected in Parameter 2.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*data2*	Information selected in Parameter 2.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specifies the data type and defines the pallet by dividing it. This command can define a pallet while saving the number of words to be used.
For the following order and combinations, the command needs to be executed several times.

(3-point specification)

(1) Set “0” for *data*, P1 for *data1*, and P2 for *data2*.
(2) Set “3” for *data* and specify P3 for *data1*.
(3) Set “2” for *data*, “columns” for *data1*, and “rows” for *data2*.

(4-point specification)

(1) Set “0” for *data*, P1 for *data1*, and P2 for *data2*.
(2) Set “1” for *data*, P3 for *data1*, and P4 for *data2*.
(3) Set “2” for *data*, “columns” for *data1*, and “rows” for *data2*.

Actual pallet definition is executed upon receiving *data* 2 for both 3- and 4-point specifications.
If the command is not executed in the above order, an error response will return and the pallet definition will be canceled.

Example

When the pallet 3 is defined by 4-point definition:

P1=255, P2=254, P3=253, P4=252
columns=252, rows=251

Command	Response
0229H 0003H 0000H 00FFH 00FEH	0229H 0000H 0000H
0229H 0003H 0001H 00FDH 00FCH	0229H 0000H 0000H
0229H 0003H 0001H 00FBH 00FAH	0229H 0000H 0000H

Command 554: Get 4 Point Pallet Definition

Acquires the details of a 4-point definition of the specified pallet.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*palletNumber*	Returns the pallet number of the specified pallet.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*P1*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*P2*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15	*P3*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 5	15	*P4*	Point variable of 4-point definition.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 6	15	*columns*	Integer representing the number of points on the P1-to-P2 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 7	15	*rows*	Integer representing the number of points on the P1-to-P3 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	│		
	1		
	0		

Description

Acquires the details of a 4-point pallet definition of the specified pallet all at once.
If the specified pallet is not registered or defined by 3 points, an error response will be returned.

Example

When the pallet 3 is defined by 4-point definition:

P1=1, P2=2, P3=3, P4=4
columns=10, rows=15

Command	Response
022AH 0003H	022AH 0003H 0001H 0002H 0003H 0004H 000AH 000FH

Command 555: Get 3 Point Pallet Definition

Acquires the details of 3-point definition of the specified pallet.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	│		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*palletNumber*	Returns the pallet number of the specified pallet.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 2	15	*P1*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 3	15	*P2*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 4	15	*P3*	Returns the point variable to be used to define the pallet (standard 3-point definition).
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 5	15	*columns*	Integer representing the number of points on the P1-to-P2 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 6	15	*rows*	Integer representing the number of points on the P1-to-P3 side of the pallet. Range is from 1 to 32767. (columns × rows <32767)
	14		
	│		
	1		
	0		

Description

Acquires the details of a 3-point pallet definition of the specified pallet all at once.
If the specified pallet is not registered or defined by 4 points, an error response will be returned.

Example

When the pallet 3 is defined by 3-point definition:

P1=1, P2=2, P3=3
columns=10, rows=15

Command	Response
022BH 0003H	022BH 0003H 0001H 0002H 0003H 000AH 000FH

Command 556: Get Pallet Definition (Restricted)

Acquires the details of a pallet definition defined with restrictions.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	palletNumber	Specifies the pallet number by an integer from 0 to 15.

Response Syntax

	bit	Name	Description
Response 1	15 14 │ 1 0	*palletNumber*	Returns the pallet number of the specified pallet.

	bit	Name	Description
Response 2	15 │ 8	*P1*	Returns the point number by an integer from 0 to 255.
	7 │ 0	*P2*	Returns the point number by an integer from 0 to 255.

	bit	Name	Description
Response 3	15 │ 8	*P3*	Returns the point number by an integer from 0 to 255.
	7 │ 0	*P4*	Returns the point number by an integer from 0 to 255. *If the specified pallet is defined by 3-point definition, the same number as P3 is returned.

	bit	Name	Description
Response 4	15 │ 8	*columns*	Integer ranging from 1 to 255 which represents the number of points on the P1-to-P2 side of the pallet.
	7 │ 0	*rows*	Integer ranging from 1 to 255 which represents the number of points on the P1-to-P3 side of the pallet.

Description

Acquires the details of a pallet definition defined with restrictions by Command 552.
If the specified pallet is not registered, or point numbers and division numbers are out of range, an error response will be returned.
If the specified pallet is defined by 3-point definition, the same number as P3 will be returned to P4.

Example

When the pallet 3 is defined by 4-point definition:

P1=1, P2=2, P3=3, P4=4
columns=10, rows=15

Command	Response
022CH 0003H	022CH 0003H 0102H 0304H 0A0FH

When the pallet 3 is defined by 3-point definition:

P1=1, P2=2, P3=3
columns=10, rows=15

Command	Response
022CH 0003H	022CH 0003H 0102H 0303H 0A0FH

Command 557: Get Pallet Definition Using Data Type

Acquires the details of a pallet definition for the specified pallet by specifying the data type.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*data*	Select the types of data 1 and data 2.
	14		0: Data 1= P1 / Data 2 = P2
	｜		1: Data 1= P3 / Data 2 = P4
	1		2: Data 1 = columns 1 / Data 2 = rows 2
	0		3: Data 1= P3 / Data 2 = No data

Response Syntax

	bit	Name	Description
Response 1	15	*palletNumber*	Returns the pallet number of the specified pallet.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 2	15	*data*	Returns the types of data 1 and data 2.
	14		0: Data 1= P1 / Data 2 = P2
	｜		1: Data 1= P3 / Data 2 = P4
	1		2: Data 1 = columns 1 / Data 2 = rows 2
	0		3: Data 1= P3 / Data 2 = No data

	bit	Name	Description
Response 3	15	*Data1*	Data returned by data selection.
	14		
	｜		
	1		
	0		

Response 4	bit	Name	Description
	15	*Data2*	Data returned by data selection. *Data will not be returned if Data selection is 3.
	14		
	│		
	1		
	0		

Description

This command acquires the details of pallet definition for the specified pallet separately.
Acquires data by selecting its type.
Data selection can be in a random order.
An error response will be returned in following cases:

- The specified pallet is not defined.
- The specified pallet is defined by 3-point definition, while “3” is selected in Data selection.
- The specified pallet is defined by 4-point definition, while “3” is selected in Data selection.

Example

When the pallet 3 is defined by 4-point definition:

P1=1, P2=2, P3=3, P4=4
columns=10, rows=15

Command	Response
022DH 0003H 0000H	022DH 0003H 0000H 0001H 0002H
022DH 0003H 0001H	022DH 0003H 0001H 0002H 0003H
022DH 0003H 0002H	022DH 0003H 0002H 000AH 000FH

When the pallet 3 is defined by 3-point definition:

P1=1, P2=2, P3=3
columns=10, rows=15

Command	Response
022DH 0003H 0000H	022DH 0003H 0000H 0001H 0002H
022DH 0003H 0001H	022DH 0003H 0003H 0002H
022DH 0003H 0002H	022DH 0003H 0002H 000AH 000FH

Command 558: Get Pallet Number Of Definition Points

Acquires the number of points for the specified pallet.

Command Syntax

	bit	Name	Description
Parameter 1	15	*palletNumber*	Specifies the pallet number by an integer from 0 to 15.
	14		
	⏐		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*Number of points*	0 = not defined 3 = 3-point definition 4 = 4-point definition
	14		
	⏐		
	1		
	0		

Description

Acquires whether the specified pallet is defined by 3-point or 4-point definition.
This command also acquires whether the pallet is defined or not.

Example

When the pallet 3 is defined by 4-point definition:

P1=1, P2=2, P3=3, P4=4
columns=10, rows=15

Command	Response
022EH 0003H	022EH 0004H

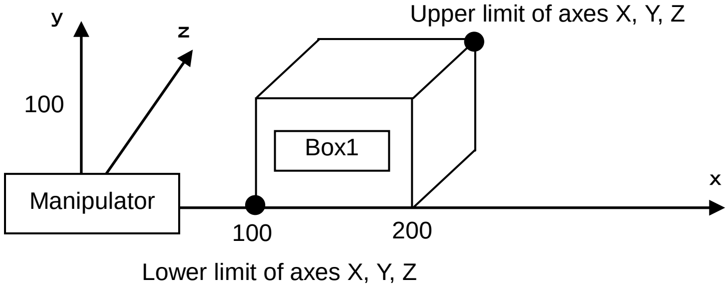
When the specified pallet is not defined.

Command	Response
022EH 0001H	022EH 0000H

10.13 Approach Check Area Settings

These commands set and acquire the approach check area (Box). The approach check area is for checking approaches of the robot end effector in the approach check area. The position of the end effector is calculated by the current tool. The approach check area is set on the base coordinate system of the manipulator and is between the specified maximum and minimum X, Y, and Z.

When the approach check area is used, the system detects approaches in any motor power status during the controller is ON.



Lower limit of axes X, Y, Z is (100, 0, 0) and upper limit is (200, 100, 100)

Robot parameter data is stored to the compact flash in the Controller. When you execute the command, the data is written to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Turning Off Approach Check Area by coordinate axis

You can turn off the approach check area of each coordinate axis. To turn off only the Z axis, define the lower limit position and the upper limit position of the *Z axis* to be 0.

Default values of Approach Check Area

The default values for the Box statement are "0, 0, 0, 0, 0, 0". (Approach Check Area Checking is turned off.)

Tool selection

The approach check is executed for the current tool. When you change the tool, the approach check may display the tool approach from inside to outside of the area or the other way although the robot is not operating.

Command 600: Set Box

Specifies the upper limit and lower limit positions for the specified approach check area.

Command Syntax

	bit	Name	Description
Parameter 1	15	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*Axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*Lower limit position* High-order word	Specifies the lower limit coordinate of the specified axis as the real number × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Lower limit position* Low-order word	Specifies the lower limit coordinate of the specified axis as the real number × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*Upper limit position* High-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 6	15	*Upper limit position* Low-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the approach check area for each axis by specifying the lower limit position and the upper limit position.
Setting will be completed by issuing the command to all axes, in order of X, Y, and Z.
If order of axes is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
0258H 0001H 0000H 0000H 0000H 0003H 0D40H	0258H 0000H 0000H
0258H 0001H 0001H 0001H 86A0H 0001H 86A0H	0258H 0000H 0000H
0258H 0001H 0002H 0000H 0000H 0001H 86A0H	0258H 0000H 0000H

Command 601: Set Box Lower Limit

Specifies the lower limit position of the specified approach check area.

Command Syntax

	bit	Name	Description
Parameter 1	15	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*Lower limit position* High-order word	Specifies the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Lower limit position* Low-order word	Specifies the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the lower limit position of the approach check area for the selected axis.
The setting will be effective after issuing the command for all axes, in order of X, Y, and Z.
If the order of axes is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

This command must be used in combination with Command 602.
Settings will be effective after specifying the lower limit position with Command 601, and then specifying the upper limit position with Command 602.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
0259H 0001H 0000H 0000H 0000H	0259H 0000H 0000H
0259H 0001H 0001H 0001H 86A0H	0259H 0000H 0000H
0259H 0001H 0002H 0000H 0000H	0259H 0000H 0000H
025AH 0001H 0000H 0003H 0D40H	025AH 0000H 0000H
025AH 0001H 0001H 0001H 86A0H	025AH 0000H 0000H
025AH 0001H 0002H 0001H 86A0H	025AH 0000H 0000H

Command 602: Set Box Upper Limit

Specifies the upper limit position for the specified approach check area.

Command Syntax

Parameter 1	bit	Name	Description
	15 14 │ 1 0	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.

Parameter 2	bit	Name	Description
	15 14 │ 1 0	*Axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis

Parameter 3	bit	Name	Description
	15 14 │ 1 0	*Upper limit position* High-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

Parameter 4	bit	Name	Description
	15 14 │ 1 0	*Upper limit position* Low-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the upper limit position of the approach check area for the selected axis.
The setting will be effective after issuing the command for all axes, in order of X, Y, and Z.
If the order of axes is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

This command must be used in combination with command No. 601.
The setting will be effective after specifying the lower limit position with command No. 601, and then specifying the upper limit position with command No. 602.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
0259H 0001H 0000H 0000H 0000H	0259H 0000H 0000H
0259H 0001H 0001H 0001H 86A0H	0259H 0000H 0000H
0259H 0001H 0002H 0000H 0000H	0259H 0000H 0000H
025AH 0001H 0000H 0003H 0D40H	025AH 0000H 0000H
025AH 0001H 0001H 0001H 86A0H	025AH 0000H 0000H
025AH 0001H 0002H 0001H 86A0H	025AH 0000H 0000H

Command 603: Get Box

Acquires the lower limit and upper limit positions of the specified approach check area.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.

	bit	Name	Description
Parameter 2	15 14 │ 1 0	*Axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis

Response Syntax

	bit	Name	Description
Response 1	15 14 │ 1 0	*areaNum*	Returns the specified area number.

	bit	Name	Description
Response 2	15 14 │ 1 0	*Axis selection*	Returns the specified axis. 0 = X axis 1 = Y axis 2 = Z axis

	bit	Name	Description
Response 3	15 14 │ 1 0	*Lower limit position* High-order word	Returns the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

Response 4	bit	Name	Description
	15	*Lower limit position* Low-order word	Returns the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response 5	bit	Name	Description
	15	*Upper limit position* High-order word	Returns the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Response 6	bit	Name	Description
	15	*Upper limit position* Low-order word	Returns the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the lower and upper limit positions of the specified axis for the specified area number.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
025BH 0001H 0000H	025BH 0001H 0000H 0000H 0000H 0003H 0D40H
025BH 0001H 0001H	025BH 0001H 0001H 0001H 86A0H 0001H 86A0H
025BH 0001H 0002H	025BH 0001H 0002H 0000H 0000H 0001H 86A0H

Command 604: Get Box Lower Limit

Acquires the approach check area lower limit position.

Command Syntax

	bit	Name	Description
Parameter 1	15	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*areaNum*	Returns the specified area number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*Axis selection*	Returns the specified axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*Lower limit position* High-order word	Specifies the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Response 4	bit	Name	Description
	15	*Upper limit position* Low-order word	Specifies the lower limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the lower limit position of the specified axis for the specified area number.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
025CH 0001H 0000H	025CH 0001H 0000H 0000H 0000H
025CH 0001H 0001H	025CH 0001H 0001H 0001H 86A0H
025CH 0001H 0002H	025CH 0001H 0002H 0000H 0000H
025DH 0001H 0000H	025DH 0001H 0000H 0003H 0D40H
025DH 0001H 0001H	025DH 0001H 0001H 0001H 86A0H
025DH 0001H 0002H	025DH 0001H 0002H 0001H 86A0H

Command 605: Get Box Upper Limit

Acquires the approach check area upper limit position.

Command Syntax

	bit	Name	Description
Parameter 1	15	*areaNum*	Specifies the area number to be set by an integer from 1 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*Axis selection*	Specifies the axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*areaNum*	Returns the specified area number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*Axis selection*	Returns the specified axis. 0 = X axis 1 = Y axis 2 = Z axis
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*Lower limit position* High-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 4	15 14 │ 1 0	*Upper limit position* Low-order word	Specifies the upper limit coordinate of the specified axis as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Description

Acquires the lower limit position of the specified axis for the specified area number.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When setting the Area 1:

	X axis	Y axis	Z axis
Lower limit position	0.000	100.000	0.000
Upper limit position	200.000	100.000	100.000

Command	Response
025CH 0001H 0000H	025CH 0001H 0000H 0000H 0000H
025CH 0001H 0001H	025CH 0001H 0001H 0001H 86A0H
025CH 0001H 0002H	025CH 0001H 0002H 0000H 0000H
025DH 0001H 0000H	025DH 0001H 0000H 0003H 0D40H
025DH 0001H 0001H	025DH 0001H 0001H 0001H 86A0H
025DH 0001H 0002H	025DH 0001H 0002H 0001H 86A0H

10.14 Approach Check Plane Settings

The approach check plane is used for checking whether the robot end effector is in one of the two areas divided by a plane. The position of the end effector is calculated by the current tool. The approach check plane is set using the XY plane of the base coordinate system. The approach check plane detects the end effector when it approaches the area on the + Z side of the approach check plane.

When the approach check plane is used, the system detects approaches in any motor power status while the controller is ON.

Specifies a coordinate system to create the approach check plane using the point data representing the translation and rotation based on the base coordinate system, and sets the approach check plane.

Robot parameter data is stored to the compact flash in the Controller. When you execute the command, the data is written to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Tool selection

The approach check is executed for the current tool. When you change the tool, the approach check may display the tool approach from inside to outside of the plane or the other way although the manipulator is not operating.

Command 650: Set Plane

Defines an approach check plane.

Command Syntax

Parameter 1	bit	Name	Description
	15	*planeNum*	Specifies the approach check plane number. Approach check plane can be defined by an integer from 1 to 15. Up to 15 approach check planes can be defined.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*coordinate selection*	Specifies the coordinate. 0 ＝ X 1 ＝ Y 2 ＝ Z 3 ＝ U 4 ＝ V 5 ＝ W
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*pCoordinateData* High-order word	Specifies the coordinate system of the approach check plane directory by a point data. Specifies the coordinate (real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*pCoordinateData* Low-order word	Specifies the coordinate system of the approach check plane directory by a point data. Specifies the coordinate (real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the approach check plane for the specified plane number for each coordinate.

Setting will be completed by issuing the command to all coordinates, in order of X, Y, Z, U, V, and W.
If order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.
4-axis manipulator: set the coordinates X, Y, Z, and U.
6-axis manipulator: set the coordinates X, Y, Z, U, V, and W.

Example

6-axis manipulator: *PlaneNum 1*

X:	100.123
Y:	200.123
Z:	300.123
U:	400.123
V:	500.123
W:	600.123

Command	Response
028AH 0001H 0000H 0001H 871BH	028AH 0000H 0000H
028AH 0001H 0001H 0003H 0DBBH	028AH 0000H 0000H
028AH 0001H 0002H 0004H 945BH	028AH 0000H 0000H
028AH 0001H 0003H 0006H 1AFBH	028AH 0000H 0000H
028AH 0001H 0004H 0007H A19BH	028AH 0000H 0000H
028AH 0001H 0005H 0009H 283BH	028AH 0000H 0000H

Command 651: Get Plane

Acquires an approach check plane definition.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 ｜ 1 0	*planeNum*	Specifies the approach check plane number. Approach check plane can be defined by an integer from 1 to 15. Up to 15 approach check planes can be defined.

	bit	Name	Description
Parameter 2	15 14 ｜ 1 0	*Coordinate selection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

Response Syntax

	bit	Name	Description
Response 1	15 14 ｜ 1 0	*PlaneNum*	Returns the plane number of the specified approach check plane.

	bit	Name	Description
Response 2	15 14 ｜ 1 0	*Coordinate selection*	Returns the specified coordinate. 0 ＝ X 1 ＝ Y 2 ＝ Z 3 ＝ U 4 ＝ V 5 ＝ W

	bit	Name	Description
Response 3	15 14 ｜ 1 0	*pCoordinateData* High-order word	Returns the coordinate (real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Response 4	15	*pCoordinateData* Low-order word	Returns the coordinate (real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the approach check plane of the specified plane number for each coordinate.

The value will be returned as fixed-point data which validates to three decimal places.

Example

6-axis manipulator: when setting the following to *PlaneNum 1*

X: 100.123
Y: 200.123
Z: 300.123
U: 400.123
V: 500.123
W: 600.123

Command	Response
028BH 0001H 0000H	028BH 0001H 0000H 0001H 871BH
028BH 0001H 0001H	028BH 0001H 0001H 0003H 0DBBH
028BH 0001H 0002H	028BH 0001H 0002H 0004H 945BH
028BH 0001H 0003H	028BH 0001H 0003H 0006H 1AFBH
028BH 0001H 0004H	028BH 0001H 0004H 0007H A19BH
028BH 0001H 0005H	028BH 0001H 0005H 0009H 283BH

10.15 Local Coordinate System Definition

These commands are used to define a local coordinate system.

Define a local coordinate system by specifying the origin and axis rotation angles with respect to the base coordinate system.

Command 700: Define Local

Defines a local coordinate system.

Command Syntax

	bit	Name	Description
Parameter 1	15	*localNumber*	Specifies the local coordinate system number. Local coordinate system can be defined by an integer from 1 to 15. Up to 15 coordinate systems can be defined.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*Coordinate selection*	Specifies the coordinate. 0 ＝ X 1 ＝ Y 2 ＝ Z 3 ＝ U 4 ＝ V 5 ＝ W
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*pCoordinateData* High-order word	Specifies the origin and the direction of the local coordinate system directory by a point data. Specifies the coordinate (real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*pCoordinateData* Low-order word	Specifies the origin and the direction of the local coordinate system directory by a point data. Specifies the coordinate (real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the local coordinate system of the specified coordinate number by each coordinate.

The setting will be effective after issuing the command for all coordinates, in order of X, Y, Z, U, V, and W.

If the order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

4-axis manipulator: set the coordinates X, Y, Z, and U.
6-axis manipulator: set the coordinates X, Y, Z, U, V, and W.

Example

6-axis manipulator: when setting the following to *PlaneNum 1*

X:	100.123
Y:	200.123
Z:	300.123
U:	400.123
V:	500.123
W:	600.123

Command	Response
02BCH 0001H 0000H 0001H 871BH	02BCH 0000H 0000H
02BCH 0001H 0001H 0003H 0DBBH	02BCH 0000H 0000H
02BCH 0001H 0002H 0004H 945BH	02BCH 0000H 0000H
02BCH 0001H 0003H 0006H 1AFBH	02BCH 0000H 0000H
02BCH 0001H 0004H 0007H A19BH	02BCH 0000H 0000H
02BCH 0001H 0005H 0009H 283BH	02BCH 0000H 0000H

Command 701: Get Local

Acquires a Local coordinate definition.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 ｜ 1 0	*localNumber*	Specifies the local coordinate system number. Local coordinate system can be defined by an integer from 1 to 15. Up to 15 coordinate systems can be defined.

	bit	Name	Description
Parameter 2	15 14 ｜ 1 0	*Coordinate selection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

Response Syntax

	bit	Name	Description
Response 1	15 14 ｜ 1 0	*localNumber*	Returns the specified number.

	bit	Name	Description
Response 2	15 14 ｜ 1 0	*Coordinate selection*	Returns the specified coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

	bit	Name	Description
Response 3	15 14 ｜ 1 0	*pCoordinateData* High-order word	Returns the coordinate (real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Response 4	15	*pCoordinateData* Low-order word	Returns the coordinate (real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

Description

Acquires the local coordinate system of the specified coordinate number by each coordinate.

The value will be returned as fixed-point data which validates to three decimal places.

Example

6-axis manipulator: *PlaneNum 1*

X: 100.123
Y: 200.123
Z: 300.123
U: 400.123
V: 500.123
W: 600.123

Command	Response
02BDH 0001H 0000H	02BDH 0001H 0000H 0001H 871BH
02BDH 0001H 0001H	02BDH 0001H 0001H 0003H 0DBBH
02BDH 0001H 0002H	02BDH 0001H 0002H 0004H 945BH
02BDH 0001H 0003H	02BDH 0001H 0003H 0006H 1AFBH
02BDH 0001H 0004H	02BDH 0001H 0004H 0007H A19BH
02BDH 0001H 0005H	02BDH 0001H 0005H 0009H 283BH

10.16 Motion Range Area Settings

These commands are used to specify the motion range area.

Many robot systems allow users to define joint limits, but these commands allow both joint limits and motion limits to be defined. In effect, this allows users to create a work envelope for their application.

The motion range established applies to motion command target positions only, and not to motion paths from starting position to target position. Therefore, the arm may move outside the XYLim range during motion.

Robot parameter data is stored to the compact flash in the Controller. When you execute the command, the data is written to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Turning Off Motion Range Checking

There are many applications which do not require Motion Range area setting. For that reason, there is a simple method to turn this setting off.

To turn off the setting, set the parameters (X axis lower limit / upper limit, Y axis lower limit / upper limit positions) to 0.

Default Motion Range Limit Values

The default values are “0, 0, 0, 0”. (Motion Range Limit Checking is OFF.)

Command 750: Set Motion Range

Specifies the motion range area upper limit and lower limit positions.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 ｜ 1 0	*coordinateSelection*	Specifies the coordinate. 0 ＝ X 1 ＝ Y 2 ＝ Z

	bit	Name	Description
Parameter 2	15 14 ｜ 1 0	*lowerLimit* High-order word	Specifies the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 ｜ 1 0	*lowerLimit* Low-order word	Specifies the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 ｜ 1 0	*upperLimit* High-order word	Specifies the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Parameter 5	15 14 ｜ 1 0	*upperLimit* Low-order word	Specifies the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the motion range area by specifying the lower limit position and the upper limit position for the selected coordinate.

The setting will be effective after issuing the command for all coordinates, in order of X, Y, and Z.

If order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

Example

When setting with the following coordinates.

	X axis	Y axis	Z axis
Lower limit position	10.000	20.000	30.000
Upper limit position	200.000	100.000	100.000

Command	Response
02EEH 0000H 0000H 000AH 0001H 871BH	02EEH 0000H 0000H
02EEH 0001H 0000H 0014H 0003H 0DBBH	02EEH 0000H 0000H
02EEH 0002H 0000H 001EH 0004H 945BH	02EEH 0000H 0000H

Command 751: Set Motion Range Lower Limit

Specifies the motion range area lower limit position.

Command Syntax

	bit	Name	Description
Parameter 1	15	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*lowerLimit* High-order word	Specifies the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*lowerLimit* Low-order word	Specifies the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the motion range area by specifying the lower limit position for the selected coordinate.

The setting will be effective after issuing the command for all coordinates, in order of X, Y, and Z.

If order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

This command must be used in combination with Command 752.

The setting will be executed by specifying the lower limit position with Command 751, and then specifying the upper limit position with Command 752.

Example

When coordinates are set as follows:

	X axis	Y axis	Z axis
Lower limit position	10.000	20.000	30.000
Upper limit position	200.000	100.000	100.000

Command	Response
02EFH 0000H 0000H 000AH	02EFH 0000H 0000H
02EFH 0001H 0000H 0014H	02EFH 0000H 0000H
02EFH 0002H 0000H 001EH	02EFH 0000H 0000H
02F0H 0000H 0001H 871BH	02F0H 0000H 0000H
02F0H 0001H 0003H 0DBBH	02F0H 0000H 0000H
02F0H 0002H 0004H 945BH	02F0H 0000H 0000H

Command 752: Set Motion Range Upper Limit

Specifies the motion range area upper limit position.

Command Syntax

Parameter 1	bit	Name	Description
	15 14 ｜ 1 0	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z

Parameter 2	bit	Name	Description
	15 14 ｜ 1 0	*upperLimit* High-order word	Specifies the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

Parameter 3	bit	Name	Description
	15 14 ｜ 1 0	*upperLimit* Low-order word	Specifies the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the motion range area by specifying the upper limit position for the selected coordinate.

The setting will be effective after issuing the command for all coordinates, in order of X, Y, and Z.

If the order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

This command must be used in combination with Command 751.

The setting will be effective after specifying the lower limit position with Command 751, and then specifying the upper limit position with Command 752.

Command 753: Get Motion Range

Acquires the motion range area upper limit and lower limit positions.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z

Response Syntax

	bit	Name	Description
Response 1	15 14 │ 1 0	*coordinateSelection*	Returns the specified coordinate. 0＝X 1＝Y 2＝Z

	bit	Name	Description
Response 2	15 14 │ 1 0	*lowerLimit* High-order word	Returns the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Response 3	15 14 │ 1 0	*lowerLimit* Low-order word	Returns the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.

	bit	Name	Description
Response 4	15 14 │ 1 0	*upperLimit* High-order word	Returns the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.

	bit	Name	Description
Response 5	15	*upperLimit* Low-order word	Returns the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the motion range area by returning the upper limit position and lower limit position for the selected coordinate.

The values will be returned as fixed-point data which validates to three decimal places.

Example

When coordinates are set as follows:

	X axis	Y axis	Z axis
Lower limit position	10.000	20.000	30.000
Upper limit position	200.000	100.000	100.000

Command	Response
02F1H 0000H	02F1H 0000H 0000H 000AH 0001H 871BH
02F1H 0001H	02F1H 0001H 0000H 0014H 0003H 0DBBH
02F1H 0002H	02F1H 0002H 0000H 001EH 0004H 945BH

Command 754: Get Motion Range Lower Limit

Acquires the motion range area lower limit position.

Command Syntax

	bit	Name	Description
Parameter 1	15	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*coordinateSelection*	Returns the specified coordinate. 0＝X 1＝Y 2＝Z
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*lowerLimit* High-order word	Returns the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*lowerLimit* Low-order word	Returns the lower limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the lower limit position of the motion range area for the selected coordinate.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When coordinates are set as follows:

	X axis	Y axis	Z axis
Lower limit position	10.000	20.000	30.000
Upper limit position	200.000	100.000	100.000

Command	Response
02F2H 0000H	02F2H 0000H 0000H 000AH
02F2H 0001H	02F2H 0001H 0000H 0014H
02F2H 0002H	02F2H 0002H 0000H 001EH

Command 755: Get Motion Range Upper Limit

Acquires the motion range area upper limit position.

Command Syntax

	bit	Name	Description
Parameter 1	15	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z
	14		
	｜		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*coordinateSelection*	Returns the specified coordinate. 0＝X 1＝Y 2＝Z
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 2	15	*upperLimit* High-order word	Returns the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Response 3	15	*upperLimit* Low-order word	Returns the upper limit coordinate where the Manipulator may travel as the actual value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	｜		
	1		
	0		

Description

Acquires the upper limit position of the motion range area for the selected coordinate.

The value will be returned as fixed-point data which validates to three decimal places.

Example

When coordinates are set as follows:

	X axis	Y axis	Z axis
Lower limit position	10.000	20.000	30.000
Upper limit position	200.000	100.000	100.000

Command	Response
02F3H 0000H	02F3H 0000H 0001H 871BH
02F3H 0001H	02F3H 0001H 0003H 0DBBH
02F3H 0002H	02F3H 0002H 0004H 945BH

10.17 Pulse Value Setting for Allowable Motion Area of Specified joint

These commands define the motion range for the specified joint with upper and lower limits in encoder pulse counts. While the Range command requires range settings for all six joints, the JRange command can set each joint's working limits individually. This reduces the number of parameters required.

Robot parameter data is stored on the compact flash in the Controller. When you execute the command, the data is written to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of this command.

Lower Limits Must Not Exceed Upper Limits:

The lower limit defined in the command must not exceed the upper limit. A lower limit in excess of the upper limit will cause an error, making it impossible to execute a motion command.

Factors to Change the Setting Values:

Once values are set, they will be kept until the user modifies the values by commands. Turning controller power off will not change the values.

Maximum and Minimum Working Ranges:

Refer to the specifications in the Manipulator manual for maximum working ranges for each manipulator model since these vary from model to model.

Command 800: Set Joint Range

Defines the lower limit and the upper limit of the permissible working range for the specified joint in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*jointNumber*	Specifies the joint number by an integer from 1 to 6.

	bit	Name	Description
Parameter 2	15 14 │ 1 0	*lowerLmit* High-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.

	bit	Name	Description
Parameter 3	15 14 │ 1 0	*lowerLmit* Low-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. Low-order side 16 bit.

	bit	Name	Description
Parameter 4	15 14 │ 1 0	*upperLmit* High-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. High-order side 16 bit.

	bit	Name	Description
Parameter 5	15 14 │ 1 0	*upperLmit* Low-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Defines the allowable motion range for the specified joint with upper and lower limits in encoder pulse counts.

The pulse value should be specified in 32-bit two's complement.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0320H 0001H FFFFH E890H 0000H 1B58H	0320H 0000H 0000H

Command 801: Set Joint Range Lower Limit

Defines the lower limit for the permissible working range of the specified joint in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15	*jointNumber*	Specifies the joint number by an integer from 1 to 6.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*lowerLmit* High-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*lowerLmit* Low-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command specifies the lower limit pulse when setting the pulses separately.
This command functions in combination with Command 801.

To set pulses, execute Command 801 and Command 802, in that order.
The settings will be effective after issuing Command 802.
If commands other than Command 802 are issued after this command, setting will be canceled.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0321H 0001H FFFFH E890H	0321H 0000H 0000H
0322H 0001H 0000H 1B58H	0322H 0000H 0000H

Command 802: Set Joint Range Upper Limit

Defines the upper limit of the permissible working range of the specified joint in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15	*jointNumber*	Specifies the joint number by an integer from 1 to 6.
	14		
	|		
	1		
	0		

	bit	Name	-{}-Description
Parameter 2	15	*upperLmit* High-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*upperLmit* Low-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command specifies the upper limit pulse when setting the pulses separately.
This command functions in combination with Command 801.

To set pulses, execute Command 801 and Command 802, in that order.
The settings will be effective after issuing Command 802.
If the last command is not Command 801, an error response will be returned.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0321H 0001H FFFFH E890H	0321H 0000H 0000H
0322H 0001H 0000H 1B58H	0322H 0000H 0000H

Command 803: Get Joint Range

Acquires the permissible working range of the specified joint in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*jointNumber*	Specifies the joint number by an integer from 1 to 6.

Response Syntax

	bit	Name	Description
Response 1	15 14 │ 1 0	*jointNumber*	Returns the joint number by an integer from 1 to 6.

	bit	Name	Description
Response 2	15 14 │ 1 0	*lowerLmit* High-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.

	bit	Name	Description
Response 3	15 14 │ 1 0	*lowerLmit* Low-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. Low-order side 16 bit.

	bit	Name	Description
Response 4	15 14 │ 1 0	*upperLmit* High-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. High-order side 16 bit.

	bit	Name	Description
Response 5	15	*upperLmit* Low-order side	Integer in complement form representing the upper limit pulse of the motion range for the specified joint. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the current lower and upper limit pulses of the permissible motion range of the specified joint.
The pulse values are returned in 32-bit two's complement format.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0323H 0001H	0323H 0001H FFFFH E890H 0000H 1B58H

Command 804: Get Joint Range Lower Limit

Acquires the lower limit of the permissible working range of the specified joint in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15	*jointNumber*	Specifies the joint number by an integer from 1 to 6.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*jointNumber*	Specifies the joint number by an integer from 1 to 6.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*lowerLmit* High-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*lowerLmit* Low-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the current lower limit pulse of the permissible motion range of the specified joint.
The pulse value is returned in 32-bit two's complement format.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0324H 0001H	0324H 0001H FFFFH E890H

Command 805: Get Joint Range Upper Limit

Acquires the upper limit of the motion range area setting in pulses.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*jointNumber*	Specifies the joint number by an integer from 1 to 6.

Response Syntax

	bit	Name	Description
Response 1	15 14 │ 1 0	*jointNumber*	Returns the joint number by an integer from 1 to 6.

	bit	Name	Description
Response 2	15 14 │ 1 0	*upperLmit* High-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.

	bit	Name	Description
Response 3	15 14 │ 1 0	*upperLmit* Low-order side	Integer in complement form representing the lower limit pulse of the motion range for the specified joint. High-order side 16 bit.

Description

Acquires the current upper limit pulse of the permissible motion range of the specified joint.
The pulse value is returned in 32-bit two's complement format.

Example

When setting – 6000 for the lower limit pulse of the Joint #1 and 7000 for the upper limit pulse.

Command	Response
0325H 0001H	0325H 0001H 0000H 1B58H

10.18 Base Coordinate System Definition

These commands are used to define the base coordinate system.

Manipulators have the base coordinate system which cannot be modified. This coordinate system is called “robot coordinate system”. In contrast, the base coordinate system which can change its origin coordinate and be the basis of general local coordinate systems is called “base coordinate system”.

By specifying the origin and the rotation angle of the base coordinate system in relation to the robot absolute coordinate system, you can define the local coordinate system.

To reset the Base coordinate system to default, set “0” to all coordinates. This will make the base coordinate system the same as the robot absolute coordinate system.

Robot parameter data is stored to the compact flash in the Controller. When you execute the command, the data is written to the compact flash. Frequent writing to the compact flash affects its product life. It is recommended to minimize the execution of the command.

Changing the base coordinate system affects all local definitions

When base coordinates are changed, all local coordinate systems must be re-defined.

Command 850: Define Base Coordinate System

Defines the base coordinate system.

Command Syntax

Parameter 1	bit	Name	Description
	15 14 │ 1 0	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

Parameter 2	bit	Name	Description
	15 14 │ 1 0	*specifiedCoordinate* High-order word	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg High-order side 16 bit.

Parameter 3	bit	Name	Description
	15 14 │ 1 0	*specifiedCoordinate* Low-order word	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Defines the base coordinate system by each coordinate.
If order of coordinates is not proper or another command is executed during the execution of this command, values received at that point will be canceled and an error response will be returned.

4-axis manipulator: set the coordinates X, Y, Z, and U.
6-axis manipulator: set the coordinates X, Y, Z, U, V, and W.

Setting value should be specified as fixed-point data which validates to three decimal places.
Also, if the setting value is a negative number, specify the value in 32-bit two’s complement.

Example

When defining the origin coordinate of the base coordinate system as X=100 mm and Y=100 mm.

Command	Response
0352H 0000H 0001H 86A0H	0352H 0000H 0000H
0352H 0001H 0001H 86A0H	0352H 0000H 0000H
0352H 0002H 0000H 0000H	0352H 0000H 0000H
0352H 0003H 0000H 0000H	0352H 0000H 0000H
0352H 0004H 0000H 0000H	0352H 0000H 0000H
0352H 0005H 0000H 0000H	0352H 0000H 0000H

Command 851: Get Base Coordinate System

Acquires the base coordinate system definition.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 ｜ 1 0	*coordinateSelection*	Specifies the coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

Response Syntax

	bit	Name	Description
Response 1	15 14 ｜ 1 0	*coordinateSelection*	Returns the coordinate. 0＝X 1＝Y 2＝Z 3＝U 4＝V 5＝W

	bit	Name	Description
Response 2	15 14 ｜ 1 0	*specifiedCoordinate* High-order word	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm /U,V,W ＝ deg High-order side 16 bit.

	bit	Name	Description
Response 3	15 14 ｜ 1 0	*specifiedCoordinate* Low-order word	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.

Description

Acquires the base coordinate system definition by each coordinate.
The value will be specified as fixed-point data which validates to three decimal places. Also, if the setting value is a negative number, specify the value in 32-bit two’s complement.

Example

When the origin coordinate of the base coordinate system is defined as X=100 mm and Y=100 mm.
Acquires X and Y axes.

Command	Response
0353H 0000H	0353H 0000H 0001H 86A0H
0353H 0001H	0353H 0001H 0001H 86A0H

10.19 Local Number Settings

These commands are used to set the local number for a point at motion command execution.

By specifying the valid local number using this command, points will function as local coordinates in subsequent motion commands.

Available numbers are from 1 to 15. Specifying “0” disables the setting.

This setting cannot be kept when the Controller’s power is turned off. Default is “0” (invalid).

CAUTION	■ Once the local numbers are specified, all points function as specified local coordinates.

Command 900: Set Local

Specifies the local number to be used.

Command Syntax

	bit	Name	Description
Parameter 1	15	*localNumber*	Specifies the local coordinate number to use. 0 = use no local coordinate 1 – 15 = use the specified local coordinate
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specifies the local number to be used.
Specifying the number other than 0 makes the coordinate function in the specified local coordinate.

Example

When setting the local coordinate number 1.

Command	Response
0384H 0001H	0384H 0000H 0000H

Command 901: Get Local

Acquires the current local setting.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*localNumber*	Returns the local coordinate number to be used. 0 = use no local coordinate 1 – 15 = use the specified local coordinate
	14		
	|		
	1		
	0		

Description

Acquires the current setting.

Example

When the local coordinate number 1 is set.

Command	Response
0385H	0385H 0001H

10.20 Sense Condition Settings

These commands are used to specify and display an input condition that, if satisfied, completes the Jump, Jump3, and Jump3CP in progress by stopping the robot above the target position.

Sense is used to stop approach motion during a Jump, Jump3, and Jump3CP instructions.

Settable condition is ON/OFF of one bit I/O.

Jump with Sense Modifier

Checks if the current Sense condition is satisfied. If satisfied, the Jump instruction completes with the manipulator stopped above the target position. That is, when the Sense Condition is True, the manipulator arm remains just above the target position without executing approach motion. When the Sense condition is False, the manipulator arm completes the full Jump instruction motion through to the target position.

Jump, Jump3, Jump3CP with Sense Modifier

Checks if the current Sense condition is satisfied. If satisfied, the Jump, Jump3, and Jump3CP instructions complete with the manipulator stopped above the target position.

Sense Setting at Main Power On

Default value in this interface is not registered. If the motion command is issued while Sense is specified with being undefined, an error response will be returned and the command will not be executed.

Command 950: Set Sense Condition

Sets the Sense condition.

Command Syntax

	bit	Name	Description
Parameter 1	15	*I/O number (bit)*	Specifies the bit I/O number to be used to the input condition.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*reserved*	Specify “0”
	|		
	2		
	1	*I/O type*	0=I/O 1= memory I/O
	0	*logic*	Specifies the logic to satisfy the condition 0 or 1

Response Syntax

Refer to *7. Response code.*

Description

Specifies the bit I/O number to be used for the input condition and the logic to meet the condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
03B6H 0064H 0001H	03B6H 0000H 0000H

Command 951: Get Sense Condition

Acquires the Sense condition.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*registrationStatus*	Returns the registration status. 0 = not registered 1 = registered
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 2	15	*I/O number (bit)*	Returns the bit I/O number. *Returns “0” when not registered.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 3	15	*reserved*	Returns “0”
	│		
	2		
	1	*I/O type*	0 = I/O 1 = memory I/O
	0	*logic*	Returns the logic to conclude the condition. 0=OFF (not registered) 1=ON

Description

Acquires the sense condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
03B7H	03B7H 0001H 0064H 0001H

When not registered.

Command	Response
03B7H	03B7H 0000H 0000H 0000H

Command 952: Get Sense Detected

Acquires whether the Sense condition is detected or not.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*status*	0 = not satisfied 1 = satisfied
	14		
	│		
	1		
	0		

Description

Acquires whether the Sense condition is detected when the motion command is executed with the Sense option specified.
This command is available when the Sense condition is set.

Example

When the Sense condition is satisfied.

Command	Response
03B8H	03B8H 0001H

10.21 Find Setting

These commands are used to specify the condition to store the coordinates during motion.

Settable condition is ON/OFF of one bit I/O.

Coordinates can be saved when the condition is satisfied while the Find option is specified at motion command execution.

After the motion command with option, command to acquire the condition satisfaction status is prepared. If the condition is met, the manipulator can move to the coordinate position of the condition satisfaction by setting the saved coordinate to the point by point edit command and executing the motion command.

Example: when executing the PTP motion to P0 by Find specification and move to the saved coordinate in PTP motion.

Move to P0 as Local coordinate 1.
Set the save coordinate to P1.
Specify P1 to the destination and move to the saved point by Go command.

Command		Description
No.	Code	
900	0384H 0001H	Sets the destination of the motion command as Local coordinate 1 by Local number setting command.
1000	03E8H 0000H 0001H	Sets the condition as I/O number = 0, logic = ON by Find condition setting command.
2000	07D0H 8000H 0000H	Specifies the Find option for Go command and move to P0.
1002	03EAH	Acquires the status by an acquisition command for Find condition satisfaction.
1221	04C5H 0001H	Sets the coordinate acquired by the point edit command to P1.
900	0384H 0000H	Disables the option which sets the destination of the motion command as Local coordinate 1 by Local number setting command.
2000	07D0H 0000H 0001H	Executes the motion with P1 as the destination of Go command.

CAUTION

- Coordinates which Find saves are the robot coordinates. To execute the movement to the acquired coordinate, use the robot coordinate.

  *Execute the motion command with Local number setting (900) disabled.

Command 1000: Set Find Condition

Specifies the condition to store coordinates during motion.

Command Syntax

<table>
<tr><th rowspan="6">Parameter 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">I/O number (bit)</td><td rowspan="5">Specifies the bit I/O number to be used to the input condition.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="5">Parameter 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Specify “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td><td>I/O type</td><td>0 = I/O
1 = memory I/O</td></tr>
</table>

Response Syntax

Refer to *7. Response code*.

Description

Specifies the bit I/O number to be used for the input condition and the logic to meet the condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
03E8H 0064H 0001H	03E8H 0000H 0000H

Command 1001: Get Find Condition

Acquires the condition to store coordinates during motion.

Command Syntax

No parameter.

Response Syntax

<table>
<tr><th rowspan="6">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">registrationStatus</td><td rowspan="5">Returns the registration status.
0 = not registered
1 = registered</td></tr>
<tr><td>14</td></tr>
<tr><td>｜</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">I/O number (bit)</td><td rowspan="5">Returns the bit I/O number.
0 = not registered</td></tr>
<tr><td>14</td></tr>
<tr><td>｜</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 3</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Returns “0”.</td></tr>
<tr><td>｜</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td><td>I/O type</td><td>0 = I/O
1 = memory I/O</td></tr>
<tr><td>0</td><td>logic</td><td>Returns the logic to meet the condition.
0＝OFF
1＝ON</td></tr>
</table>

Description

Acquires the condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
03E9H	03E9H 0001H 0064H 0001H

When not registered.

Command	Response
03E9H	03E9H 0000H 0000H 0000H

Command 1002: Get Find Condition Detected

Acquires the status whether the Find condition is met and the coordinate is saved.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*status*	Returns the status whether the condition is met during the execution of motion command specified by Find and the coordinate is saved. 0 = condition not met 1 = condition is met and the coordinate is saved
	14		
	│		
	1		
	0		

Description

Acquires the status of condition satisfaction during the execution of motion command specified with Find.

Example

When the condition is met and the coordinate is stored.

Command	Response
03EAH	03EAH 0001H

When the condition is not met.

Command	Response
03EAH	03EAH 0000H

10.22 Till Condition Setting

These commands are used to specify and display an input condition that, if satisfied, completes the motion command (Jump, Go, Move, etc.) in progress by decelerating and stopping the robot at an intermediate position.

Settable condition is ON/OFF of one bit I/O.

Command to confirm whether the condition is satisfied after executing the motion command which specified Till option is also provided.

Till Setting at Main Power On

Default value in this interface is not registered. If the motion command is issued while Till is specified with being undefined, an error response will be returned and the command will not be executed.

Command 1050: Set Till Condition

Specifies the condition to terminate the process during the motion command execution.

Command Syntax

Parameter 1	bit	Name	Description
	15	*I/O number (bit)*	Specifies the bit I/O number to be used to the input condition.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*reserved*	Specify “0”.
	|		
	2		
	1	*I/O type*	0= I/O 1= memory I/O
	0	*logic*	Specifies the logic to meet the condition. 0 or 1

Response Syntax

Refer to *7. Response code.*

Description

Specifies the bit I/O number to be used for the input condition and the logic to meet the condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
041AH 0064H 0001H	041AH 0000H 0000H

Command 1051: Get Till Condition

Acquires the condition to terminate the process during the motion command execution.

Command Syntax

No parameter.

Response Syntax

<table>
<tr><th rowspan="6">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">registrationStatus</td><td rowspan="5">Returns the registration status.
0 = not registered
1 = registered</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">I/O number (bit)</td><td rowspan="5">Returns the bit I/O number.
0 = not registered</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 3</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Returns “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td><td>I/O type</td><td>0= I/O
1= memory I/O</td></tr>
<tr><td>0</td><td>logic</td><td>Returns the logic to conclude the condition.
0＝OFF
1＝ON</td></tr>
</table>

Description

Acquires the condition.

Example

When setting the condition under the timing when port number 100 turns ON.

Command	Response
041BH	041BH 0001H 0064H 0001H

When not registered.

Command	Response
041BH	041BH 0000H 0000H 0000H

Command 1052: Get Till Condition Detected

Acquires the status of condition detection during the motion command executed by Till.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*status*	Returns the status whether the condition is met during the motion command executed which specified Till. 0 = condition not met 1 = condition met
	14		
	|		
	1		
	0		

Description

Acquires the status of condition satisfaction during the execution of motion command executed by Till.

Example

When the condition is met and the coordinate is stored.

Command	Response
041CH	041CH 0001H

When the condition is not met.

Command	Response
041CH	041CH 0000H

10.23 CP Control

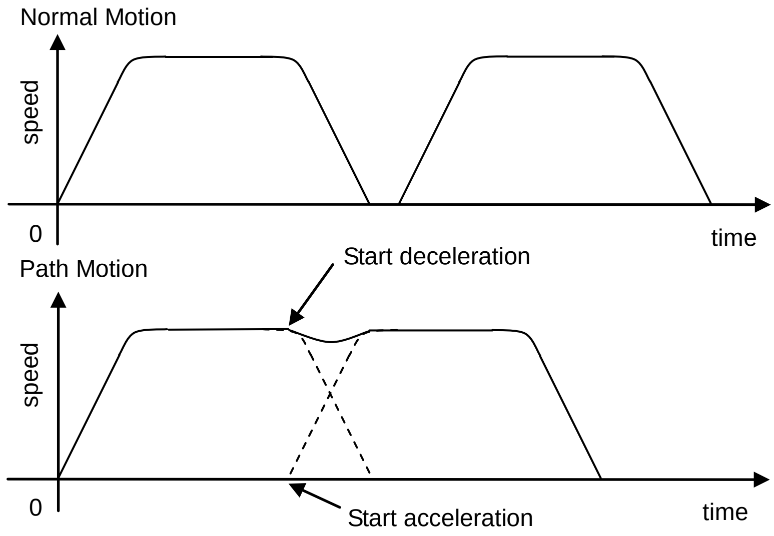
These commands are used to set CP (Continuous Path) motion.

CP (Continuous Path) motion can be used for the following commands:

Arc, Arc3, Go, Jump, Jump3, Jump3CP, Move

When CP is On, each motion command executes the next statement as deceleration starts. Continuous path motion will continue regardless of whether the CP parameter is specified in each motion command or not.

When CP is Off, this function is active only when the CP parameter is specified in each motion command.



When CP is On, path motion will continue without full deceleration between two CP motion (Arc, Arc3, Jump3, Jump3CP, Move), or two PTP motion (Go, Jump). In contrast, full deceleration will occur between a CP motion and a PTP motion.

In contrast, full deceleration will occur between a CP motion and a PTP motion.

Controller startup
Reset
Reset
Switching the Auto / Programming operation mode
Motor On
Motor On

Command 1100: Set CP

Sets CP (Continuous Path) motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*control*	Specify whether to enable or disable the path motion. 1 = enable 0 = disable
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specify whether to enable or disable the path motion.

Example

When enabling the path motion.

Command	Response
044CH 0001H	044CH 0000H 0000H

Command 1101: Get CP

Acquires the current CP motion setting.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*status*	Returns whether the path motion is enabled or disabled. 1 = enabled 0 = disabled
	14		
	|		
	1		
	0		

Description

Acquires the current path motion setting.

Example

When the path motion is enabled.

Command	Response
044DH	044DH 0001H

10.24 Power Control

These commands are used to switch Power Mode to high or low and display the current status.

Low: When Power is set to Low, Low Power Mode is On. This means that the manipulator runs slow (below 250 mm/sec) and the servo stiffness is set light so as to remove servo power if the robot bumps into an object.

High: When Power is set to High, Low Power Mode is Off. This means that the manipulator runs at full speed with the full servo stiffness.

The following operations switch the mode to low power mode. In this case, speed and acceleration settings will be limited to default values. For details of the default values, refer to the specification in each manipulator manual.
Also refer to the *User's Guide 2. Safety*.

Conditions to Cause Power Low

Controller's power is turned ON
Motor On is executed
SFree, SLock, and Brake are executed
Reset and Reset Error are executed
All tasks are aborted by STOP button or Quit All.

Values Limited

Speed
Accel
SpeedS
AccelS

Low Power Mode (Power Low) and Its Effect on Max Speed:

In low power mode, motor power is limited, and effective motion speed setting is lower than the default value. If a higher speed is specified from the Command window (directly) or in a program in Low Power mode, the speed is set to the default value. If a higher speed motion is required, set Power High.

High Power Mode (Power High) and Its Effect on Max Speed:

In high power mode, higher speeds than the default value can be set.

Command 1150: Set Power Mode

Sets the power mode.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*control*	Sets the power mode to High or Low. 1 = Power High 0 = Power Low

Response Syntax

Refer to *7. Response code*.

Description

Sets the power mode to High or Low.

Example

When setting the power mode to High.

Command	Response
047EH 0001H	047EH 0000H 0000H

Command 1151: Get Power Mode Status

Acquires the status of power mode.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*control*	Returns the current status. 1 = Power High 0 = Power Low
	14		
	│		
	1		
	0		

Description

Returns the current status of power mode.

Example

When the power mode is High.

Command	Response
047FH	047FH 0001H

10.25 Point Editing

Edits the specified point (coordinate, flag) or acquires the status of the point (coordinate, flag).

The commands can edit the points as follows:

Sets the current manipulator position to the specified point.

Offsets the coordinate value of the specified point.

Sets the coordinate value to the specified point.

Sets the point to the specified point

Sets and acquires the Hand orientation of the specified point.

Sets and acquires the Elbow orientation of the specified point.

Sets and acquires the Wrist orientation of the specified point.

Sets and acquires j4flag of the specified point.

Sets and acquires j6flag of the specified point.

Sets and acquires the Local number of the specified point.

Sets the coordinate stored by Find to the specified point.

Acquires the coordinate of the specified point.

Command 1200: Set Current Point Number

Sets the current manipulator position to the point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the current manipulator position to the specified point.

Example

When setting the current position to P1.

Command	Response
04B0H 0001H	04B0H 0000H 0000H

Command 1201: Set Two Point Coordinates Offsets

Offsets the specified coordinate value and sets it to the coordinate of specified axes.
Specifies two axes.

Command Syntax

Parameter 1	bit	Name	Description
	15 14 │ 1 0	*pointNumber*	Specifies the point number.

Parameter 2	bit	Name	Description
	15	*toolOffset*	Specify whether to offset in tool offset or not. 0: Normal 1: Tool offset
	14 │ 7	*reserved*	Specify “0”.
	6 5 4	*coordinate 2 specification*	Specifies the second coordinate axis. 0: X axis 1: Y axis 2: Z axis 3: U axis 4: V axis 5: W axis
	3	*reserved*	Specify “0”.
	2 1 0	*coordinate 1 specification*	Specifies the first coordinate axis. 0: X axis 1: Y axis 2: Z axis 3: U axis 4: V axis 5: W axis

Parameter 3	bit	Name	Description
	15 14 │ 1 0	*coordinate1* High-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg High-order side 16 bit.

Parameter 4	bit	Name	Description
	15	*coordinate1* Low-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 5	bit	Name	Description
	15	*coordinate2* High-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg High-order side 16 bit.
	14		
	|		
	1		
	0		

Parameter 6	bit	Name	Description
	15	*coordinate2* Low-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Offsets the specified coordinate value and sets it to the coordinate of specified axes.
Setting value should be specified as fixed-point data which validates to three decimal places.
Also, if the setting value is a negative number, specify the value in 32-bit two's complement.
When tool offset is selected in the 15th bit of Parameter 2, offsetting will be done by tool coordinate system.

Example

When offsetting 20 mm in X-axis direction and -100.003 mm in Y-axis direction for P1.
Specify X axis direction to Coordinate 1 and Y axis to Coordinate 2.

Command	Response
04B1H 0010H 0000H 4E20H FFFEH 795DH	04B1H 0000H 0000H

When offsetting 20 mm in X-axis direction and -100.003 mm in Y-axis direction by tool offset.
Specify X axis direction to Coordinate 1 and Y axis to Coordinate 2.

Command	Response
04B1H 8010H 0000H 4E20H FFFEH 795DH	04B1H 0000H 0000H

*Select tool offset in the 15th bit of Parameter 2.

Command 1202: Set One Point Coordinate Offset

Offsets the specified coordinate value and sets it to the coordinate of specified axes.
Specifies one axis.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*toolOffset*	Specifies whether to offset in tool offset or not. 0: Normal 1: Tool offset
	14	*reserved*	Specify “0”.
	|		
	3		
	2	*coordinate1*	Specifies the first coordinate axis. 0: X axis 1: Y axis 2: Z axis 3: U axis 4: V axis 5: W axis
	1		
	0		

	bit	Name	Description
Parameter 3	15	*coordinate1* High-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*coordinate1* Low-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Offsets the specified coordinate value and sets it to the coordinate of specified axes.
Setting value should be specified as fixed-point data which validates to three decimal places. Also, if the setting value is a negative number, specify the value in 32-bit two's complement.
When tool offset is selected in the 15th bit of Parameter 2, offsetting will be done by tool coordinate system.

Example

When offsetting 20 mm in X-axis direction for P1.

Command	Response
04B2H 0000H 0000H 4E20H	04B2H 0000H 0000H

When offsetting -100.003 mm in Y-axis direction for P1 by tool offset.

Command	Response
04B2H 8001H FFFEH 795DH	04B2H 0000H 0000H

*Select tool offset in the 15th bit of Parameter 2.

Command 1203: Set Two Point Coordinates

Sets the specified coordinate value to the coordinate of specified axes.
Specifies two axes.

Command Syntax

<table>
<tr><th rowspan="6">Parameter 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">pointNumber</td><td rowspan="5">Specifies the point number.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="12">Parameter 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Specify “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>7</td></tr>
<tr><td>6</td><td rowspan="3">coordinate2</td><td rowspan="3">Specifies the second coordinate axis.
0: X axis
1: Y axis
2: Z axis
3: U axis
4: V axis
5: W axis</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td><td>reserved</td><td>Specify “0”.</td></tr>
<tr><td>2</td><td rowspan="3">coordinate1</td><td rowspan="3">Specifies the first coordinate axis.
0: X axis
1: Y axis
2: Z axis
3: U axis
4: V axis
5: W axis</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Parameter 3</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">coordinate1
High-order side</td><td rowspan="5">Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer.
X,Y,Z = mm / U,V,W ＝ deg
High-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Parameter 4</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">coordinate1
Low-order side</td><td rowspan="5">Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer.
X,Y,Z = mm / U,V,W ＝ deg
Low-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

	bit	Name	Description
Parameter 5	15 14 │ 1 0	*coordinate2* High-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm/ U,V,W ＝ deg High-order side 16 bit.

	bit	Name	Description
Parameter 6	15 14 │ 1 0	*coordinate2* Low-order side	Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W ＝ deg Low-order side 16 bit.

Response Syntax

Refer to *7. Response code.*

Description

Sets the specified coordinate value to the coordinate of specified axes.
Setting value should be specified as fixed-point data which validates to three decimal places. Also, if the setting value is a negative number, specify the value in 32-bit two's complement.

Example

When offsetting 20 mm in X-axis direction and -100.003 mm in Y-axis direction for P1.
Specify X axis direction to Coordinate 1 and Y axis to Coordinate 2.

Command	Response
04B3H 0010H 0000H 4E20H FFFEH 795DH	04B3H 0000H 0000H

Command 1204: Set One Point Coordinate

Sets the specified coordinate value to the coordinate of specified axes.
Specifies one axis.

Command Syntax

<table>
<tr><th rowspan="6">Parameter 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">pointNumber</td><td rowspan="5">Specifies the point number.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="7">Parameter 2</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Specify “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>3</td></tr>
<tr><td>2</td><td rowspan="3">coordinate1</td><td rowspan="3">Specifies the first coordinate axis.
0: X axis
1: Y axis
2: Z axis
3: U axis
4: V axis
5: W axis</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Parameter 3</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">coordinate1
High-order side</td><td rowspan="5">Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer.
X,Y,Z = mm / U,V,W ＝ deg
High-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Parameter 4</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">coordinate1
Low-order side</td><td rowspan="5">Specifies the coordinate value (real number) as the value × 1000 converted to a 32-bit integer.
X,Y,Z = mm / U,V,W ＝ deg
Low-order side 16 bit.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

Response Syntax

Refer to 7. Response code.

Description

Sets the specified coordinate value to the coordinate of specified axes.
Setting value should be specified as fixed-point data which validates to three decimal places. Also, if the setting value is a negative number, specify the value in 32-bit two's complement.

Example

When setting -100.003 mm to Y axis of P1.

Command	Response
04B4H 0001H FFFEH 795DH	04B4H 0000H 0000H

Command 1205: Copy Point

Copies the specified point to the other point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*copyDestination pointNumber*	Specifies the point number of the copy destination.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*copySource pointNumber*	Specifies the point number of the copy source.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Copies the content of the point specified in Parameter 2 to the point specified in Parameter 1.
This command can be used to copy the point as a point for work piece and specify the offset point as the destination of the motion command while keeping the copy source point data.

Example

When copying Point 2 to Point 1.

Command	Response
04B5H 0001H 0002H	04B5H 0000H 0000H

Command 1206: Set Hand To Righty

Sets the hand orientation of the specified point to Righty.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the hand orientation of the specified point to Righty.

Example

When setting the hand orientation of the specified point to Righty.

Command	Response
04B6H 000AH	04B6H 0000H 0000H

Command 1207: Set Hand To Lefty

Sets the hand orientation of the specified point to Lefty.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the hand orientation of the specified point to Lefty.

Example

When setting the hand orientation of P10 to Righty.

Command	Response
04B7H 000AH	04B7H 0000H 0000H

Command 1208: Set Elbow To Above

Sets the elbow orientation of the specified point to ABOVE.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the elbow orientation of the specified point to ABOVE.

Example

When setting the elbow orientation of P10 to ABOVE.

Command	Response
04B8H 000AH	04B8H 0000H 0000H

Command 1209: Set Elbow To Below

Sets the elbow orientation of the specified point to BELOW.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the elbow orientation of the specified point to BELOW.

Example

When setting the elbow orientation of P10 to BELOW.

Command	Response
04B9H 000AH	04B8H 0000H 0000H

Command 1210: Set Wrist To Flip

Sets the wrist orientation of the specified point to FLIP.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

Refer to 7. Response code.

Description

Sets the wrist orientation of the specified point to FLIP.

Example

When setting the wrist orientation of P10 to FLIP.

Command	Response
04BAH 000AH	04BAH 0000H 0000H

Command 1211: Set Wrist To NoFlip

Sets the wrist orientation of the specified point to NOFLIP.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the wrist orientation of the specified point to NOFLIP.

Example

When setting the wrist orientation of P10 to NOFLIP.

Command	Response
04BBH 000AH	04BBH 0000H 0000H

Command 1212: Set J4Flag

Specifies j4flag of the specified point.

Command Syntax

Parameter 1	bit	Name	Description
	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*flagValue*	Specifies the flag value. 0: J4F0 1: J4F1
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specifies j4flag of the specified point.

Example

When setting J4F1 to P10.

Command	Response
04BCH 000AH 0001H	04BCH 0000H 0000H

Command 1213: Set J6Flag

Specifies j6flag of the specified point.

Command Syntax

Parameter 1	bit	Name	Description
	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*flagValue*	Specifies the flag value. 0: J6F0 | 127: J6F127
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specifies j6flag of the specified point.

Example

When setting J6F127 to P10.

Command	Response
04BDH 000AH 007FH	04BDH 0000H 0000H

Command 1214: Set Point Local

Specifies the Local number to the specified point.

Command Syntax

Parameter 1	bit	Name	Description
	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*localCoordinate conversion*	Specify whether to convert the local coordinate. 0 = not convert 1 = convert
	14	*reserved*	Specify “0”.
	|		
	4		
	3	*localNumber*	Specifies the local number by a value from 1 to 15.
	2		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Sets the Local number to the specified point.

Difference by specifying the local coordinate conversion.

Selecting “not convert”: the coordinate becomes local.

Selecting “convert”: the coordinate will be converted to the local coordinate.

Example

When setting the local number 15 without converting P1 to the local coordinate.

Command	Response
04BEH 0001H 000FH	04BEH 0000H 0000H

Command 1215: Get Hand

Acquires the hand orientation of the specified point

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	｜		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*handOrientation*	Returns the current hand orientation. 0＝Lefty 1＝Righty
	14		
	｜		
	1		
	0		

Description

Acquires the hand orientation of the specified point.

Example

When the hand orientation of P0 is Righty.

Command	Response
04BFH 0000H	04BFH 0001H

Command 1216: Get Elbow

Acquires the elbow orientation of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*elbowOrientation*	Returns the current elbow orientation. 0＝Above 1＝Below
	14		
	|		
	1		
	0		

Description

Acquires the elbow orientation of the specified point.

Example

When the elbow orientation of P0 is Below.

Command	Response
04C0H 0000H	04C0H 0001H

Command 1217: Get Wrist

Acquires the wrist orientation of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*wristOrientation*	Returns the current wrist orientation. 0＝NoFlip 1＝Flip
	14		
	|		
	1		
	0		

Description

Acquires the wrist orientation of the specified point.

Example

When the wrist orientation of P0 is Flip.

Command	Response
04C1H 0000H	04C1H 0001H

Command 1218: Get J4Flag

Acquires the j4flag value of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	│		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*j4flag*	Returns the setting value of j4flag. 0＝J4F0 1＝J4F1
	14		
	│		
	1		
	0		

Description

Acquires the j4flag value of the specified point.

Example

When j4flag is J4F1.

Command	Response
04C2H 0000H	04C2H 0001H

Command 1219: Get J6Flag

Acquires the j6flag value of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*j6flag*	Returns the setting value of j6flag. 0 ＝ J6F0 | 127＝J6F127
	14		
	|		
	1		
	0		

Description

Acquires the j6flag value of the specified point.

Example

When j6flag is J6F127.

Command	Response
04C3H 0000H	04C3H 007FH

Command 1220: Get Point Local

Acquires the local number of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	｜		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*localNumber*	Returns the local number. * "0" means that the local number is not set.
	14		
	｜		
	1		
	0		

Description

Acquires the local number of the specified point.

Example

When the local number 15 is set to P0.

Command	Response
04C4H 0000H	04C4H 000FH

Command 1221: Copy Find To Point

Sets the coordinate recorded by Find to the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is available only when the condition is satisfied in the execution of command which specified Find. Check the status of condition by command No. 1002 and execute this command only when the condition is satisfied.

*Refer to the Description of Find Condition Setting.

Example

When setting the coordinate to P100.

Command	Response
04C5H 0064H	04C5H 0000H 0000H

Command 1222: Get Point Coordinate

Acquires the coordinate of the specified point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axisSelection*	Select the axis to acquire the coordinate. 0: X axis 1: Y axis 2: Z axis 3: U axis 4: V axis 5: W axis
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*coordinate* High-order side	Returns the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z =mm / U,V,W = deg High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*coordinate* Low-order side	Returns the coordinate value (real number) as the value × 1000 converted to a 32-bit integer. X,Y,Z = mm / U,V,W = deg Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Acquires the coordinate of the specified point.

The value will be returned as fixed-point data which validates to three decimal places. Also, if the setting value is a negative number, the value is returned in 32-bit two’s complement.

Example

When acquring the Y axis when P1 is X: 0.000 Y: 495.336 Z: 246.281 U: 90.000

Command	Response
04C6H 0001H	04C6H 0007H 8EE8H

Command 1223: Set J1Flag

Sets the J1flag attribute for the specified point.
Available for 6-axis manipulators.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*attribute*	0 (/J1F0) J1 range : from -90 to +270 (unit: degree) 1 (/J1F1) J1 range : from -270 to -90, or from +270 to +450 (unit: degree)
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code*.

Description

J1Flag attribute specifies the value range of the Joint #1 for one point.

Example

When setting /J1F1 to P1.

Command	Response
04C7H 0001H 0001H	04C7H 0000H 0000H

Command 1224: Get J1Flag

Acquires the J1flag attribute of the specified point.
Available for 6-axis manipulators.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*attribute*	0 (/J1F0) 1 (/J1F1)
	14		
	|		
	1		
	0		

Example

Acquires the attribute of P1.

When the attribute is set to 1 (/J1F1):

Command	Response
04C8H 0001H	04C8H 0001H

Command 1225: Set J2Flag

Specifies the J2flag attribute for the specified point.
Available for 6-axis manipulators.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*attribute*	0 (/J2F0) J2 range : from -180 to +180 (unit: degree) 1 (/J2F1) J2 range : from -360 to -180, or from +180 to +360 (unit: degree)
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code*.

Description

J2Flag attribute specifies the value range of the Joint #2 for one point.

Example

When setting /J2F1 to P1.

Command	Response
04C7H 0001H 0001H	04C7H 0000H 0000H

Command 1226: Get J2Flag

Acquires the J2flag attribute for the specified point.
Available for 6-axis manipulators.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specifies the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*attribute*	0 (/J2F0) 1 (/J2F1)
	14		
	|		
	1		
	0		

Example

Acquires the attribute of P1.

When the attribute is set to 1 (/J2F1):

Command	Response
04C8H 0001H	04C8H 0001H

Command 1227: Set J1 Angle Attribute

Specifies J1angle attribute of the point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*option*	0= omit the setting value 1= use the setting value If "0= omit the setting value" is selected, Parameter 3 and 4 are not necessary.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*pointNumber*	Specify the point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*setValue* High-order side	Specify the real value by increasing the value thousandfold and converting it to the 32-bit integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*setValue* Low-order side	Specify the real value by increasing the value thousandfold and converting it to the 32-bit integer.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

J1Angle attribute is only available for RS series manipulators. The attribute specifies the Joint #1 angle in singularity where X coordinate and Y coordinate are both "0".
J1Angle attribute value is not effective in the points without singularity.

Example

When omitting the setting value: Specify the P1 attribute

Command	Response
04CBH 0000H 0001H	04CBH 0000H 0000H

When using the setting value: Specify 2.001 to P1

Command	Response
04CBH 0000H 0001H	04CBH 0000H 0000H

Command 1228: Get J1 Angle Attribute

Acquires J1angle attribute of the point.

Command Syntax

	bit	Name	Description
Parameter 1	15	*pointNumber*	Specify the point number.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*attribute* High-order side	Returns the real value by increasing the value thousandfold and converting it to the 32-bit integer.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*attribute* Low-order side	Returns the real value by increasing the value thousandfold and converting it to the 32-bit integer.
	14		
	|		
	1		
	0		

Description

Returns the real value of Joint #1 angle in singularity where X coordinate and Y coordinate are both “0”.

Example

When the P1 attribute is set to 1.002:

Command	Response
04CCH 0001H	04CCH 0000H 03EAH

10.26 LimZ

These commands are used to determine the default value of the Z joint height for Jump commands.

LimZ determines the maximum Z joint height which the arm move to when using the Jump instruction, wherein the manipulator arm raises on the Z joint, moves in the X-Y plane, then lowers on the Z joint. LimZ is simply a default Z joint value used to define the Z joint ceiling position for use during motion caused by the Jump instruction. When a specific LimZ value is not specified in the Jump instruction, the last LimZ setting is used for the Jump instruction.

Resetting LimZ to 0

Restarting the controller, or executing the SFree, SLock, Motor On commands will initialize LimZ to 0.

LimZ Value is Not Valid for Arm, Tool, or Local Coordinates

LimZ Z joint height limit specification is the Z joint value for the robot coordinate system. It is not the Z joint value for Arm, Tool, or Local coordinates. Therefore, take the necessary precautions when using tools or end effectors with different operating heights.

LimZ does not affect Jump3 and Jump3CP

LimZ has no effect on Jump3 or Jump3CP since the span motion is not necessarily perpendicular to the Z axis of the coordinate system.

Command 1250: Set LimZ

Sets the default value of the Z joint height for Jump commands.

Command Syntax

Parameter 1	bit	Name	Description
	15	*height* High-order side	Specifies the coordinate value (mm / real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

Parameter 2	bit	Name	Description
	15	*height* Low-order side	Specifies the coordinate value (mm / real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Determines the default value of the Z joint height for Jump commands.

For the setting value, specify the coordinate value which is in the motion range of Joint #3.

Setting value should be specified as fixed-point data which validates to three decimal places.
Also, if the setting value is a negative number, specify the value in 32-bit two’s complement.

Example

When setting –100 mm for the default value.

Command	Response
04E2H 0001H 86A0H	04E2H 0000H 0000H

Command 1251: Get LimZ

Acquires the default value of the Z joint height for Jump commands.

Command Syntax

No parameter

Response Syntax

	bit	Name	Description
Response 1	15	*height* High-order side	Returns the coordinate value (mm / real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	│		
	1		
	0		

	bit	Name	Description
Response 2	15	*height* Low-order side	Returns the coordinate value (mm / real number) as the value× 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	│		
	1		
	0		

Description

Acquires the default value of the Z joint height for Jump commands.

Setting value should be specified as fixed-point data which validates to three decimal places.
Also, if the setting value is a negative number, specify the value in 32-bit two’s complement.

Example

When –100 mm is set for the default value.

Command	Response
04E3H	04E3H 0001H 86A0H

10.27 Parallel Processing List

Parallel processing allows you to control the specified bit ports by the specified logic in parallel with the execution of motion command, according to progress rate of command execution.

The number of process which can be specified during one motion command is up to 16. This function instructs the list of up to 16 processes to the motion command and executes the parallel processing during the execution of the command. There are 16 lists provided. The user needs to set which of the previously-configured lists to use before executing the motion command.

Example:

Progress rate 50% Bit port number 512 ON

Progress rate 100% Bit port number 512 OFF

By registering the above process to the list and executing the motion command with the “parallel processing available” is specified, ON will be output to the bit port number 512 at 50% of the moving distance and OFF will be output at 100%.

CAUTION	■ Registration of processes to the list should be done in ascending order of progress.

Command 1300: Set Parallel Processing Parameters

Registers parameters in the parallel processing list.

Command Syntax

	bit	Name	Description
Parameter 1	15 14 │ 1 0	*listNumber*	Specifies the list number to register processes by an integer from 0 to 15. Processes are registered to the list of the specified number.

	bit	Name	Description
Parameter 2	15 14 │ 1 0	*progressRate*	Specify progress rate of the motion by an integer from 0 to 100.

	bit	Name	Description
Parameter 3	15 14 │ 1 0	*portNumber* (bit)	Specifies the bit port number to control.

	bit	Name	Description
Parameter 4	15 14 │ 1 0	*logic*	Specifies the logic to control. 0= OFF 1=ON

Response Syntax

Refer to *7. Response code.*

Description

Registers the processing conditions to the specified list by each progress rate.

Registration of processes to the list should be done in ascending order of progress rates.

If the number of registrations has already reached 16, an error response will be returned.

Example

When registering the following processes to the list 5.

Progress rate 50% Bit port number 512 ON
Progress rate 100% Bit port number 512 OFF

Command	Response
0514H 0005H 0032H 0200H 0001H	0514H 0000H 0000H
0514H 0005H 0064H 0200H 0000H	0514H 0000H 0000H

Command 1301: Get Parallel Processing Parameters

Acquires parameters from the parallel processing list.

Command Syntax

	bit	Name	Description
Parameter 1	15	*listNumber*	Specifies the list number to acquire the contents by an integer from 0 to 15.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*type*	Specify whether to acquire the contents from the top of the list. 0 = continue 1 = start from the top
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*position*	Returns the position in the list.
	|		
	8		
	7	*number of registration*	Returns the number of processes registered in the list.
	|		
	0		

	bit	Name	Description
Response 2	15	*progressRate*	Returns the progress rate.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 3	15	*bitNumber*	Returns the port number.
	14		
	|		
	1		
	0		

Response 4	bit	Name	Description
	15	*logic*	Returns the control logic. 0= OFF 1=ON
	14		
	│		
	1		
	0		

Description

Acquires the processing conditions registered in the specified list.

To start the acquisition, specify “start from the top (1)” in *type* of Parameter 2 before executing the first command. For subsequent commands, specify “continue (2)”.

The final determination is done by receiving the same response for *number of registration* and *position* in Response 1.

If no process is registered, *number of registration* (0) and *position* (0) will be returned to Response 1. In this case, values are indeterminate after Response 2. Do not use the syntaxes.

After receiving the last response, it will be returned when request is send again continuously.

Example

When registering the following processes to the list 5.

Progress rate 50% Bit port number 512 ON
Progress rate 100% Bit port number 512 OFF

Command	Response
0515H 0005H 0001H	0515H 0102H 0032H 0200H 0001H
0515H 0005H 0000H	0515H 0202H 0064H 0200H 0000H

Command 1302: Initialize Parallel Processing List

Initializes the specified parallel processing list.

Command Syntax

	bit	Name	Description
Parameter1	15	*listNumber*	Specifies the list number to initialize by an integer from 0 to 15.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Initializes the specified parallel processing list.

The number of registration will be 0.

Example

When initializing the list 15.

Command	Response
0516H 000FH	0516H 0000H 0000H

Command 1303: Configure Parallel Processing List

Configures the parallel processing list.

Command Syntax

	bit	Name	Description
Parameter1	15	*listNumber*	Specifies the list number by an integer from 0 to 15.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Specifies the parallel processing list to be used at the motion command execution.

If the list with no registration is specified, an error will occur at the command execution.

Example

When setting the list 15.

Command	Response
0517H 000FH	0517H 0000H 0000H

Command 1304: Get Current Parallel Processing List

Acquires the current setting of the parallel processing list.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response1	15	*listNumber*	Specifies the list number by an integer from 0 to 15.
	14		
	|		
	1		
	0		

Description

Acquires the current setting of the parallel processing list to be used during motion command execution.

The list 0 is set as default.

Example

Acquires the status of the list 15.

Command	Response
0518H	0518H 000FH

10.28 Singularity Avoidance

Command 1350: Set AutoLJM

Sets Auto LJM (Least Joint Motion).

Command Syntax

	bit	Name	Description
Parameter 1	15	*setting*	0: Auto LJM disabled (default) 1: Auto LJM enabled
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

AutoLJM is available for following commands.
Arc, Arc3, Go, Jump3, Jump3CP, Move
When AutoLJM is On, the manipulator operates with a least joint motion, just like using the LJM function, whether the LJM function is applied to the position data to be passed to each command or not.
If the Auto LJM is enabled, the function will be applied to all commands until it is disabled.

In any of the following cases, AutoLJM has the setting specified in the controller settings (factory default: Off).

Controller startup
Reset
All task stop
Motor On
Switching the Auto / Programming operation mode

AutoLJM Usage Precaution

You can set the AutoLJM function to be enabled at the controller startup by setting the controller preferences. However, if Auto LJM is enabled at all times by controller preferences or commands, this function automatically adjusts the posture of the manipulator to reduce the motion distance, even when you intended to move the joint widely. Therefore, it is recommended to create a program to apply the LJM function only when necessary by using LJM function or AutoLJM command.

Example

Executes Go command with the Auto LJM enabled.

Command	Response
0540H 0001H	0540H 0000H 0000H ←AutoLJM enabled
07D0H 0000H 0000H	07D0H 0000H 0000H ←Motion command (Go)
0540H 0000H	0540H 0000H 0000H ←AutoLJM disabled

Command 1352: Set Avoid Singularity

Sets the singularity avoidance function.

Command Syntax

	bit	Name	Description
Parameter 1	15	*setting*	1: Enables the singularity avoiding function. 0: Disables the singularity avoiding function.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

This command is available for following commands.
Move, Arc, Arc3

A singularity avoiding function is to prevent acceleration errors when the vertical 6-axis robot approaches to the singularity in CP motion by passing a different trajectory and returning to the original trajectory after passing the singularity. This function is only applicable for the wrist singularity. Since the singularity avoiding function is usually set to “1: Enabled” at the controller startup, it is not necessary to change the setting. If you do not want a singularity avoidance to ensure compatibility with software which does not support the singularity avoiding function, or to avoid a trajectory gap, disable the function.

If the parameter is changed, this function remains enabled until the next controller startup.
At the controller startup, the singularity avoiding function has the setting specified in the controller setting (factory default: 1).

Note

Condition setting of singularity neighborhood

To determine whether the manipulator approaches to the singularity neighborhood, angle of Joint #5 and angular velocity of Joint #4 are used. By default, Joint #5 angle is set to ±5 degree, and Joint #4 angle is set to ±10% with respect to the maximum joint velocity.

Command 1400: Motor Control

Controls the motor.

Command Syntax

	bit	Name	Description
Parameter 1	15	*control*	0 = Motor ON 1 = Motor OFF
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code*.

Description

Controls the motor.

Example

Turns on the motor.

Command	Response
0578H 0001H	0578H 0000H 0000H

Command 1401: Get Motor Status

Acquires the current motor status.

Command Syntax

No parameter.

Response Syntax

<table>
<tr><th rowspan="6">Parameter 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">control</td><td rowspan="5">0 = Motor ON
1 = Motor OFF</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

Description

Acquires the current motor status.

Example

The motor is OFF.

Command	Response
0578H 0001H	0578H 0000H 0000H

Command 1450: Controller Reset

Resets the controller to an initial status.

Command Syntax

No parameter.

Response Syntax

Refer to *7. Response code.*

Description

This command resets the following items:

Emergency Stop Status
Error status
Output Bits (All Outputs, except I/O assigned to remote output, set to Off; User can set EPSON RC+ to turn this feature off)
Current robot Speed, SpeedR, SpeedS (Initialized to default values)
Current robot Accel, AccelR, AccelS (Initialized to default values)
Current robot LimZ parameter (Initialized to 0)
Current robot Fine (Initialized to default values)
Current robot Power Low (Low Power Mode set to On)
Current robot PTPBoost (Initialized to default values)

For servo related errors, Emergency Stop status, and any other conditions requiring a reset, no command other than this one will be accepted. In this case first execute this command, then execute other processing as necessary. For example, after an emergency stop, first verify safe operating conditions, execute Reset, and then execute Motor On.

Critical error state will not be canceled by Reset.
When critical error occurs, turn Off the controller and solve the cause of the error.

Notes

Reset Option Switch

If the "Reset turns off outputs" controller preference is on, then when the **Reset** instruction is issued, all outputs will be turned off. This is important to remember when wiring the system such that turning the outputs off should not cause tooling to drop or similar situations. See [Setup]-[Controller]-[Preferences] in the User's Guide for details.

Example

Command	Response
05DCH	05DCH 0000H 0000H

10.29 Motion Commands

These commands are used to move the Arm to a target position in various ways.

Following is the description common in each command. Note that some functions are not available depending on the commands. Follow the descriptions of each command.

Setting of Target Position

This section describes method of designation for the target position.

Setting by Point Number

This method specifies the target position by a point number. The point should be define beforehand to use this method.

Setting by Pallet

This method specifies the pallet number and position in the defined pallet. There are two methods to specify the position.

A: Specify the divided position directly.

B: Specify by the division coordinate.

The pallet can be defined by dividing the area of P1, P2, and P3 by 3×5 as follows.

```
P2
  13  14  15
  10  11  12
   7   8   9
   4   5   6
   1   2   3
  P1        P2
```

For setting-by-position method, specify the position you want to move to by a number ranging from 0 to 15.

For setting-by-coordinate method, specify the row and line. To move to the position “1”, specify (1, 1). For the position “2”, specify (2, 1). And for the position “8”, specify (2, 3).

Speed / Accel setting

By setting Speed/Accel to the option of the motion command, the command will be executed after setting Speed/Accel only by issuing the motion command. This can save the number of command issues. However, prior registration of Speed/Accel table is necessary.

Command 2000: Go

Moves the Arm from the current position to the specified position using PTP motion.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed/Accel setting and target position specifying method affect the number of parameters. Other options do not affect the number of parameters.
For Parameter 2 and later by options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

	bit	Name	Description
Parameter 1	15 14	Till / Find	Specify Till and Find options 0 = not specify 1 = Till 2 = Find
	13	Parallel processing	Select whether to do parallel processing 0 = No 1 = Yes
	12	CP	Select whether to do a CP motion 0 = No 1 = Yes
	11 10 9 8	Speed / Accel	Select whether to set Speed / Accel before the motion command execution 0 = do not set 1 = set only Speed 2 = set only SpeedS 3 = set only SpeedR 4 = set only Accel 5 = set only AccelS 6= set only AccelR 7 = set Speed and Accel 8 = set SpeedS and AccelS 9 = set SpeedR and AccelR
	7 | 3	Reserved	Specify “0”
	2 1 0	Target position specifying method	Select the specifying method of the target position 0 = Setting by point number 1 = Setting by position in the pallet 2 = Setting by coordinate in the pallet

(2) When selected “Setting by point number” for Target position specifying method and “do not set” in Speed / Accel.

Inthis case, use until Parameter 2.

Command No., Parameter1, Parameter2

	bit	Name	Description
Parameter 2	15	*pointNumber*	Specifies the target position by an point number
	14		
	｜		
	1		
	0		

(3) When selected “Setting by position in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

Inthis case, use until Parameter 3.

Command No., Parameter1, Parameter2, Parameter3

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the point number to be used
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*position*	Specifies the pallet position
	14		
	｜		
	1		
	0		

(4) When selected “Setting by coordinate in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

Inthis case, use until Parameter4.

Command No., Parameter1, Parameter2, Parameter3, Parameter4

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the point number to be used
	14		
	｜		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*row*	Specifies the row in the pallet
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*line*	Specifies the line in the pallet
	14		
	|		
	1		
	0		

(5) When selected "Setting by point number" for Target position specifying method and "set Speed and Accel" in Speed / Accel.

Inthis case, use until Parameter3.

Command No., Parameter1, Parameter2, Parameter3

Parameter 3	bit	Name	Description
	15	*pointNumber*	Specifies the target position by an point number
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*Speed/SpeedS/SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration and deceleration are not set.
	|		
	8		
	7	*Accel/AccelS/AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration and deceleration are not set.
	|		
	0		

(6) When selected "Setting by position in the pallet" for Target position specifying method and "set Speed and Accel" in Speed / Accel.

Inthis case, use until Parameter4.

Command No., Parameter1, Parameter2, Parameter3, Parameter4

Parameter 2	bit	Name	Description
	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*position*	Specifies the position in the pallet
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*Speed/SpeedS/SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration and deceleration are not set.
	|		
	8		
	7	*Accel/AccelS/AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration and deceleration are not set.
	|		
	0		

(7) When selected “Setting by coordinate in the pallet” for Target position specifying method and “set Speed and Accel” in Speed / Accel.

Inthis case, use until Parameter 5.

Command No., Parameter1, Parameter2, Parameter3, Parameter4, Parameter5

Parameter 2	bit	Name	Description
	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*row*	Specifies the row in the pallet
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*line*	Specifies the line in the pallet
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15 | 8	*Speed/SpeedS/SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration and deceleration are not set.
	7 | 0	*Accel/AccelS/AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration and deceleration are not set.

Response Syntax

Refer to *7. Response code.*

Description

Moves the arm from the current position to the specified position using PTP motion.

The Go command moves all manipulator arms simultaneously in PTP motion.

With this function, target position is determined by specifying the point number and specifying the pallet.

The path is not predictable because each joint interpolates between the current point and the target point.
Be careful of the interference with peripherals.

Speed of Go command can be set by Speed command. And Accel command determines acceleration and deceleration.

With CP parameter, the arm can accelerate for the next motion command while the arm starts decelerating to a stop. In this case, the arm is not positioned at the target point.

You can use the parallel processing option to output to I/O during command execution. To use the option, register the condition list by parallel processing list command and select the list to be used during the command execution beforehand.

Notes

Difference between Go and Move

The Move and Go commands each cause the manipulator arm to move. The primary difference between the two instructions is that the Go command causes point to point motion whereas the Move command causes the arm to move in a straight line. The Go command is used when the user is primarily concerned with the orientation of the arm when it arrives on point. The Move command is used when it is important to control the path of the robot arm while it is moving.

Difference between Go and Jump

Jump command and Go command each cause the manipulator arm to move in a point to point type fashion. However, the Jump command has one additional feature. Jump causes the robot end effector to first move up to the LimZ value, then in a horizontal direction until it is above the target point, and then finally down to the target point. This allows Jump to be used to guarantee object avoidance and more importantly to improve cycle times for pick and place motion.

Proper Speed and Acceleration Instructions with Go

The Speed and Accel commands are used to specify the speed and acceleration of the manipulator during motion caused by the Go command. Pay close attention to the fact that the Speed and Accel commands apply to point to point type motion (like that for the Go command) while linear and circular interpolation motion uses the SpeedS and AccelS commands.

Using Go with the Optional Till Modifier

The optional Till modifier allows the user to specify a condition to cause the robot to decelerate to a stop at an intermediate position prior to completing the motion caused by the Go command. If the Till condition is not satisfied, the robot travels to the target position.

Checks if the current Till condition becomes satisfied. If satisfied, this command completes by decelerating and stopping the robot at an intermediate position prior to completing the motion caused by the Go command.

To use Till command, specify the conditions by Till setting commands beforehand.

Using Go with the Optional Find Modifier

The optional Find modifier allows the user to specify a condition to cause the robot to record a position during the motion caused by the Go command.

Checks if the current Find condition is satisfied. If satisfied, the current position is stored in the special point FindPos. By using the point edit command, the user can acquire the coordinate of the desired point. With the acquired point, the user can move the manipulator to the position where the condition is satisfied.

Go Command Always Decelerates to a Stop

The Go command always causes the arm to decelerate to a stop prior to reaching the final destination of the move.

Potential errors

Attempt to Move Outside of Robots Work Envelope.

When using explicit coordinates with the Go instruction, you must make sure that the coordinates defined are within the robots valid work envelope. Any attempt to move the robot outside of the valid work envelope will result in an error

Example

When specifying P1 by point number determination without an option.

Command	Response
07D0H 0000H 0001H	07D0H 0000H 0000H

When specifying Pallet 15 by pallet position determination, with position =10, without an option.

Command	Response
07D0H 0001H 000FH 000AH	07D0H 0000H 0000H

When specifying Pallet 15 by pallet coordinate determination, with row = 1 line = 3, without an option.

Command	Response
07D0H 0002H 000FH 0001H 0003H	07D0H 0000H 0000H

When specifying Pallet 15 by pallet coordinate determination, with row = 1 line = 3, without an option.

Command	Response
07D0H 0100H 0001H 0800H	07D0H 0000H 0000H

Command 2001: Jump

Moves the arm in gate trajectory using PTP motion.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed/Accel setting and target position specifying method affect the number of parameters. Other options do not affect the number of parameters.
For Parameter 2 and later by options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

<table>
<tr><th></th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="20">Parameter 1</td><td>15</td><td rowspan="2">Till / Find / Sense</td><td rowspan="2">Specify Till, Find, and Sense options.
0 = not specify
1 = Till
2 = Find
3 = Sence</td></tr>
<tr><td>14</td></tr>
<tr><td>13</td><td>Parallel processing</td><td>Select whether to use a parallel processing
0 = do not use
1 = use</td></tr>
<tr><td>12</td><td>CP</td><td>Select whether to use a path motion
0 = do not use
1 = use</td></tr>
<tr><td>11</td><td rowspan="4">Speed / Accel</td><td rowspan="4">Specify whether to set Speed / Accel before executing the motion command.
0 = do not set
1 = set only Speed
2 = set only SpeedS
3 = set only SpeedR
4 = set only Accel
5 = set only AccelS
6= set only AccelR
7 = set Speed and Accel
8 = set SpeedS and AccelS
9 = set SpeedR and AccelR</td></tr>
<tr><td>10</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td><td>Reserved</td><td>Specify “0”.</td></tr>
<tr><td>6</td><td rowspan="3">Arch</td><td rowspan="3">When using Arch:
Specify the arch number by an integer from 0 to 6
When not using Arch:
Specify “7”</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td><td>Reserved</td><td>Specify “0”.</td></tr>
<tr><td>2</td><td rowspan="3">Target position specifying method</td><td rowspan="3">Select the specifying method of the target position
0 = Setting by point number
1 = Setting by position in the pallet
2 = Setting by coordinate in the pallet</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

(2) When selected “Setting by point number” for Target position specifying method and “do not set” in Speed / Accel.

Inthis case, use until Parameter 2.

Command No., Parameter1, Parameter2

	bit	Name	Description
Parameter 2	15	*pointNumber*	Specifies the target position by an point number
	14		
	|		
	1		
	0		

(3) When selected “Setting by position in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

In this case, use until Parameter 3.

Command No., Parameter1, Parameter2, Parameter3

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*position*	Specifies the position in the pallet
	14		
	|		
	1		
	0		

(4) When selected “Setting by coordinate in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

In this case, use until Parameter 4.

Command No., Parameter1, Parameter2, Parameter3, Parameter4

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*row*	Specifies the row in the pallet
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*column*	Specifies the column in the pallet
	14		
	|		
	1		
	0		

(5) When selected "Setting by point number" for Target position specifying method and "set Speed and Accel" in Speed / Accel.

In this case, use until Parameter 3.

Command No., Parameter1, Parameter2, Parameter3

Parameter 2	bit	Name	Description
	15	*pointNumber*	Specifies the target position by an point number
	14		
	|		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*Speed/SpeedS/SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if speed is not set.
	|		
	8		
	7	*Accel/AccelS/AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration is not set.
	|		
	0		

(6) When selected "Setting by position in the pallet" for Target position specifying method and "set Speed and Accel" in Speed / Accel.

In this case, use until Parameter 4.

Command No., Parameter1, Parameter2, Parameter3, Parameter4

Parameter 2	bit	Name	Description
	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*position*	Specifies the position in the pallet
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Speed/SpeedS/SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. 0 – 15 *Specify “0” if speed is not set.
	|		
	8		
	7	*Accel/AccelS/AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration is not set.
	|		
	0		

(7) When selected “Setting by coordinate in the pallet” for Target position specifying method and “set Speed and Accel” in Speed / Accel.

In this case, use until Parameter 5.

Command No., Parameter1, Parameter2, Parameter3, Parameter4, Parameter5

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number to be used
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*row*	Specifies the row in the pallet
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*column*	Specifies the column in the pallet
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter5	15 | 8	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. 0 – 15 *Specify “0” if speed is not set.
	7 | 0	*Accel / AccelS / AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration is not set.

Response Syntax

Refer to *7. Response code.*

Description

This command moves the arm from the current position to the specified position with a gate motion (a motion which the arm lifts first, and then moves horizontally and lowers vertically at the end).

Jump command moves the arm from the current position to the target position in “Arch motion”. This command can be considered as a statement which executes three movements at one time. For example, if the arch number is defined, one issue of Jump command executes following three commands.

1) First, only the Joint #3 lifts up to the Z axis height calculated by the Arch number used for the Jump command .
2) Then, while still moving upward in Z-axis direction, the arm moves horizontally towards the target position until it reaches the upper Z Limit defined by LimZ. Then the arm starts lowering in Z-axis direction while continuing each motion of Joint #1, #2 and #4. The arm moves until final X, Y and U axis coordinates are acquires.
3) Jump command is then completed by moving the arm down with only Z-axis motion until the target Z coordinate position is reached.

   The coordinates of *destination* (the target position for the move) must be taught previously before executing the Jump command. Acceleration and deceleration for the Jump is controlled by the Accel command. Speed for the move is controlled by the Speed command.

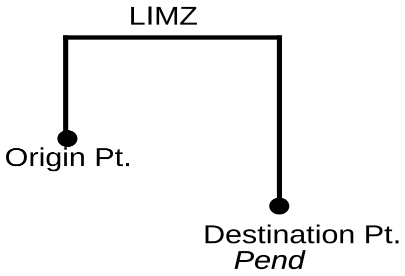
   Jump command cannot be used for horizontal 6-axis manipulators. For those manipulators, use Jump3.

About CP Parameter

When CP parameter is added, acceleration of subsequent command can be overwrapped to deceleration of the prior command. In this case, the arm is not positioned at the target point.

archNumber Details

The arch shape of Jump command can be modified by the *archNumber* value optionally specified in Jump command. This allows the user to define how much the manipulator moves in Z-axis direction before the Joint #1, #2, and #4 move. Valid *archNumber* entries for Jump command are from 0 to 7. The Arch table entries from 0 to 6 are user definable with the Arch command. However, the arch table entry 7 always defines a Gate Motion. Gate Motion is a motion which the manipulator moves only the Joint #3 to the Z-axis coordinate defined by LimZ command before moving Joint #1, #2, and #4. In Gate Motion, Joint #1, #2 and #4 motion begin when the manipulator reaches Z limit defined by LimZ. After the Joint #1, #2, and #4 reach each final destination position, Joint #3 begins moving downward to the final Z-axis coordinate position as defined by *destination* (the target point). Gate Motion looks as follows:



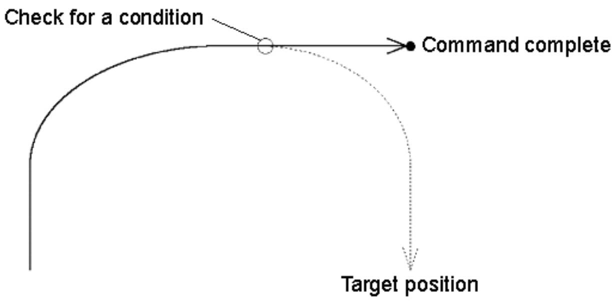
LimZ Details

LimZ *zLimit* specifies the upper Z coordinate value for the horizontal moving plane of the current local coordinate system. The specified arch settings can cause the Joint #1, #2, and #4 to begin movement before reaching LimZ, but LimZ is always the maximum Z height for the move.

The limit value in height direction specified by LimZ is the Z-axis coordinate of local robot coordinate system. It is not the Z-axis coordinate for Arm or Tool coordinates. Therefore, pay enough attention and take necessary measures to use the tools or hands with different work heights.

Sense Details

The Sense optional parameter allows the user to check for an input condition or memory I/O condition before beginning the final Z motion downward. If satisfied, this command completes with the manipulator stopped above the target position where only Z motion is required to reach the target position. It is important to note that the manipulator arm does not stop immediately upon sensing the Sense input modifier.



Command No. 952 can then be used to verify whether the Sense condition was satisfied and the manipulator stopped prior to its target position or that the Sense condition was not satisfied and the manipulator continued until stopping at its target position.

Till Details

The optional Till qualifier allows the user to specify a condition to cause the manipulator to decelerate to stop prior to completing Jump. The user can check if the input is On or Off and cause the arm to decelerate and stop based on the condition specified.

CAUTION	■ **Jump cannot be executed for 6-axis manipulators** Use Jump3 or Jump3CP for 6-axis manipulators.

Omitting *archNumber* Parameter

If the *archNumber* optional parameter is set to "7", manipulator motion will be Gate Motion, as described above.

Difference between Jump and Jump3, Jump3CP

Jump3 and Jump3CP commands can be used for 6-axis manipulators, while Jump command cannot. For SCARA manipulators (including RS series), using Jump command shortens the joint motion time for depart and approach motion. Depart and approach motion in Jump3 can be executed along the Z axis and in other directions.

Difference between Go and Jump

The most important difference is that Go command simply causes Point to Point motion where all joints start and stop at the same time (they are synchronized). Jump is different since it causes vertical Z movement at the beginning and end of the move. Jump is ideal for pick and place type applications.

Decelerating to stop with Jump command

Jump command always causes the arm to decelerate to stop prior to reaching the destination point.

Proper Speed and Acceleration instructions with Jump:

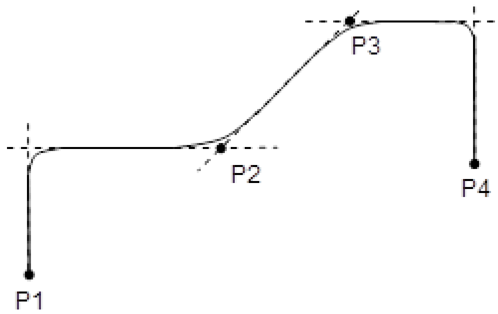
Speed and Accel commands are used to specify the speed and acceleration of the manipulator during Jump motion. Note that Speed and Accel apply to point to point type motion (Go, Jump, etc.). For linear and circular interpolated motion commands such as Move or Arc, use SpeedS and AccelS commands. For Jump command, it is possible to separately specify speeds and accelerations for upward motion of Joint #3, horizontal travel including Joint #4 rotation, and Joint #3 downward motion.

Pass function of Jump

When the CP parameter is specified for Jump with 0 downward motion, the Jump horizontal travel does not decelerate to a stop but goes on smoothly to the next PTP motion.

When the CP parameter is specified for a PTP motion command right before a Jump with 0 upward motion, the PTP motion does not decelerate to a stop but connects smoothly with the Jump horizontal travel.

This is useful when you want to replace the horizontal travel of Jump (a PTP motion) with several PTP motion.



Important concerns for use of Arch

Actual arch motion trajectory cannot be guaranteed since the arch motion is comprised of vertical motion and horizontal motion executed on trajectory control. The trajectory may change depending on motion speed or Arm motion. Check the actual trajectory with actual speed and posture used in operation.

- Even if Jump command with the same arch number is executed at the same position, trajectory in low speed mode goes lower than that of in high speed mode. Therefore, even if collision with obstacle is not seen in high speed mode, the manipulator may hit with obstacle in los speed mode.
- Amount of vertical lift tends to increase and vertical-drop tends to decrease in high speed mode compared to low speed mode. When the fall distance of the trajectory is shorter than expected, lower the speed and/or the deceleration, or set the fall distance larger.
- Even if Jump command with the same distance and speed is executed, the trajectory is affected by motion of the robot arms. As a general example, the vertical upward distance increases and the vertical downward distance decreases for a SCARA robot when the movement of the first arm is large. When the vertical fall distance decreases and the trajectory is shorter than expected, lower the speed and/or the deceleration, or set the fall distance larger.

Potential Errors

LimZ Value Not High Enough

When the current arm position of the Z joint is higher than the value set for LimZ and a Jump instruction is attempted, an Error 4005 will occur .

Example

When specifying P1 by point number determination without an option.

Command	Response
07D1H 0070H 0001H	07D1H 0000H 0000H

When specifying Pallet 15 by pallet position determination, with position =10, without an option.

Command	Response
07D1H 0071H 000FH 000AH	07D1H 0000H 0000H

When specifying Pallet 15 by pallet coordinate determination, with row = 1 line = 3, without an option.

Command	Response
07D1H 0072H 000FH 0001H 0003H	07D1H 0000H 0000H

When specifying P1 by point number determination, with Speed/Accel=set Speed only, Table number =8.

Command	Response
07D1H 0170H 0001H 0800H	07D1H 0000H 0000H

Command 2002: JumpCP

Moves the manipulator in 3D gate trajectory, using a combination of two CP motions and one PTP motion.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed/Accel setting affects the number of parameters. Other options do not affect the number of parameters.
For Parameter 2 and later by options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

<table>
<tr><th rowspan="18">Parameter 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="2">Till / Find / Sense</td><td rowspan="2">Specify Till, Find, and Sense options.
0 = not specify
1=Till
2=Find
3=Sence</td></tr>
<tr><td>14</td></tr>
<tr><td>13</td><td>Parallel processing</td><td>Select whether to use a parallel processing
0 = do not use
1 = use</td></tr>
<tr><td>12</td><td>CP</td><td>Select whether to use a path motion
0 = do not use
1 = use</td></tr>
<tr><td>11</td><td rowspan="4">Speed / Accel</td><td rowspan="4">Specify whether to set Speed/Accel before executing the motion command.
0 = do not set
1 = set only Speed
2 = set only SpeedS
3 = set only SpeedR
4 = set only Accel
5 = set only AccelS
6= set only AccelR
7 = set Speed and Accel
8 = set SpeedS and AccelS
9 = set SpeedR and AccelR</td></tr>
<tr><td>10</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td><td>Reserved</td><td>Specify “0”.</td></tr>
<tr><td>6</td><td rowspan="3">Arch</td><td rowspan="3">When using Arch:
Specify the arch number by an integer from 0 to 6
When not using Arch:
Specify “7”</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td><td>Reserved</td><td>Specify “0”.</td></tr>
</table>

(2) No Speed / Accel setting

In this case, use until Parameter 4.

Command No., Parameter1, Parameter2, Parameter3, Parameter4

Parameter 2	bit	Name	Description
	15	*pointNumber*	Specifies the target position by an point number
	14		
	⏐		
	1		
	0		

Parameter 3	bit	Name	Description
	15	*approachStart*	Specifies the approach start point over the target coordinate by a point number.
	14		
	⏐		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*destination*	Specifies the target coordinate where the manipulator reaches to by a point number.
	14		
	⏐		
	1		
	0		

(3) With Speed / Accel setting

In this case, use until Parameter 5.

Command No., Parameter1, Parameter2, Parameter3, Parameter4, Parameter5

Parameter2	bit	Name	Description
	15	*depart*	Specifies the depart point over the target coordinate by a point number.
	14		
	⏐		
	1		
	0		

Parameter3	bit	Name	Description
	15	*approachStart*	Specifies the approach start point over the target coordinate by a point number.
	14		
	⏐		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*destination*	Specifies the target coordinate where the manipulator reaches to by a point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. 0 – 15 *Specify “0” if speed is not set.
	|		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if speed is not set.
	|		
	0		

Response Syntax

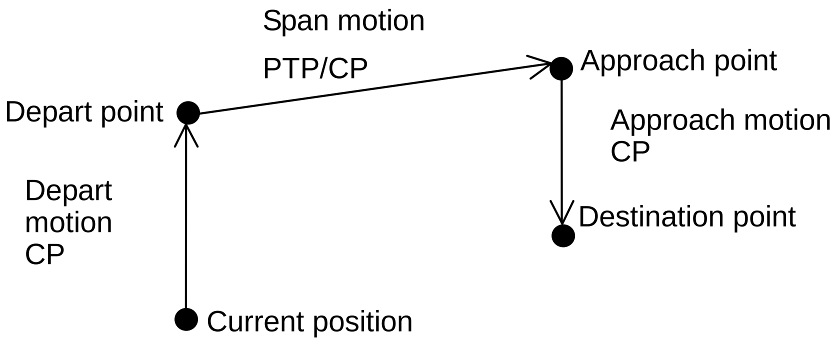
Refer to *7. Response code.*

Description

Moves the arm from the current position to the destination in 3D gate motion.

3D gate motion is a combination of two CP motion and one PTP motion.

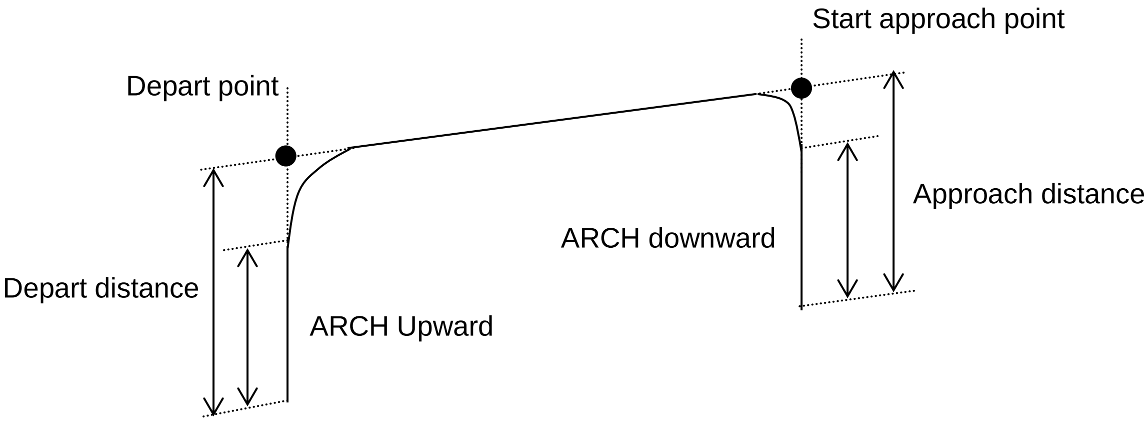
Moves the arm from the current position to the destination point with 3D gate motion. 3D gate motion. Jump3 is a combination of two CP motion and one PTP motion. The depart motion form the current position to the depart point is always CP motion. The span motion from the depart point to the start approach point is PTP motion in Jump3, and the CP motion in Jump3CP. The approach motion from the starting approach point to the target point is always CP motion.



Arch motion is achieved by specifying the arch number.

The arch motion for Jump3, Jump3CP is as shown in the figure below.

For arch motion to occur, the Depart distance must be greater than the arch upward distance and the Approach distance must be greater than the arch downward distance.



Jump3CP uses the SpeedS speed value and AccelS acceleration and deceleration values. Refer to *Using Jump3CP with CP* below on the relation between the speed/acceleration and the acceleration/deceleration. If, however, the ROT modifier parameter is used, Jump3CP uses the SpeedR speed value and AccelR acceleration and deceleration values. In this case SpeedS speed value and AccelS acceleration and deceleration value have no effect.

Usually, when the move distance is 0 and only the tool orientation is changed, an error will occur. However, by using the ROT parameter and giving priority to the acceleration and the deceleration of the tool rotation, it is possible to move without an error. When there is not an orientational change with the ROT modifier parameter and movement distance is not 0, an error will occur.

Also, when the tool rotation is large as compared to move distance, and when the rotation speed exceeds the specified speed of the manipulator, an error will occur. In this case, please reduce the speed or append the ROT modifier parameter to give priority to the rotational speed/acceleration/deceleration.

Notes

LimZ does not affect Jump3 and Jump3CP

LimZ has no effect on Jump3 or Jump3CP since the span motion is not necessarily perpendicular to the Z axis of the coordinate system.

Jump3 span motion is PTP (point to point)

It is difficult to predict Jump3 span motion trajectory. Therefore, be careful that the robot doesn't collide with peripheral equipment and that robot arms don’t collide with the robot.

Using Jump3, Jump3CP with CP

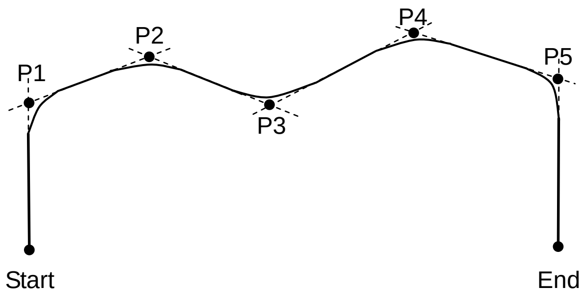
The CP parameter causes the arm to move to *destination* without decelerating or stopping at the point defined by *destination*. This is done to allow the user to string a series of motion instructions together to cause the arm to move along a continuous path while maintaining a specified speed throughout all the motion. The Jump3 and Jump3CP instructions without CP always cause the arm to decelerate to a stop prior to reaching the point *desination*.

Pass function of Jump3

When the CP parameter is specified for Jump3 with 0 approach motion, the Jump3 span motion does not decelerate to a stop but goes on smoothly to the next PTP motion.

When the CP parameter is specified for a PTP motion command right before Jump3 with 0 depart motion, the

PTP motion does not decelerate to stop but connects smoothly with the Jump3 span motion.
This is useful when you want to replace the span motion of Jump3 (a PTP motion) with several PTP motion.



Important concern for Use of Arch

Actual arch motion trajectory cannot be guaranteed since the arch motion is comprised of vertical motion and horizontal motion executed on trajectory control. The trajectory may change depending on motion speed or Arm motion. Check the actual trajectory with actual speed and posture used in operation.

- Even if Jump command with the same arch number is executed at the same position, trajectory in low speed mode becomes lower than that of in high speed mode. Therefore, even if collision with obstacle is not seen in high speed mode, the manipulator may hit with obstacle in low speed mode.
- Vertical lift distance tends to increase and vertical-drop distance tends to decrease in high speed mode compared to low speed mode. When the fall distance of the trajectory is shorter than expected, lower the speed and/or the deceleration, or set the fall distance larger.
- Even if Jump command with the same distance and speed is executed, the trajectory may change due to the motion of the manipulator arms.

Potential errors

When the majority of depart (approach) motion uses the same joint as the span motion

An acceleration error may occur during an arch motion executed by the Jump3 andJump3CP commands. This error occurs frequently when the majority of the motion during depart or approach uses the same joint as the span motion. To avoid this error, reduce the acceleration/deceleration speed of the span motion using Accel command for Jump3 or using AccelS command for Jump3CP.

Depending on the motion and orientation of the manipulator, it may also help to reduce the acceleration and deceleration of the depart motion (approach motion) using the AccelS command.

Example

When the depart coordinate is P1, the approach start coordinate is P2, and the target coordinate is P3 without options.

Command	Response
07D2H 0070H 0001H 0002H 0003H	07D2H 0000H 0000H

Command 2003: Jump3CP

Moves the manipulator in 3D gate trajectory, using three CP motions.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed / Accel setting affects the number of parameters.
Other options do not affect the number of parameters.
For Parameter 2 and later for options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

	bit	Name	Description
Parameter 1	15 14	Till / Find / Sense	Specify Till, Find, and Sense options. 0 =not specify 1=Till 2=Find 3=Sence
	13	Parallel processing	Select whether to use a parallel processing 0 = do not use 1 = use
	12	CP	Select whether to use a path motion 0 = do not use 1 = use
	11 10 9 8	Speed / Accel	Specify whether to set Speed/Accel before executing the motion command. 0 = do not set 1 = set only Speed 2 = set only SpeedS 3 = set only SpeedR 4 = set only Accel 5 = set only AccelS 6= set only AccelR 7 = set Speed and Accel 8 = set SpeedS and AccelS 9 = set SpeedR and AccelR
	7	ROT	Give priority to tool orientation change and determine motion speed and acceleration/deceleration 0 = do not use 1 = use
	6 5 4	Arch	When using Arch: Specify the arch number by an integer from 0 to 6 When not using Arch: Specify "7"
	3 | 0	Reserved	Specify "0".

(2) No Speed / Accel setting

In this case, use until Parameter 4.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4

	bit	Name	Description
Parameter 2	15	*depart*	Specifies the depart start point over the target coordinate by a point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*approachStart*	Specifies the approach start point over the target coordinate by a point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*destination*	Specifies the target coordinate where the manipulator reaches to by a point number.
	14		
	|		
	1		
	0		

(3) With Speed / Accel setting

In this case, use until Parameter 5.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4, Parameter 5

	bit	Name	Description
Parameter 2	15	*depart*	Specifies the depart start point over the target coordinate by a point number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*approachStart*	Specifies the approach start point over the target coordinate by a point number.
	14		
	|		
	1		
	0		

Parameter 4	bit	Name	Description
	15	*destination*	Specifies the target coordinate where the manipulator reaches to by a point number.
	14		
	|		
	1		
	0		

Parameter 5	bit	Name	Description
	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if speed is not set.
	|		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration is not set.
	|		
	0		

Response Syntax

Refer to *7. Response code.*

Description

A combination of 3 CP motion.

Giving priority to tool orientation change by ROT option, the command determines motion speed and acceleration / deceleration.

For more information, refer to Description of Command 2002.

Example

When the depart coordinate is P1, the approach start coordinate is P2, and the target coordinate is P3 without options.

Command	Response
07D3H 0070H 0001H 0002H 0003H	07D3H 0000H 0000H

Command 2005: Move

Moves the manipulator from the current position to a target position using linear interpolated motion.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed / Accel setting and target position specifying method affect the number of parameters. Other options do not affect the number of parameters.

For Parameter 2 and later for options, follow the descriptions (2) and later.

	bit	Name	Description
Parameter 1	15 14	Till / Find	Specify Till and Find options. 0 = not specify 1=Till 2=Find
	13	Parallel processing	Select whether to use a parallel processing 0 = do not use 1 = use
	12	CP	Select whether to use a path motion 0=do not use 1=use
	11 10 9 8	Speed / Accel	Specify whether to set Speed/Accel before executing the motion command. 0 = do not set 1 = set only Speed 2 = set only SpeedS 3 = set only SpeedR 4 = set only Accel 5 = set only AccelS 6= set only AccelR 7 = set Speed and Accel 8 = set SpeedS and AccelS 9 = set SpeedR and AccelR
	7	ROT	Give priority to tool orientation change and determine motion speed and acceleration / deceleration 0 = do not use 1 = use
	6 | 3	Reserved	Specify “0”.
	2 1 0	Target position specifying method	Select the specifying method of the target position 0 = Setting by point number 1 = Setting by position in the pallet 2 = Setting by coordinate in the pallet

(2) When selected “Setting by point number” for Target position specifying method and “do not set” in Speed / Accel.

In this case, use until Parameter 2.

Command No., Parameter 1, Parameter 2

	bit	Name	Description
Parameter 2	15	*pointNumber*	Specifies the target position by an point number
	14		
	|		
	1		
	0		

(3) When selected “Setting by position in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

In this case, use until Parameter 3.

Command No., Parameter 1, Parameter 2, Parameter 3

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*position*	Specifies the position in the pallet.
	14		
	|		
	1		
	0		

(4) When selected “Setting by coordinate in the pallet” for Target position specifying method and “do not set” in Speed / Accel.

In this case, use until Parameter 4.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*row*	Specifies the row in the pallet.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*line*	Specifies the line in the pallet.
	14		
	｜		
	1		
	0		

(5) When selected “Setting by point number” for Target position specifying method and “set Speed and Accel” in Speed / Accel.

In this case, use until Parameter 3.

Command No., Parameter 1, Parameter 2, Parameter 3

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify “0” if speed is not set.
	｜		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the acceleration table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration is not set.
	｜		
	0		

(6) When selected “Setting by position in the pallet” for Target position specifying method and “set Speed and Accel” in Speed / Accel.

In this case, use until Parameter 4.

Command No., Parameter 1, Parameter 2, Parameter 3,Parameter 4

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number.
	14		
	｜		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*position*	Specifies the position in the pallet.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. 0 – 15 0 - 15 *Specify “0” if speed is not set.
	|		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify “0” if acceleration is not set.
	|		
	0		

(7) When selected “Setting by coordinate in the pallet” for Target position specifying method and “set Speed and Accel” in Speed / Accel.

In this case, use until Parameter 5.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4, Parameter 5

	bit	Name	Description
Parameter 2	15	*palletNumber*	Specifies the pallet number.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*row*	Specifies the row in the pallet.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*line*	Specifies the line in the pallet.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 5	15 ｜ 8	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if speed is not set.
	7 ｜ 0	*Accel / AccelS / AccelR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration is not set.

Response Syntax

Refer to *7. Response code.*

Description

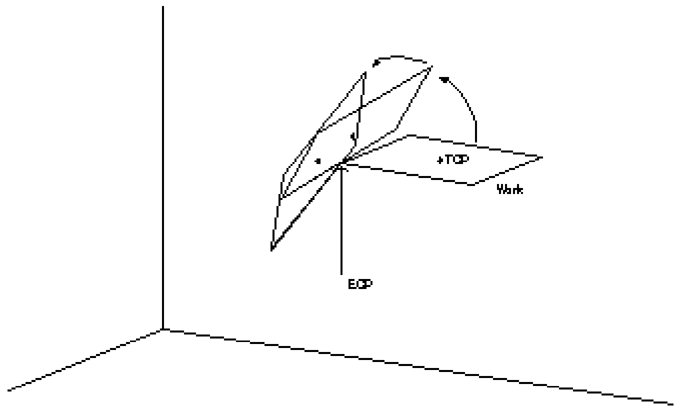
Moves the arm from the current position to *destination* in a straight line.

This command moves all axes to start and stop at the same time. The coordinates of the target position must be taught previously before executing the command. Acceleration and deceleration of the command is controlled by the AccelS command and speed is controlled by the SpeedS command. If the SpeedS value exceeds the allowable speed for any joint, power to all four joint motors will be turned off, and the manipulator will stop.

Move uses the SpeedS speed value and AccelS acceleration and deceleration values. For the relation between the speed/acceleration and the acceleration/deceleration, refer to *Using Move with CP* below. However, if the ROT modifier parameter is used, Move uses the SpeedR speed value and AccelR acceleration and deceleration values. In this case, SpeedS speed value and AccelS acceleration and deceleration value have no effect.

Usually, when the move distance is 0 and only the tool orientation is changed, an error will occur. However, by using the ROT parameter and giving priority to acceleration and deceleration of the tool rotation, the manipulator can be moved without an error. When there is no orientation change and movement distance is not 0 with the ROT modifier parameter, an error will occur.

Also, when the tool rotation is significantly large compared to the moving distance, and when the rotation speed exceeds the specified manipulator speed, an error will occur. In this case, reduce the speed or append the ROT modifier parameter to give priority to the acceleration and deceleration.



The optional Till qualifier allows the user to specify a condition where the manipulator decelerates to stop before completing the Move. The user can check if the input is On or Off and make the arm to stop based on the specified condition . This feature works like an interrupt where the Move is interrupted (stopped) once the Input condition is met. If the input condition is never met during the Move then the arm successfully arrives on the point specified as the target position.

To use the Till qualifier, the condition must be specified by the Till setting command beforehand.

Notes

Move Cannot Execute Range Verification Prior To Motion

Move cannot execute range verification of the trajectory prior to motion. Therefore, even if the target position is within an allowable range, it is possible for the system to find a prohibited position along the way to a target point. In this case, the arm may abruptly stop and cause shock to the servo, resulting in failure. To prevent this, be sure to perform range verifications at low speed prior to using Move at high speeds. In summary, even though the target position is within the range of the arm, the arm cannot reach the position if the trajectory has intermediate points which are out of the physical motion range of the arm.

Using Move with CP

The CP parameter causes the arm to move to the target position without decelerating or stopping at the point. This allows the user to string commands and execute a series of motion commands at a constant speed. The Move instruction without CP always causes the arm to decelerate to stop before reaching the target point.

Proper Speed and Acceleration Instructions with Move

The SpeedS and AccelS commands are used to specify the speed and acceleration of the manipulator during Move motion. Note that SpeedS and AccelS are applied to the linear and circular interpolated motion while point to point motion uses the Speed and Accel instructions.

Potential Errors

Attempt to execute motion with Linear distance is 0

Move command causes errors when you attempt to execute motion to change only U coordinate value of four-degree-of-freedom manipulators (SCARA including RS series) or U, V, and W coordinate values of six-degree-of-freedom manipulators (vertical 6-axis manipulators). In this case, use the ROT parameter.

Joint Overspeed Errors

When even one of the axes exceeds its allowable speed during rmotion, an overspeed error will occur. In this case, the arm stops moving and the motor excitation turns OFF.

Attempt to Pass the Origin Point (RS series)

If RS series manipulator attempts to pass the point near the origin point by Move command, an overspeed error may occur. For commands which pass near the origin point, take the following countermeasures.

- Lower the SpeedS value
- Take a different path to avoid the origin point
- Use PTP motion such as Go command instead of Move command.

Example

When specifying P1 by point number determination without an option.

Command	Response
07D5H 0000H 0001H	07D5H 0000H 0000H

When specifying Pallet 15 and Position 10 by pallet position determination without an option.

Command	Response
07D5H 0001H 000FH 000AH	07D5H 0000H 0000H

When specifying Pallet 15 with row = 1 Line = 3 by pallet coordinate determination without an option.

Command	Response
07D5H 0002H 000FH 0001H 0003H	07D5H 0000H 0000H

When specifying P1 by point number determination, with Speed/Accel= set Speed only, Table number = 8.

Command	Response
07D5H 0100H 0001H 0800H	07D5H 0000H 0000H

Command 2006: Arc

Moves the arm to the specified position using circular interpolation in the XY plane.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed / Accel setting affects the number of parameters.
Other options do not affect the number of parameters.
For Parameter 2 and later for options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

	bit	Name	Description
Parameter 1	15 14	Till / Find	Specify Till and Find options. 0 = not specify 1=Till 2=Find
	13	Parallel processing	Select whether to use a parallel processing 0 = do not use 1 = use
	12	CP	Select whether to use a path motion 0=do not use 1=use
	11 10 9 8	Speed / Accel	Specify whether to set Speed / Accel before executing the motion command. 0 = do not set 1 = set only Speed 2 = set only SpeedS 3 = set only SpeedR 4 = set only Accel 5 = set only AccelS 6= set only AccelR 7 = set Speed and Accel 8 = set SpeedS and AccelS 9 = set SpeedR and AccelR
	7	ROT	Give priority to tool orientation change and determine motion speed and acceleration / deceleration 0 = do not use 1 = use
	6 | 0	Reserved	Specify “0”.

(2) No Speed / Accel setting

In this case, use until Parameter 3.

Command No., Parameter 1, Parameter 2, Parameter 3

	bit	Name	Description
Parameter 2	15	*midPoint*	Specify by a point number. The middle point (taught previously by the user) which the arm travels through on its way from the current point to *endPoint*.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*endPoint*	Specify by a point number. The end point (taught previously by the user) which the arm travels to during the arc type motion. This is the final position at the end of the circular move.
	14		
	|		
	1		
	0		

(3) With Speed / Accel setting

In this case, use until Parameter 4.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4

	bit	Name	Description
Parameter 2	15	*midPoint*	Specify by a point number. The middle point (taught previously by the user) which the arm travels through on its way from the current point to *endPoint*.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*endPoint*	Specify by a point number. The end point (taught previously by the user) which the arm travels to during the arc type motion. This is the final position at the end of the circular move.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if speed is not set.
	|		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration is not set.
	|		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Arc and Arc3 move the arm to the specified point using circular interpolation in the XY plane.

These two commands are available for SCARA (including RS series) and 6-axis manipulators.

Arc and Arc3 are used to move the arm in a circular interpolation motion from the current position to *endPoint* through *midPoint*. The system automatically calculates a trajectory based on three points (current position, *endPoint*, and *midPoint*) and then moves the arm to the point defined by *endPoint* along the trajectory. The coordinates of *midPoint* and *endPoint* must be taught previously before executing the command.

Arc and Arc3 use the SpeedS speed value and AccelS acceleration and deceleration values. For the relation between the speed/acceleration and the acceleration/deceleration, refer to *Using Arc3 with CP* below. However, if the ROT modifier parameter is used, Arc and Arc3 use the SpeedR speed value and AccelR acceleration and deceleration values. In this case, SpeedS speed value and AccelS acceleration and deceleration value have no effect.

Usually, when the move distance is 0 and only the tool orientation is changed, an error occurs. However, by using the ROT parameter and giving priority to the acceleration and the deceleration of the tool rotation, it is possible to move the manipulator without the error. When there is no orientation change and movement distance is not 0 with the ROT modifier parameter, an error will occur.

Setting Speed and Acceleration for Arc motion

SpeedS and AccelS are used to set speed and acceleration for the Arc and Arc3 instructions.
SpeedS and AccelS allow the user to specify a velocity in mm/sec and acceleration in mm/sec2.

Notes

Arc Instruction works in Horizontal Plane Only

The Arc path is a true arc in the Horizontal plane. The path is interpolated using the values for *endPoint* as its basis for Z and U. Use Arc3 for 3 dimensional arcs.

Range verification for Arc command

The Arc and Arc3 statements cannot compute a range verification of the trajectory prior to the arc motion. Therefore, even for target positions that are within an allowable range, en route the robot may attempt to traverse a path which has an invalid range, stopping with a severe shock which may damage the arm. To prevent this from occurring, be sure to perform range verifications by running the program at low speeds prior to running at faster speeds.

Suggested motion to setup for the Arc move

Because the arc motion begins from the current position, it may be necessary to use the Go, Jump or other related motion command to bring the robot to the desired position prior to executing Arc or Arc3.

Using Arc, Arc3 with CP

The CP parameter causes the arm to move to the end point without decelerating or stopping at the point defined by *endPoint*. This allows the user to string commands and execute a series of motion commands at a constant speed. The Arc and Arc3 instructions without CP always make the arm to decelerate to stop before reaching the end point.

Potential Errors

Changing Hand Attributes

Pay attention to the Hand attribute of each point when using the Arc command. If the hand orientation is changed (from Righty to Lefty or vice-versa) during the circular interpolation move, an error will occur. The arm attributes (/L: Lefty, or /R: Righty) must be the same for the current position, *midPoint* and *endPoint* points.

Attempt to move the arm outside the work envelope

If the specified circular motion attempts to move the arm outside the work envelope of the arm, an error will occur.

Example

When the midpoint is P1 and endPoint is P2, without an option.

Command	Response
07D6H 0000H 0001H 0002H	07D6H 0000H 0000H

Command 2007: Arc3

Moves the arm to the specified point using circular interpolation in 3 dimensions.

Command Syntax

(1) Option

Specify the target position specifying method and each option. The number of necessary parameters varies depending on the specified options. Speed / Accel setting affects the number of parameters.
Other options do not affect the number of parameters.
For Parameter 2 and later for options, follow the descriptions (2) and later.

Options are specified by Parameter 1.

	bit	Name	Description
Parameter 1	15 14	Till / Find	Specify Till and Find options. 0 = not specify 1=Till 2=Find
	13	Parallel processing	Select whether to use a parallel processing 0 = do not use 1 = use
	12	CP	Select whether to use a path motion 0=do not use 1=use
	11 10 9 8	Speed / Accel	Specify whether to set Speed / Accel before executing the motion command. 0 = do not set 1 = set only Speed 2 = set only SpeedS 3 = set only SpeedR 4 = set only Accel 5 = set only AccelS 6= set only AccelR 7 = set Speed and Accel 8 = set SpeedS and AccelS 9 = set SpeedR and AccelR
	7	ROT	Give priority to tool orientation change and determine motion speed and acceleration / deceleration 0 = do not use 1 = use
	6 | 0	Reserved	Specify “0”.

(2) No Speed / Accel setting

In this case, use until Parameter 3.

Command No., Parameter 1, Parameter 2, Parameter 3

	bit	Name	Description
Parameter 2	15	*midPoint*	Specify by a point number. The middle point (taught previously by the user) which the arm travels through on its way from the current point to *endPoint*.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*endPoint*	Specify by a point number. The end point (taught previously by the user) which the arm travels to during the arc type motion. This is the final position at the end of the circular move.
	14		
	|		
	1		
	0		

(3) With Speed / Accel setting

In this case, use until Parameter 4.

Command No., Parameter 1, Parameter 2, Parameter 3, Parameter 4

	bit	Name	Description
Parameter 2	15	*midPoint*	Specify by a point number. The middle point (taught previously by the user) which the arm travels through on its way from the current point to *endPoint*.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*endPoint*	Specify by a point number. The end point (taught previously by the user) which the arm travels to during the arc type motion. This is the final position at the end of the circular move.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*Speed / SpeedS / SpeedR*	Specifies the speed table number of the selected type by an integer from 0 to 15. *Specify "0" if speed is not set.
	|		
	8		
	7	*Accel / AccelS / AccelR*	Specifies the acceleration/deceleration table number of the selected type by an integer from 0 to 15. *Specify "0" if acceleration is not set.
	|		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Moves the arm to the specified point using circular interpolation in 3 dimensions.

For the rest about the command, refer to *Description* of Command 2006.

Example

When the midpoint is P1 and endPoint is P2, without an option.

Command	Response
07D7H 0000H 0001H 0002H	07D7H 0000H 0000H

10.30 Jog & Teach

Command 2050: Jog

Moves the manipulator using Jog motion.

Command Syntax

	bit	Name	Description
Parameter 1	15	*reserved*	Specify “0”.
	|		
	3		
	2	*mode*	Select the Jog mode. 0=World 1=Joint
	1		
	0		

	bit	Name	Description
Parameter 2	15	*axisSelection*	Select the target axis. When World is selected: 1=X axis 2=Y axis 3=Z axis 4=U axis 5=V axis 6=W axis When Joint is selected: 1=J1 2=J2 3=J3 4=J4 5=J5 6=J6
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 3	15	*distance* High-order word	Specifies the move distance (real number) as the value × 1000 converted to a 32-bit integer. When World is selected: X, Y, Z = mm U, V, X = deg When Joint is selected: For prismatic joints, (Unit: mm) For rotational joints, (Unit: deg) High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 4	15	*distance* Low-order word	Specifies the move distance (real number) as the value × 1000 converted to a 32-bit integer. When World is selected: X, Y, Z = mm U, V, W = deg When Joint is selected: For prismatic joints, (Unit: mm) For rotational joints, (Unit: deg) Low-order side 16 bit.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Moves the manipulator using Jog motion by selecting World or Joint.

World moves the manipulator in the World coordinate system while Joint moves the manipulator by each joint.

This command only supports step jogs.

Example

When moving the X axis 120.005 mm in World mode.

Command	Response
0802H 0000H 0001H 0001H D4C5H	0802H 0000H 0000H

Command 2051: Teach Point

Teaches the result of Jog to a point.

Command Syntax

Parameter 1	bit	Name	Description
	15 │ 0	*reserved*	Specify “0”.

Parameter 2	bit	Name	Description
	15 14 │ 1 0	*pointNumber*	Specifies the point number to be taught.

Response Syntax

Refer to *7. Response code.*

Description

Teaches the coordinate where Jog is executed to the point specified.

Use the point edit command to configure the flags as necessary.

Example

When teaching the coordinate to P5.

Command	Response
0803H 0000H 0005H	0803H 0000H 0000H

Command 2052: Save Points

Saves the taught points to the point file.

Command Syntax

	bit	Name	Description
Parameter 1	15 | 0	*reserved*	Specify “0”.

Response Syntax

Refer to *7. Response code.*

Description

Saves the taught points to the point file.

Point file: Points.pts

Example

When teaching the coordinate to P5.

Command	Response
0803H 0000H 0005H	0803H 0000H 0000H

Command 2053: Set Locked Axes

Controls the excitation of joints.

Command Syntax

Parameter 1	bit	Name	Description
	15 │ 1	*reserved*	Specify "0".
	0	*control*	Select ON/OFF of excitation. 0=OFF 1=ON

Parameter 2	bit	Name	Description
	15 14 │ 1 0	*jointSelect*	Specifies the joint number to be controlled. 0 = All joints 1 = Joint #1 2 = Joint #2 3 = Joint #3 4 = Joint #4 5 = Joint #5 6 = Joint #6

Response Syntax

Refer to *7. Response code.*

Description

The command controls the excitation of the specified joints.

Select ON/OFF of the joint in Parameter 1.

Select the target joints in Parameter 2. When 0 is selected, all joints can be controlled at one time.

Example

To turn ON Joint #3.

Command	Response
0805H 0001H 0003H	0805H 0000H 0000H

Command 2054: Get Locked Axes

Acquires the excitation status of joints.

Command Syntax

No parameter

Response Syntax

<table>
<tr><th rowspan="10">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Returns “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td><td>Joint #6</td><td rowspan="6">0= no excitation
1= excitation</td></tr>
<tr><td>4</td><td>Joint #5</td></tr>
<tr><td>3</td><td>Joint #4</td></tr>
<tr><td>2</td><td>Joint #3</td></tr>
<tr><td>1</td><td>Joint #2</td></tr>
<tr><td>0</td><td>Joint #1</td></tr>
</table>

Description

Returns the excitation status of each joint.

10.31 I/O Control

The following commands are to output and inout to I/O.

- Inout/Output by bit (1bit).
- Inout/Output by byte(8bits).
- Inout/Output by word(16bits).

Command 2100: Get Input Byte

Returns the status of the specified byte port.

Each port contains 8 input channels.

Command Syntax

	bit	Name	Description
Parameter 1	15	*byteportNumber*	Specifies the byte port of I/O.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*reserved*	Returns "0".
	|		
	8		
	7	*returnValue*	Returns the acuqires state by binary.
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Description

By using the command, you can check values of eight input bits at the same time.

Since eight channels can be checked at a time, the return values are integers ranging from 0-255.

See the chart below to check the correspondence of the integer return values and each input channel.

Input Channel Result (Using Byte port #0)

Return Value	7	6	5	4	3	2	1	0
1	Off	Off	Off	Off	Off	Off	Off	On
5	Off	Off	Off	Off	Off	On	Off	On
15	Off	Off	Off	Off	On	On	On	On
255	On	On	On	On	On	On	On	On

Input Channel Result (Using Byte port #2)

Return Value	7	6	5	4	3	2	1	0
3	Off	Off	Off	Off	Off	Off	On	On
7	Off	Off	Off	Off	Off	On	On	On
32	Off	Off	On	Off	Off	Off	Off	Off
255	On	On	On	On	On	On	On	On

Example

When acquiring the value from byte port #2.

Value	23	22	21	20	19	18	17	16
7	Off	Off	Off	Off	Off	On	On	On

Command	Response
0834H 0002H	0834H 0007H

Command 2101: Get Input Word

Returns the status of the specified input word port.
Each word port contains 16 input bits.

Command Syntax

	bit	Name	Description
Parameter 1	15	*byteportNumber*	Specifies the byte port of I/O.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*returnValue*	Returns the status of the input port (from 0 to 65535).
	14		
	|		
	1		
	0		

Description

Returns the status of the specified input port by word.

Example

When inputing from the word port #10.

Word Port #10 = 5AA5H

Command	Response
0835H 000AH	0835H 5AA5H

Command 2102: Set Output Byte

Sets 8 output bits simultaneously.

Command Syntax

<table>
<tr><th></th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="5">Parameter 1</td><td>15</td><td rowspan="5">byteportNumber</td><td rowspan="5">Specifies the byte port of I/O.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th></th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="11">Parameter 1</td><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Specify “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td><td rowspan="8">outData</td><td rowspan="8">Specifies the output byte.</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td></tr>
<tr><td>2</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

Response Syntax

Refer to *7. Response code.*

Description

The command simultaneously sets 8 output I/O bits using the combination of the *portNumber* and *outData* values.

The *portNumber* parameter specifies which group of 8 outputs to use where *portNumber* = 0 means outputs 0-7, *portNumber* = 1 means outputs 8-15, etc.

Once a *portNumber* is selected, a specific output pattern must be defined. The *outData* parameter may have a value between 0-255.

The table below shows some of the possible I/O combinations and their associated *outData* values assuming that *portNumber* is 0, and 1 accordingly.

Output Settings When *portNumber*=0 (Output number)

outData Value	7	6	5	4	3	2	1	0
01	Off	Off	Off	Off	Off	Off	Off	On
02	Off	Off	Off	Off	Off	Off	On	Off
03	Off	Off	Off	Off	Off	Off	On	On
08	Off	Off	Off	Off	On	Off	Off	Off
09	Off	Off	Off	Off	On	Off	Off	On
10	Off	Off	Off	On	Off	Off	Off	Off
11	Off	Off	Off	On	Off	Off	Off	On
99	Off	On	On	Off	Off	Off	On	On
255	On	On	On	On	On	On	On	On

Output Settings When *portNumber*=1 (Output number)

outData Value	15	14	13	12	11	10	9	8
01	Off	Off	Off	Off	Off	Off	Off	On
02	Off	Off	Off	Off	Off	Off	On	Off
03	Off	Off	Off	Off	Off	Off	On	On
08	Off	Off	Off	Off	On	Off	Off	Off
09	Off	Off	Off	Off	On	Off	Off	On
10	Off	Off	Off	On	Off	Off	Off	Off
11	Off	Off	Off	On	Off	Off	Off	On
99	Off	On	On	Off	Off	Off	On	On
255	On	On	On	On	On	On	On	On

Example

When outputing 255 to Byte Port #10.

Command	Response
0836H 000AH 00FFH	0836H 0000H 0000H

Command 2103: Set Output Word

Sets the status of output port to 16 output bits by word simultaneously.

Command Syntax

	bit	Name	Description
Parameter 1	15	*wordPortNum*	Specifies the word port of the I/O.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*outData*	Specify t he output data (integer from 0 to 65535).
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Changes the current status of the user I/O output port group specified by the word port number to the specified output data.

Example

When outputting 23205(5AA5H) to the Word Port #10.

Command	Response
0837H 000AH 5AA5H	0837H 0000H 0000H

Command 2104: Get Input Bit

Acquires the status of selected input port.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bitNumber*	Specifies the word port of the I/O.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*status*	Return the status of the selected inout. 0=OFF 1=ON
	14		
	|		
	1		
	0		

Description

The command checks the status of I/O input. This is most commonly used to check the status of sensors connected to the loader, conveyor, gripper solenoid, or other peripheral devices which works via I/O. Input status are “1” or “0”. They indicate ON (1) or OFF (0) of the device.

Example

When acquiring the status of bit number 15.
Bit number 15 is ON.

Command	Response
0838H 000FH	0838H 0001H

Command 2105: Set Output Bit On

Turns ON the specified output bit.

Command Syntax

<table>
<tr><th></th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="5">Parameter 1</td><td>15</td><td rowspan="5">bitNumber</td><td rowspan="5">Specifies the bit number to be turned ON.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

Response Syntax

Refer to *7. Response code.*

Description

The command turns ON (sets to 1) the specified output.

CAUTION

- **Output bits configured as remote**

  If an output bit which was set up as remote is specified, an error will occur. Remote output bits are turned ON or OFF automatically according to system status.

- **When an emergency stop occurs**

  The Controller has a feature which causes all outputs to go off when an E-Stop occurs. To keep the setting in emergency, follow the procedure described in the *EPSON RC+ User's Guide* for your Controller.

Example

When turning ON the output bit number 15.

Command	Response
0839H 000FH	0839H 0000H 0000H

Command 2106: Set Output Bit Off

Turns OFF the specified output bit.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bitNumber*	Specifies the bit number to be turned OFF.
	14		
	│		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

The command turns OFF (sets to 0) the specified output.

CAUTION	■ **Output bits configured as remote** If an output bit which was set up as remote is specified, an error will occur. Remote output bits are turned ON or OFF automatically according to system status. ■ **When an emergency stop occurs** The Controller has a feature which causes all outputs to go off when an E-Stop occurs. To keep the setting in emergency, follow the procedure described in the User's Guide for your Controller.

Example

When turning OFF the output bit number 15.

Command	Response
083AH 000FH	083AH 0000H 0000H

Command 2107: Get Memory Byte

Acquires the status of the specified memory I/O port.
Each port contains 8 memory bits.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bytePortNumber*	Specifies the byte port of the memory I/O.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*reserved*	Returns “0”.
	|		
	8		
	7	*returnValue*	Returns the integer from 0 to 255. The return value is 8 bits, with each bit corresponding to 1 memory I/O bit.
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Description

By using the command, you can check values of eight memory I/O bits at the same time.

Since eight channels can be checked at a time, the return values are integers ranging from 0-255.
See the chart below to check the correspondence of the integer return values and each memory I/O bit.

Memory I/O Bit Result (Using Port #0)

Return Value	7	6	5	4	3	2	1	0
1	Off	Off	Off	Off	Off	Off	Off	On
5	Off	Off	Off	Off	Off	On	Off	On
15	Off	Off	Off	Off	On	On	On	On
255	On	On	On	On	On	On	On	On

Memory I/O Bit Result (Using Port #31)

Return Value	255	254	253	252	251	250	249	248
3	Off	Off	Off	Off	Off	Off	On	On
7	Off	Off	Off	Off	Off	On	On	On
32	Off	Off	On	Off	Off	Off	Off	Off
255	On	On	On	On	On	On	On	On

Example

When acquiring the status of Port #0.

The status of Port #0 is 32.

Command	Response
083BH 0000H	083BH 0010H

Command 2108: Get Memory Word

Acquires the status of the specified memory I/O word port.
Each word port contains 16 memory I/O bits.

Command Syntax

	bit	Name	Description
Parameter 1	15	*wordPortNumber*	Specifies the memory I/O word port.
	14		
	│		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*returnValue*	Returns the status of the memory I/O (from 0 to 65535).
	14		
	│		
	1		
	0		

Description

Returns the status of the specified memory I/O word port.

Example

When acquiring the status of Port #1.

The statsu of Port #1 is 65535.

Command	Response
083CH 0001H	083CH FFFFH

Command 2109: Set Memory Byte

Simultaneously sets 8 memory I/O bits.

Command Syntax

	bit	Name	Description
Parameter 1	15	*portNumber*	Specifies the byte port of the memory I/O.
	14		
	│		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*reserved*	Returns “0”.
	│		
	8		
	7	*outData*	Returns the output pattern of the output group specified by *portNumber* by an integer from 0 to 255.
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Response Syntax

Refer to 7. Response code.

Description

The command simultaneously sets eight memory I/O bits using the combination of *portNumber* and *outData* values. The *portNumber* parameter specifies which group of 8 outputs to be used. For example, if *portNumber* = 0, output bits 0-7 are used. If *portNumber* = 1, output bits 8-15 are used..

Once *portNumber* is selected, a specific output pattern must be defined using the *outData* parameter. The *outData* parameter is an integer value between 0-255.

The tables below show combination examples of I/O and their associated *outData* values when *portNumber* is 0 or 1 accordingly.

Output Settings When *portNumber*=0 (Output number)

outData Value	7	6	5	4	3	2	1	0
01	Off	Off	Off	Off	Off	Off	Off	On
02	Off	Off	Off	Off	Off	Off	On	Off
03	Off	Off	Off	Off	Off	Off	On	On
08	Off	Off	Off	Off	On	Off	Off	Off
09	Off	Off	Off	Off	On	Off	Off	On
10	Off	Off	Off	On	Off	Off	Off	Off
11	Off	Off	Off	On	Off	Off	Off	On
99	Off	On	On	Off	Off	Off	On	On
255	On	On	On	On	On	On	On	On

Output Settings When *portNumber*=1 (Output number)

outData Value	15	14	13	12	11	10	9	8
01	Off	Off	Off	Off	Off	Off	Off	On
02	Off	Off	Off	Off	Off	Off	On	Off
03	Off	Off	Off	Off	Off	Off	On	On
08	Off	Off	Off	Off	On	Off	Off	Off
09	Off	Off	Off	Off	On	Off	Off	On
10	Off	Off	Off	On	Off	Off	Off	Off
11	Off	Off	Off	On	Off	Off	Off	On
99	Off	On	On	Off	Off	Off	On	On
255	On	On	On	On	On	On	On	On

Example

When outputting 254 to Port #1.

Command	Response
083DH 0001H 00FEH	083DH 0000H 0000H

Command 2110: Set Memory Word

Sets the status of 16 memory I/O ports by word simultaneously.

Command Syntax

	bit	Name	Description
Parameter 1	15	*wordPortNum*	Specifies the memory I/O word (from 0 to 31).
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*outData*	Specifies the memory I/O data (integer from 0 to 65535).
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Changes the current status of memory I/O port group specified by the word port number to the specified output data.

Example

When outputting 65535 to Port #1.

Command	Response
083EH 0001H FFFFH	083EH 0000H 0000H

Command 2111: Get Memory Bit

Acquires the status of the specified memory I/O bit.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bitNumber*	Specifies the value representing the memory I/O bit number.
	14		
	｜		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*returnValue*	Returns the status of specified bit. 0 = OFF 1 = ON
	14		
	｜		
	1		
	0		

Description

Returns the status of specified memory I/O bit.

Example

When acquring the statsu of Bit #20.
Bit #20 is ON.

Command	Response
083FH 0014H	083FH 0001H

Command 2112: Set Memory Bit On

Turns ON the specified bit of the memory I/O.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bitNumber*	Specifies the memory I/O bit by an integer.
	14		
	⏐		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Turns ON (set to 1) the specified bit of the memory I/O.

Example

When turing ON the Bit #30.

Command	Response
0840H 001EH	0840H 0000H 0000H

Command 2113: Set Memory Bit Off

Turns OFF the specified bit of the memory I/O.

Command Syntax

	bit	Name	Description
Parameter 1	15	*bitNumber*	Specifies the memory I/O bit by and integer.
	14		
	|		
	1		
	0		

Response Syntax

Refer to *7. Response code.*

Description

Turns OFF (set to 0) the specified bit of the memory I/O.

Example

When turing OFF the Bit #30.

Command	Response
0841H 001EH	0840H 0000H 0000H

Command 2150: Get Current Position

Returns the current position of the manipulator.

Command Syntax

<table>
<tr><th></th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="11">Parameter 1</td><td>15
|
6</td><td>reserved</td><td>Specify “0”.</td></tr>
<tr><td>5
4</td><td>format</td><td>Select the format of data.
0: World
1: Joint
2: Pulse
3: Flag</td></tr>
<tr><td>3</td><td>reserved</td><td>Specify “0”.</td></tr>
<tr><td>2
1
0</td><td>jointNumber</td><td>Specifies the coordinate or axis to acquire the position.
* If “3: Flag” is selected in Format, specify “0”.

[Format: World]
1: X coordinate
2:Y coordinate
3: Z coordinate
4: U coordinate
5: V coordinate
6: W coordinate
[Format: Joint or Pulse]
1: Joint #1
2: Joint #2
3: Joint #3
4: Joint #4
5: Joint #5
6: Joint #6</td></tr>
</table>

Response Syntax

Format of the response data varies depending on the format of data to be acquired.

When format is either World, Joint, or Pulse.

<table>
<tr><th rowspan="6">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">position
High-order word</td><td rowspan="5">Returns the high-order word (16 bit) of the position.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

<table>
<tr><th rowspan="6">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="5">position
Low-order word</td><td rowspan="5">Returns the low-order word (16 bit) of the position.</td></tr>
<tr><td>14</td></tr>
<tr><td>|</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

When World or Joint is selected

Returns the position information as the actual value × 1000 converted to a 32-bit integer. If the setting value is a negative number, returns the value in 32-bit two’s complement.

When World is selected:

X, Y, Z = mm
U, V, W = deg

When Joint is selected:

For prismatic joints, Unit: mm
For rotational joints, Unit: deg

When Pulse is selected

Returns the value by pulse (32-bit integer without offset)

When format is Flag.

<table>
<tr><th rowspan="7">Response 1</th><th>bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td rowspan="3">reserved</td><td rowspan="3">Specify “0”.</td></tr>
<tr><td>|</td></tr>
<tr><td>3</td></tr>
<tr><td>2</td><td>wrist</td><td>0=NoFlip
1=Flip
* Effective only for 6-axis manipulators.</td></tr>
<tr><td>1</td><td>elbow</td><td>0=Above
1=Below
* Effective only for 6-axis manipulators.</td></tr>
<tr><td>0</td><td>hand</td><td>0=Lefty
1=Righty</td></tr>
</table>

Response 2	bit	Name	Description
	15	*localNumber*	Returns the local number
	14		
	｜		
	1		
	0		

Response 3	bit	Name	Description
	15	*j4flag*	Returns the statsu of j4flag. 0＝J4F0 1＝J4F1 * Available only for 6-axis manipulators.
	14		
	｜		
	1		
	0		

Response 4	bit	Name	Description
	15	*j6flag*	Returns the status of j6flag. 0 = J6F0 ｜ 127 = J6F127 * Available only for 6-axis manipulators.
	14		
	｜		
	1		
	0		

Description

The command returns the information of current manipulator position. This command can be executed when the manipulator is stopped.
Select the information to be acquired in Parameter 1 .
To acquire the position information, for instance, select either World, Joint, or Pulse in *format* and select axis to acquire the information.

To acquire all information, the command must be executed several times.

Example

When acquring Y-axis coordinate in World. Y coordinate is 100.002 mm.

Command	Response
0866H 0002H	0866H 0001H 86A2H

Command 2151: Get Distance Between Points

Acquires the distance between two manipulator coordinates.

Command Syntax

	bit	Name	Description
Parameter 1	15	*point1*	Specify one of two point numbers to acquire the distance.
	14		
	|		
	1		
	0		

	bit	Name	Description
Parameter 2	15	*point2*	Specify one of two point numbers to acquire the distance.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*position* High-order word	Returns the acquired distance (mm/ real number) as the value × 1000 converted to a 32-bit integer. High-order side 16 bit.
	14		
	|		
	1		
	0		

	bit	Name	Description
Response 2	15	*position* Low-order word	Returns the acquired distance (mm/ real number) as the value × 1000 converted to a 32-bit integer. Low-order side 16 bit.
	14		
	|		
	1		
	0		

Description

Returns the distance between two manipulator coordinates. (Unit: mm)

Example

When acquiring the distance between P1 and P2. The distance is 100.002 mm.

Command	Response
0867H 0001H 0002H	0867H 0001H 86A2H

Command 2152: Get Target OK

Returns the status whether the PTP (Point to Point) motion from the current position to a target position is possible or not.

Command Syntax

	bit	Name	Description
Parameter 1	15	*targetPos*	Specifies the point number to verify.
	14		
	|		
	1		
	0		

Response Syntax

	bit	Name	Description
Response 1	15	*result*	Returns whether the PTP motion to the target position ir possible or not. 0 = Impossible 1 = Possible
	14		
	|		
	1		
	0		

Description

This command verifies whether the manipulator can reach to the target position and the orientation before actual operation. The motion trajectory to the target point is not considered.

Example

When verifying the motion to P2. When the motion is possible.

Command	Response
0868H 0002H	0868H 0001H

Command 2153: Get Manipulator Type

Acquires the manipulator type.

Command Syntax

No parameter.

Response Syntax

	bit	Name	Description
Response 1	15	*type*	Returns the manipulator type. 1: Joint 2: Cartesian 3: SCARA 5: 6-AXIS 6: RS series
	14		
	|		
	1		
	0		

Description

Returns the manipulator type.

Example

When the manipulator is 6-AXIS .

Command	Response
0869H	0869H 0005H

Command 2154: Get Manipulator Model

Returns the manipulator model name.

Command Syntax

	bit	Name	Description
Parameter 1	15 | 1	*reserved*	Specify “0”.
	0	*Start / Continue*	Specify whether to acquire the manipulator model name from the top. 0=Acquire continuously 1=Start from the top

Response Syntax

	bit	Name	Description
Response 1	15 | 3	*reserved*	Returns “0”.
	2 1 0	*acquisitionStatus*	0= Indicates the last character. 1= Indicates there is character(s) remained. 2= Indicates the sending is completed.

	bit	Name	Description
Response 2	15 | 8	*reserved*	Returns “0”.
	7 | 0	*charaCode*	ASCII code

Description

Returns the string containing the model name. This is the name shown on the rear panel of the manipulator.

To acquire the model name, follow the steps below.

1) Specify “1=Start from the top” in Parameter 1 and issue the command.
2) Acquire the first character from the response.
3) Specify “0=Acquire continuously” in Parameter 1 and issue the command.
4) Acquire one character from the response.
5) Check the status of response and repeat the steps from the step 3) if the character is left (1).
6) If the response status is “0” (the last character), finish the acquisition.

Example

When the model name is “G6-551S-II”.

Command	Response
086AH 0001H	086AH 0001H 0047H
086AH 0000H	086AH 0001H 0036H
086AH 0000H	086AH 0001H 002DH
086AH 0000H	086AH 0001H 0035H
086AH 0000H	086AH 0001H 0035H
086AH 0000H	086AH 0001H 0031H
086AH 0000H	086AH 0001H 0053H
086AH 0000H	086AH 0001H 002DH
086AH 0000H	086AH 0001H 0049H
086AH 0000H	086AH 0000H 0049H

Command 2155: Get Error Code

Acquires the error code during the controller error.

Command Syntax

No parameter

Response Syntax

	bit	Name	Description
Response 1	15	*errorCode*	Returns the controller error code.
	14		
	|		
	1		
	0		

Description

Acquires the error code when controller is in the error state.
When executed the command during normal state, error code 0000H will be returned.