ESTIMATE No.

ORDER No.

CUSTOMER
Synetics Solutions, Inc.

USER
KLA-Tencor Corporation

YASKAWA

CONTENTS: **Specifications for Communications
For Wafer Transfer Manipulator**

SER-S

NOTE	COPIES		YASKAWA ELECTRIC CORPORATION
			Technology Department Section1
	CUSTOMER		Semiconductor Robot Plant
	USER		Semiconductor Robot Business Division
			Robotics Automation Division
			DR. M.Matuo
			CK. -
			APP. T.Fukunaga
			DWG No. **HE0400985**

Revisions

Drawing No.	Name		Revised by	Approved by
HE0400985	Specifications for Communications for Wafer Transfer Manipulator			
Date	Revision No.	Revised content		
Jul.02.2003	0		-	-
Apl.11.2005	<6>	Overall revision	M.Matsuo	M.Ogasawara
Oct.14.2005	<7>	Addition of new Commands. (MCDT) P17,P52 Addition of new Parameter. (R offset for PUT) P37,P40 Addition of new Commands detail (MCDT). P95a,P95b (V2.60)	N.Nakashima	M.Ogasawara

CONTENTS

1. Introduction

This manual describes the specifications of the communication between the tool host controller (hereafter called "Host") and the Yaskawa Wafer Transfer Manipulator controller (hereafter called "Controller").

Use this manual in conjunction with the following:
• Parameter Lists for Wafer Transfer Manipulator: HE0400986
• Error Code Lists for Wafer Transfer Manipulator: HW0481772
• Teach Manual for Wafer Transfer Manipulator (Z524i): HE0401335
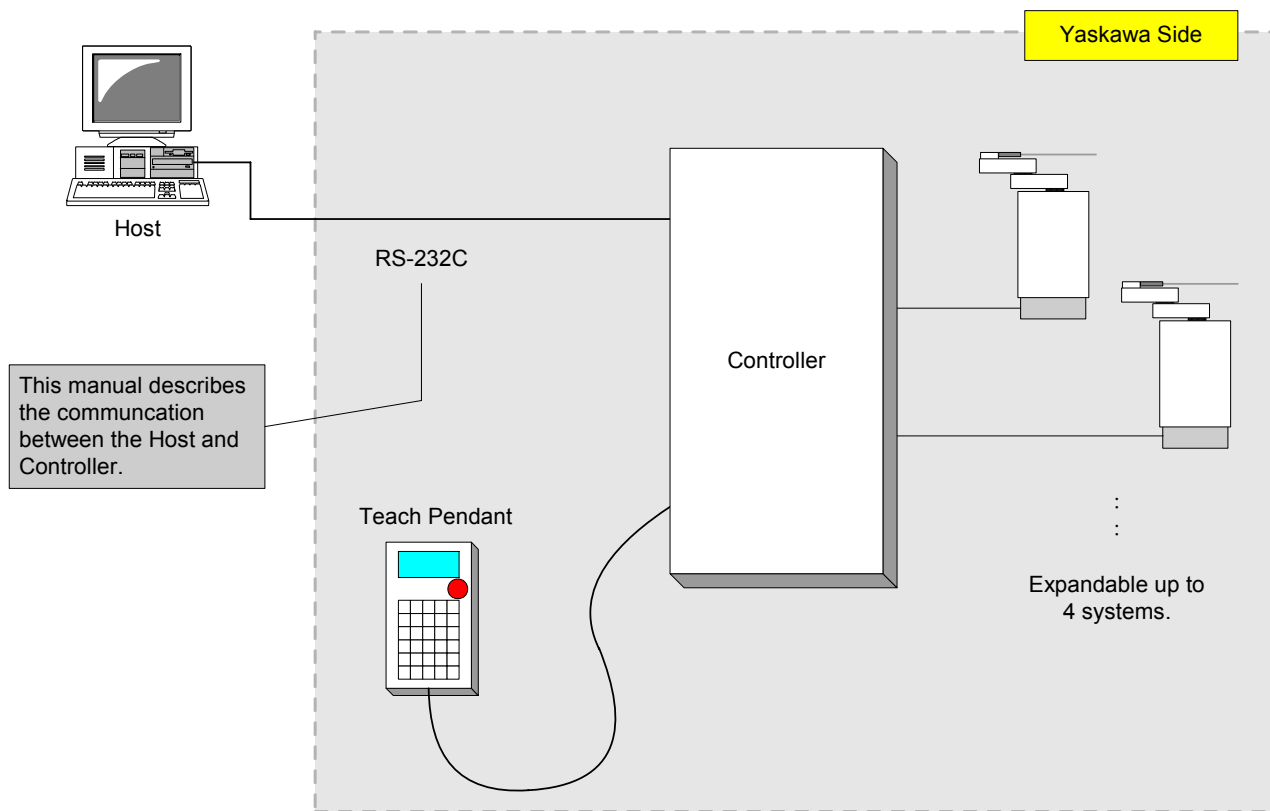
2. System Configuration

Fig. 1 System Configuration

2-1 Outline

The Host and the Controller are connected via RS-232C.
In this manual, "unit" refers to either the wafer transfer manipulator (hereafter called "manipulator") or the Pre-aligner.

2-2 Basic Operations of the Host and the Controller

The basic operations of the Host and the Controller are as follows:

❑ Host

- Requests operation to the Controller for a specified unit (manipulator or Pre-aligner) and receives the results of the request.
- Requests settings and status reports to the Controller for a specified unit (manipulator or Pre-aligner) and receives the results of the request.
- Retries communication and handles errors if communication errors occur.

❑ Controller

- Operates a specified unit (manipulator or Pre-aligner) per the Host's request and reports the results of the operation.
- Provides status reports and performs setting changes per the Host's request and reports the results of the operation.
- Reports error occurrences to the Host as required when a communication errors occur.

3. Communications Specifications

Item	Standards
Communications Code	ASCII code
Synchronizing System	Conforming to RS-232C standards, non-synchronous (start-stop synchronizing)
Baud Rate	150 / 300 / 600 / 1200 / 2400 / 4800 / 9600 / 19200 bps
Communications Method	Full-duplex, serial
Data Type	Data length: 7 / 8 bits
	Parity: NONE / ODD / EVEN
	Stop bit: 1 / 1.5 / 2 bits
Transmission Control	Not Available / Available

The baud rate and data type (data length, parity, and stop bit) can be set by the integer parameters at the Controller. Refer to "Parameter Lists for Wafer Transfer Manipulator" for details.

Default settings are as follows:

- Baud rate: 9600 bps
- Data length: 8 bits
- Parity: NONE
- Stop bit: 1 bit
- Transmission Cntrol   Not Available

> ### Related Parameters
>
> - Baud rate setting parameter
> - Data length setting parameter
> - Parity setting parameter
> - Stop bit setting parameter
> - Transmission Control setting parameter

4.   Message Transmission Specifications

4-1   Types of Commands

The following types of commands can be sent to the Controller:

Motion Command: Moves a specified unit (manipulator or Pre-aligner).

Control Command: Controls the state of a specified unit (e.g. motor servo ON/OFF).

Setting Command: Sets parameters and position information that are required for a unit's operation (e.g. speed setting and position registration).

Reference Command: References parameters or position information that have been set for a unit (manipulator or Pre-aligner).

Execution Complete Acknowledgement: Reports the acknowledgement of the **Execution Complete Message** to the Controller.

Maintenance Command: Upload/Download internal parameters and position data to the Controller.

4-2   Command Message (Host to Controller)

A command is transmitted from the Host in the following format.

$	Uno	COMMAND	Parameter	Sum	CR

1) $:   Start mark (1 byte)
   - Indicates the start of the message.
2) UNo: Unit Number (1 byte)
   - Indicates the unit type
       '1': Manipulator
       '2': Pre-aligner
3) COMMAND: Command (4 bytes)
   - All commands are defined as fixed length character strings.
4) Parameter: Parameter (Command Dependent.)
   - Sets the axis of operation, the motion amount, etc.

5) Sum: Checksum (2 bytes)

- Checks the message for error caused by a communication error.
- The check sum is the last two digits (in ASCII code) of the sum of the character string containing the "Uno", "COMMAND" and "Parameter" represented in ASCII code ('0' to '9' and 'A' to 'F').

<u>Example</u>

Message: $1MHOMF<Sum><CR>
'1' = 31H, 'M' = 4DH, 'H' = 48H, 'O' = 4FH, 'M' = 4DH, 'F' = 46H

Sum = 31H + 4DH + 48H + 4FH + 4DH + 46H = 1A8H
(Note: H indicates a hexadecimal number)

The checksum value is A8H.

Therefore, the message would read "$1MHOMFA8<CR>."

- The check sum can be enabled or disabled by parameter setting. When the checksum is disabled, the two bytes for the checksum section are omitted
  Refer to "Parameter Lists for Transfer Manipulator" for details.
- Checksum is enabled as the default setting.
  .

6) <CR>:   End mark (1 byte)

- Indicates the end of the message. (<CR> = 0DH)

Related Parameters

- Checksum enable/disable parameter

4-3   Response Message (Controller to Host)

When a command from the Host is correctly received, a **Response Message** is transmitted from the Controller in the following format.

@	Uno	Sts₁ₛₜ	Sts₂ₙₐ	Ackcd	Subcd	Sum	CR

1) @:   Start mark (1 byte)
   ▪   Indicates the start of the message.
2) UNo: Unit number (1 byte)
   ▪   Indicates the unit type.
            '1': Manipulator
            '2': Pre-aligner
3) Sts: Status (2 byte)
   ▪   Responds the status of the specified unit by the command.
   ▪   Refer to the following table for the status that is sent to the Host. The value will appear in ASCII code for each item.

* 1ˢᵗ Byte
<Case of Manipulator>

Bit No.	Item	Contents
1	Sensor Status 1	Wafer sensor status of End-Effector1 0: Workpiece provided (ON) 1: Workpiece not provided (OFF)
2	Sensor Status 2	Wafer sensor status of End-Effector2 0: Workpiece provided (ON) 1: Workpiece not provided (OFF)
3	Solenoid valve Status 1	Solenoid valve status of End-Effector1 0: Workpiece released (ON) 1: Holding workpiece (OFF)
4	Solenoid valve Status 2	Solenoid valve status of End-Effector2 0: Workpiece released (ON) 1: Holding workpiece (OFF)

<Case of Pre-aligner>

Bit No.	Item	Contents
1	Sensor Status 1	Wafer sensor status (Vacuum sensor) 0: Workpiece provided (ON) 1: Workpiece not provided (OFF)
2	Sensor Status 2	Wafer sensor status (CCD) 0: Workpiece provided (ON) 1: Workpiece not provided (OFF)
3	Solenoid valve Status 1	Solenoid valve status of chuck 0: Workpiece released (ON) 1: Holding workpiece (OFF)
4		Open for future use

* 2nd Byte

Bit No.	Item	Contents
1	Battery Status	The battery voltage status for encoder and NXC100 memory 1: Either of Battery voltage dropped 0: Normal status
2	Unit Status	Motion/Control Command execution status for a specified unit 1: Ready 0: Busy
3	Servo Status	The servo status for a specified unit 1: Servo OFF 0: Servo ON
4	Error Status	Serious Error 1: Error occurence 0: No error occurrence

4) Ackcd: Response code (4 bytes)
- Indicates whether the conditions described in the explanation for each command are satisfied or not and the arguments are correct or not. 0 indication means that the conditions are satisfied and the arguments are correct.
- The most significant digit shows the alarm level.
   0 ~ 3: Major Alarm
   4~9: Minor Alarm (ex. Parameter error response code is '9033')
   When either alarm does not occur, response code is '0000'.

5) Subcd: Sub code (4 bytes)
- Detailed alarm information is shown.
- When either alarm does not occur, sub code is '0000'.

6) Sum: Checksum (2 bytes)
- See Above 4-2 5).

7) <CR>:   End mark (1 byte)
- Indicates the end of the message. (<CR> = 0DH)

4-4   Communication Error Message (Controller to Host)

When a command from the Host is incorrectly received, a **Response Message** is sent to the Host from the Controller as shown below.

The suspected causes for the message error are:

➢ Checksum error
➢ Incorrect unit number

?	Ackcd	Subcd	Sum	CR

1) ?:   Start mark (1 byte)
   ▪ Indicates the start of a **Communication Error Message**.
2) Ackcd: Response code (4 bytes)
   ▪ See Above 4-3 4).
3) Subcd: Sub code (4 bytes)
   ▪ See Above 4-3 5).
4) Sum: Checksum (2 bytes)
   ▪ See Above 4-2 6).
5) <CR>:   End mark (1 byte)
   ▪ Indicates the end of the message. (<CR> = 0DH)

4-5   Execution Complete Message (Controller to Host)

When the execution of a motion or control has been completed, an **Execution Complete Message** is sent to the Host from the Controller as shown below.

$	Uno	Sts	Errcd	Subcd	COMMAND	Value	Sum	CR

1) $: Start mark (1 byte)
- Indicates the start of the message.

2) UNo: Unit number (1 byte)
- Indicates the unit type.
  - '1': Manipulator
  - '2': Pre-aligner

3) Sts: Status (2 byte)
- See Above 4-3 3).

4) Errcd: Error code (4 bytes)
- Indicates the error that occurred when the Host request was executed.
- The most significant digit shows the alarm level.
  - 0 ~ 3: Major Alarm
  - 4~9: Minor Alarm
  - When either alarm does not occur, error code is '0000'.
- In case Ackcd of Response Message is other than 0, Errcd of Execution Complete Message is same value as Ackcd's of Response Message.

4') Subcd: Sub code (4 bytes)
- See Above 4-3 5).

5) COMMAND: Command (4 bytes)
- Indicates the executed command that was completed.

5') Value: Response data (Command Defined.)
- Response value for the reference command sent from the Host. Refer to the command explanations described following section 7.

6) Sum: Checksum (2 bytes)
- See Above 4-2 5).

7) <CR>: End mark (1 byte)
- Indicates the end of the message. (<CR> = 0DH)

4-6   Execution Complete Acknowledgement
(Host to Controller)

The Host sends an acknowledgement message to the Controller after receipt of an **Execution Complete Message**. This messaging can be enabled or disabled by parameter setting. Default is disable(0). (Refer to the "Parameter Lists for Wafer Transfer Manipulator" for details.)

Note: If the **Execution Complete Message** (see 4-5) is disabled, the acknowledgement messaging described here is automatically disabled.

$	Uno	COMMAND	Sum	CR

1) $:   Start mark (1 byte)
   - Indicates the start of the message.
2) UNo: Unit number (1 byte)
   - Indicates the unit type.
        '1': Manipulator
        '2': Pre-aligner
3) COMMAND: Command (4 bytes)
   - A fixed length character string defining "ACKN".
4) Sum: Checksum (2 bytes)
   - See Above 4-2 5).
5) <CR>: End mark (1 byte)
   - Indicates the end of the message. (<CR> = 0DH)

Related Parameters

- **Execution Complete Acknowledgement** enable/disable parameter

4-7   Asynchronous Information

When the pre-defined event is occurred, **Asynchronous Information Message** is sent to the Host from the Controller as shown below.

!	Uno	InfoMsg	Sum	CR

1) !:   Start mark (1 byte)
   - Indicates the start of the message.
2) UNo: Unit number (1 byte)
   - Indicates the unit type.
     '1': Manipulator
     '2': Pre-aligner
3) InfoMsg: Information Massage

Information Massage of Motion Commands=Current Event Code & Sub Code

Event	Event Code	Sub Code	
		Station Name 2 Bytes	Slot Number 2 Bytes
Get Event	"WGET"		
Put Event	"WPUT"		
Retracted Event	"ARET"		

Station name: "P1" to "P8", "UA" to "UL"
Slot Number: P1 to P2 Station: "01" to "25"
             UA to UL Station: "00"

Information Massage of User macro communication command=UNSOLICMSG

Event	UNSOLICMSG
Get Event	"GTPT-GET"
Put Event	"GTPT-PUT"
End Event	"GTPT-END"

4) Sum: Checksum (2 bytes)
   - See Above 4-2 5).
5) <CR>:   End mark (1 byte)
   - Indicates the end of the message. (<CR> = 0DH)

<u>Host</u>   <u>Robot</u>

$Command

@ACK

!Information
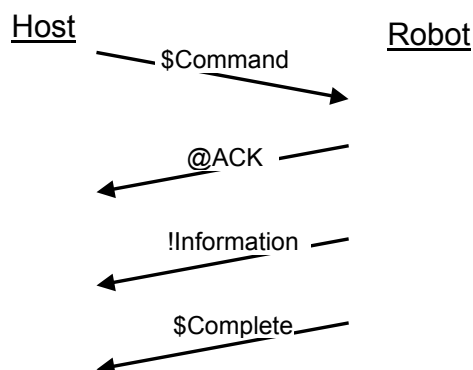
$Complete

4-8   STS data definition

Description about STS data in command completion response.

Basic rule

If the solenoid valve is in "wafer release" (ON)state, wafer status in STS might not correct. Even if the solenoid valve is in "wafer hold" (OFF)state, wafer status may not be correct if STS was checked right after solenoid valve control.

Example

1. When reference command like "RSTS" command was sent when the plunger is still moving during wafer transfer command execution.
2. STS in CSOL command. Current CSOL command is simply solenoid valve control command, and does not wait wafer sensor status become stable. Thus wafer status in STS is not correct even solenoid valve status is "wafer hold" (OFF)state.

Commands which do not control the solenoid valve by themselves

Setting command,

Reference command

Maintenance command

Control commands except CSOL command

If the solenoid valve is in "wafer release" (ON)state, wafer state in STS may not be correct.

Motion commands which do not control the solenoid valve by themselves

MABS

MRLK

MRLN

If the solenoid valve is in "wafer release" (ON)state, wafer state in STS may not be correct.

Motion commands which control the solenoid valve by themselves

STS in the completion response of these command is basically correct. In the case of vacuum End-Effector, robot will automatically turn off the vacuum if no wafer was detect on the blade in the motion command execution. Thus even the solenoid valve is in "wafer released" (ON)state, wafer status is basicaly correct.

Commands which should be mentioned specially

CRSM	This command has only ACK as response. STS in ACK is the data when robot
CEMG	Turn off the servo power. It does not wait until servo packs respond the result. Servo status in STS is correct
CSRV	Control the servo power and wait until servo pack respond the result. Servo status in STS is correct
CCLR	Clear the error and wait until the result. Then respond
CSOL	Respond immediate state. This command does not wait the plunger or vacuum sensor state become stable.

Command		STS data in ACK							STS data in completion data						
		Wafer	Solenoid valve		Battery	Unit status	Servo	Error	Wafer	Solenoid valve		Battery	Unit status	Servo	Error
			Vacuum	Edge						Vacuum	Edge				
Motion command															
1	MHOM	i	i	i	i	B	i	c	c	c	c	i	R	i	C
2	MTRS	i	i	i	i	B	i	c	c	c	c	i	R	i	C
3	MGET	i	i	I	i	B	i	c	c	c	c	i	R	i	C
4	MPUT	i	i	I	i	B	i	c	c	c	c	i	R	i	C
5	MGT2	i	i	i	i	B	i	c	c	c	c	i	R	i	C
6	MPT2	i	i	i	i	B	i	c	c	c	c	i	R	i	C
7	MSP2	i	i	i	i	B	i	c	c	c	c	i	R	i	C
8	MPNT	i	i	i	i	B	i	c	c	c	c	i	R	i	C
9	MMAP	i	i	i	i	B	i	c	c	c	c	i	R	i	C
10	MALN	i	i	i	i	B	i	c	c	c	c	i	R	i	C
11	MTCH	i	i	i	i	B	i	c	c	c	c	i	R	i	C
12	MABS	i	i	i	i	B	i	c	i	i	i	i	R	i	C
13	MRLK	i	i	i	i	B	i	c	i	i	i	i	R	i	C
14	MRLN	i	i	i	i	B	i	c	i	i	i	i	R	i	C
15	MACA	i	i	i	i	B	i	c	c	c	c	i	R	i	C
16	MMCA	i	i	i	i	B	i	c	c	c	c	i	R	i	C
17	MGTW	i	i	i	i	B	i	c	c	c	c	i	R	i	C
18	MPTW	i	i	i	i	B	i	c	c	c	c	i	R	i	C
19	MGWI	i	i	i	i	B	i	c	c	c	c	i	R	i	C
20	MPWI	i	i	i	i	B	i	c	c	c	c	i	R	i	C
21	MSWP	i	i	i	i	B	i	c	c	c	c	i	R	i	C
22	MSWI	i	i	i	i	B	i	c	c	c	c	i	R	i	C
23	MXTW	i	i	i	i	B	i	c	c	c	c	i	R	i	C
24	MXWI	i	i	i	i	B	i	c	c	c	c	i	R	i	C
24a	MCDT	i	i	i	i	B	i	c	c	c	c	i	R	i	C	<7>
25	ISYS	i	i	i	i	B	i	c	c	c	c	i	R	i	C
26	MWRM	i	i	i	i	B	i	c	c	c	c	i	R	i	C
Control command															
27	CHLT	i	i	i	i	B	i	c	i	i	i	i	R	i	C
28	CRSM	i	i	i	i	B	i	c	No completion message						
29	CEMG	i	i	i	i	B	i	c	No completion message						
30	CSRV	i	i	i	i	B	i	c	i	i	i	i	R	c	C
31	CCLR	i	i	i	i	B	i	c	i	i	i	i	R	i	C
32	CSOL	i	i	i	i	B	i	c	i	c	c	i	R	i	C
33	CCHK	i	i	i	i	B	i	c	c	c	c	i	R	i	C
34	CLFT	i	i	i	i	B	i	c	i	i	i	i	R	i	C
Setting command															
35	SSPP	No acknowledge message							i	i	i	i	R	I	C
36	SPOS								i	i	i	i	R	I	C
37	SABS								i	i	i	i	R	I	C
38	SPSV								i	i	i	i	R	I	C
39	SOFS								i	i	i	i	R	I	C

Command		STS data in ACK							STS data in completion data						
		Wafer	Solenoid valve		Battery	Unit status	Servo	Error	Wafer	Solenoid valve		Battery	Unit status	Servo	Error
			Vacuum	Edge						Vacuum	Edge				
Setting command															
40	SPIT								i	i	i	i	R	i	C
41	SSLT								i	i	i	i	R	i	C
42	SRSV								i	i	i	i	R	i	C
43	SMSK	No acknowledge message							i	i	i	i	R	i	C
44	SPRM								i	i	i	i	R	i	C
45	SALM								i	i	i	i	R	i	C
46	SSTD								i	i	i	i	R	i	C
47	SWSZ								i	i	i	i	R	i	C
Reference command															
48	RSPP								i	i	i	i	i	i	C
49	RPOS								i	i	i	i	i	i	C
50	ROFS								i	i	i	i	i	i	C
51	RCST								i	i	i	i	i	i	C
52	RMAP								i	i	i	i	i	i	C
53	RSTS								i	i	i	i	i	i	c
54	RERR								i	i	i	i	i	i	c
55	RMSK								i	i	i	i	i	i	c
56	RVER	No acknowledge message							i	i	i	i	i	i	c
57	RCFG								i	i	i	i	i	i	c
58	RSTT								i	i	i	i	i	i	c
59	RPRM								i	i	i	I	i	i	c
60	RCCD								i	i	i	I	i	i	c
61	RALM								i	i	i	i	i	i	C
62	RSTD								i	i	i	i	i	i	C
63	RWSZ								i	i	i	i	i	i	C
64	RIOS								i	i	i	i	i	i	C
Ack															
65	ACKN	No acknowledge message							No completion response						
Maintenance Command															
66	UPOS								i	i	i	i	R	i	c
67	UPRM	No acknowledge message							i	i	i	i	R	i	c
68	DPOS								i	i	i	i	R	i	c
69	DPRM								i	i	i	i	R	i	c
Reset Command															
70	HRST	i	i	i	i	B	i	c	i	i	i	i	R	i	C
Macro addition command															
71	MACR	User decides the response													

i: immediate, c: confirmed, B: Busy, R: Ready

The firmware update the status every 4msec.
the status in STS is the latest value at the time it make the response message.

5.   Communication Flow

This chapter describes the communication flow between the Host and the Controller.

5-1   Processing at the Host Side

1) After sending a **Command Message**, the Host will receive either a **Response Message** (indicating that the command is normal/good) or **Communication Error Message** (indicating there was an error in the command -checksum error, invalid unit number, etc.) from the Controller.

   The Host waits for this reponse from the Controller using a ***response time-out***. If the Host detects the response time-out, the Host should resend the commnad. If the Controller still cannot receive the command (due to a communications line error, etc.), no **Response Message** will be sent and the Host will detect a response time-out again. The Host should resend the command again until the ***maximum number of retries*** is reached.

2) When the Host sends a motion or control command (hereafter referred to as "execution commands") to the Controller, the Host performs further processing as follows:

   i.   The Host will receive an **Execution Complete Message** from the Controller. This informs the Host that no communication errors were in the **Command Message** and that it is not necessary to resend the **Command Message**.

   ii.   After receiving an **Execution Complete Message** from the Controller, the Host will send an **Execution Complete Acknowledgement** back to the Controller. This message is sent to the Controller to inform the Controller that the Host has confirmed the completion of the execution command. The Controller will not resend **Execution Complete Message** after receiving acknowledgement from the Host.

   In the event that the Controller cannot cannot send the **Execution Complete Message** due to communication line error, an ***operation time-out*** should be provided at the Host to avoid making the Host wait endlessly for the **Execution Complete Message**.

   The **Execution Complete Acknowledgement** can be enabled or disabled by parameter settings. Refer to "Parameter Lists for Wafer Transfer Manipulator" for details. (Both are enabled as the default settings.)

Related Parameters

- **Execution Complete Acknowledgement** enable/disable parameter

---

5-2   Controller Processing

1) After receiving a "$" character (beginning of the message) from the Host, the Controller continues to receive the command (holds it in a buffer) using *time-out monitoring between characters* until it receives the <CR> code (the end of the message).

   If the Controller detects time-out between characters, the received portion of the message (data stored in the buffer) is deleted. The Controller will not send a error notification to the Host.

   The time-out time between characters can be set by parameter setting (default is 0.1sec).

2) After a command is received from the Host, checksum error is checked. If any checksum error is detected, the Controller informs the Host of the error by the **Communication Error Message**.

3) After a commnad is received successfully, the Controller determines whether the command is to be accepted (valid or invalid) by checking the contents of parameters.

   If nothing is wrong, the Controller executes the requested command.

   If any error/invalidity is detected in the requested commnad, the Controller informs the Host of the error by the **Communication Error Message**.

4) After the execution command has been completed, the Controller informs the Host that the command is finished (**Execution Complete Message**) and waits for acknowledgement using *response time-out*.

   If the Controller detects a response time-out, the Controller resends the **Execution Complete Message**. The *maximum number of retries* must be set so that the Controller does not continue to resend the message inspite of a communications line error, etc.

   Response time-out time and the maximum number of retries can be set by parameters (default setting is 1 sec timeout and maximum 2 times retry).

The **Execution Complete Acknowledgement** can be enabled or disabled by parameter settings. Refer to "Parameter Lists for Wafer Transfer Manipulator" for details. (Both are enabled as the default settings.)

Related Parameters

- **Execution Complete Acknowledgement** enable/disable parameter
- Maximum number of retries in sending **Execution Complete Message**
- **Execution Complete Acknowledgement** wait time-out parameter
- Parameter for setting the time-out time between characters

5-3　Normal Communications Flow for Commands

❑　Communications Flow Chart for Execution Commands

After the execution commnad has been completed, the Controller informs the Host of the completion as shown in the following flow chart.

<When ACKN is omitted>

Host		Controller	Unit Status*
			Ready
Sends **Command Message** (execution command).	-->		Ready
	<--	Sends **Response Message** (normal).	Busy
			Busy
		(Execute the command.)	Busy
			Busy
	<--	Sends **Execution Complete Message**.	Ready
			Ready

<When ACKN is sent>

Host		Controller	Unit Status*
			Ready
Sends **Command Message** (execution command).	-->		Ready
	<--	Sends **Response Message** (normal).	Busy
			Busy
		(Execute the command.)	Busy
			Busy
	<--	Sends **Execution Complete Message**.	Busy
Sends **Execution Complete Acknowledgement** message ("**ACKN**").	-->		Ready
			Ready

∗ Indicates the response status of the unit.

5-4 Communications Flow Chart for Operation Interruption and Restart

Drive commands from the Host can be interruptted or restarted using the "CHLT" command and "CRSM" command, respectively.

<When ACKN is sent>

Host		Controller	Unit Status*
			Ready
Sends **Command Message** (execution command).	-->		Ready
	<--	Sends the **Response Message** (normal).	Busy
			Busy
		(Executes motion command.)	Busy
			Busy
Sends an interrupt command ("**CHLT**").	-->		Busy
	<--	Sends the **Response Message** (normal).	Busy
			Busy
	<--	Sends an **Execution Complete Message**.	Ready
			Ready
Sends a restart command ("**CRSM**").	-->		Ready
	<--	Sends the **Response Message** (normal).	Busy
			Busy
		(Executing motion command.)	Busy
			Busy
	<--	Sends an **Execution Complete Message**.	Busy
Sends **Execution Complete Acknowledgement** message ("**ACKN**").	-->		Ready
			Ready

*Indicates the response status of the unit.

Note 1: The Controller will not send an **Execution Complete Message** when a motion pause command (HOLD signal is turned ON) is given during unit operation. After the motion pause command is released (HOLD signal is turned OFF) the operation restarts, even if the unit does not receive a restart command. (Motion pause is not the same as the interrupt command shown above.)

Note2: When the unit is paused by motion pause command (HOLD signal is turned ON), the "Unit Status" of the Controller remains "Busy."

5-5   Processing for Communications Error

This section describes the processing when a communications error (excluding an error in **Command Message** request contents) occurs.

5-5-1 Error Occurs in the Message From Host

The following cases are presented as examples of Host side message errors.

1)   Garbage Character for '$' Code (Missing Character)

Host		Controller
Sends a **Command Message**.	-->	Cancels the message.
	'$'Garbage Character	(Start mark not identified)
<Response time-out> Resend the **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).

2)   Garbage Character for <CR> Code (Missing Character)

Host		Controller
Sends a **Command Message**.	-->	Cancels the message.
	Garbage character of <CR>	(End mark not identified, time-out between characters)
<Response time-out> Resend the **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).

3) Garbage Character Other Than the Above Two (Missing Character)
   The Controller detects the checksum error if any errors are found with characters in a commnad message other than '$' and <CR> codes.

Host		Controller
Sends a **Command Message**.	--> Garbage character other than '$' and <CR> codes	Cancels the message. (Checksum error)
	<--	Sends **Communication Error Message**.
Resend the **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).

5-5-2 When an Error Occurs in a **Response Message** from the Controller

The same communication error as decribed above (5-5-1) may also occur in a **Response Message** from the Controller. Communications errors identified by the Host should not be informed back to the Controller.

If a command sent from the Host is an execution command, the Controller begins processing the command as soon as it is received. The Controller will not identify any communication error in its the reponse message to the Host. If the **Response Message** contains a communication error, the Host will resend the command to the Controller; however, the Unit Status will be busy (executing original command). The Controller will respond to the resent message with an error message ("execution invalid error").   Once the command has been executed, an **Execution Complete Message** will be sent before the response time-out. (This case is not shown below.)

1) Garbage Character of '$' or <CR> Codes (Missing Character)

Host		Controller
Sends a **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).
	Garbage character of '$' or <CR>	
<Response time-out>		
Resends **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).

2) Garbage Character Other Than Above (Missing Character)
    This is for the case when no **Execution Complete Message** is sent to the Host.

Host		Controller
Sends a **Command Message**.	-->	
Deletes message.	<--	Sends the **Response Message** (normal).
(Checksum error)	Garbage character other than '$' or <CR>	
Resends **Command Message**.	-->	
	<--	Sends the **Response Message** (normal).

5-5-3 Error Occurs in the **Execution Complete Message**

When a communication error occurs in the **Execution Complete Message** from the Controller to the Host, the Host does not have to perform any special processing (verification reporting) to the Controller. The Controller will resend the **Execution Complete Message** once the response time-out (for acknowledgement) is detected.

Host		Controller
	<--	Sends an **Execution Complete Message**.
	Any garbage character	
		(Response time-out)
	<--	Resends the **Execution Complete Message**.
Sends **Execution Complete Acknowledgement**	-->	

5-5-4 Error Occurs in the **Execution Complete Acknowledgement**

When a communications error occurs in an acknowledgement message from the Host, the Controller detects the response time-out for that message and resends the **Execution Complete Message**.

Host		Controller
	<--	Sends an **Execution Complete Message**.
Sends **Execution Complete Acknowledgement**	-->	
	Garbage character other than '$' or <CR>	
		(Response time-out)
	<--	Resends the **Execution Complete Message**.
Resends **Execution Complete Acknowledgement**	-->	

5-5-5 List of Processing at Communications Errors

The following lists the processing of the Host and the Controller for each error occurrence.

❑ Host Processing

	Garbage Character of '$' or ⊥ (or Missing Character)	Garbage Character of \<CR\> (or Missing Character)	Other Garbage Character (or Missing Character)
Error in a **Command Message** (Host → Controller)	Detects response time-out.	Detects response time-out.	Host will receive checksum error response.
	Sends the command again.	Sends the command again.	Sends the command again.
Error in a **Response Message** (Controller → Host)	Detects response time-out.	Detects response time-out.	Identifies the checksum error.
	Sends the command again.	Sends the command again.	Sends the command again.
Error in an **Execution Complete Message** (Controller → Host)	(The Controller resends the **Execution Complete Message**.)	(The Controller resends the **Execution Complete Message**.)	(The Controller resends the **Execution Complete Message**.)
	No processing	No processing	No processing
Error in an **Execution Complete Acknowledgement** (Host → Controller)	(The Controller resends the **Execution Complete Message**.)	(The Controller resends the **Execution Complete Message**.)	Host will receive checksum error response.
	Resends the **Execution Complete Acknowledgement**.	Resends the **Execution Complete Acknowledgement**.	Resends the **Execution Complete Acknowledgement**.

Note: When the Host sends a reference command during processing of an execution command and a communications error occurs in the reference command's **Response Message** from the Controller, the Host may not be able to identify whether the error occurs in the reference command's **Response Message** or in the **Execution Complete Message**. In this case, the Host should regard it as the former and send the latest command again.

❑ Controller Processing

	Garbage Character of '$' or '!' (or Missing Character)	Garbage Character of <CR> (or Missing Character)	Other Garbage Character (or Missing Character)
Error in a **Command Message** (Host → Controller)	Deletes the message.	Deletes the message. Detects time-out between characters.	Identifies the checksum error.
	No processing	No processing	Sends a checksum error response to the Host.
Error in a **Response Message** (Controller → Host)	(The Host detects the response time-out.)	(The Host detects the response time-out.)	(The Host identifies a checksum error.)
	No processing	No processing	No processing
Error in **Execution Complete Message** (Controller → Host)	Detects the response time-out for an acknowledgement message.	Detects response time-out for an acknowledgement message.	Detects the response time-out for an acknowledgement message.
	Resends the **Execution Complete Message**.	Resends the **Execution Complete Message**.	Resends the **Execution Complete Message**.
Error in **Execution Complete Acknowledgement** (Host → Controller)	Detects the response time-out for an acknowledgement message.	Detects time-out between characters. Detects the response time-out for an acknowledgement message.	Identifies the checksum error. Detects the response time-out for an acknowledgement message.
	Resends the **Execution Complete Message**.	Resends the **Execution Complete Message**.	Sends a checksum error response.

6. Related Functions

6-1 Position Data and Position Types

6-1-1 Transfer Station

For wafer transfer positions (GET/PUT points), the following stages need to be set in the Controller.

1) Loadport/Cassette Stage

   A transfer station with multi slots for wafers (e.g. open-cassette, FOUP, multi-slot stage, etc). The lowest slot position is registered. Other slot positions are calculated by slot pitch. Slot pitch is defined by the parameter.

2) Transfer Stage

   A single slot stage is assumed.

3) Pre-aligner Stage (Yaskawa Pre-aligner)

   The alignment motion of 300mm Wafer and 200mm Wafer is enabled by moving Stage.

   Teaching for the manipulator is performed depending on the stage which has been moved to the position according to the wafer size.

6-1-2 Types of Teaching Positions

1) Lowest Slot Position (STA position)

   The lowest slot position is registered for loadports/cassette stages. The teaching point in vertical (hereafter referred to as Z) direction is the lower surface of the wafer at that position.

2) Mapping Position

   For manipulators that are equipped with a mapping function (wafer presence/state in a FOUP/Cassette), the position where the mapping sensor detects the first wafer is taught as the mapping positon

3) From Ready Position (RDY position) to Minimum Sweep Position (MIN position)

   It is necessary to register the route from the ready position to the minimum sweep position to access each station.

   RDY   Ready position that is generated automatically by automatic ready position generation.

   MIN   Minimum sweep position. It is pre-determined position and preset as default.

   IM1   Intermediate position to prevent interference with manipulator and when moving from RDY to MIN. It is pre-determined position and preset as default.

   IM2   Same as above. It is pre-determined position and preset as default.

   IM3   Same as above. It is pre-determined position and preset as default.

   Refer to Teach Manual for Wafer Transfer Manipulator (Z524i) : HE0401335 for Position mentioned above.

Related Parameters

・Number of via point parameter
・End-Effector swappable position parameter
・Automatic ready position generation parameter
・Stroke between STA and RDY position for automatic ready position generation parameter
・Angle between End-Effector 1 and 2 for automatic End-Effector position generation parameter

---

6-1-3 Number of Registration Position Data Items

The following table lists the current number of positions and types that can taught to the unit.

Transfer Station		Registration Positions						
		Lowest-layer Position STA	Ready Position RDY	IM3	IM2	IM1	Minimum Sweep Position MIN	Mapping Position
Cassette Stage (8 sets)	Cassette stage 1 (P1)	YES	YES	-	-	-	YES	YES
	Cassette stage 2 (P2)	YES	YES	-	-	-	YES	YES
	Cassette stage 3 (P3)	YES	YES	-	-	-	YES	YES
	Cassette stage 4 (P4)	YES	YES	-	-	-	YES	YES
	Cassette stage 5 (P5)	YES	YES	-	-	-	YES	YES
	Cassette stage 6 (P6)	YES	YES	-	-	-	YES	YES
	Cassette stage 7 (P7)	YES	YES	-	-	-	YES	YES
	Cassette stage 8 (P8)	YES	YES	-	-	-	YES	YES
Transfer Stage (12 sets)	Transfer stage 1 (UA)	YES	YES	-	-	-	YES	-
	Transfer stage 2 (UB)	YES	YES	-	-	-	YES	-
	Transfer stage 3 (UC)	YES	YES	-	-	YES	YES	-
	Transfer stage 4 (UD)	YES	YES	-	-	-	YES	-
	Transfer stage 5 (UE)	YES	YES	-	YES	YES	YES	-
	Transfer stage 6 (UF)	YES	YES	-	-	-	YES	-
	Transfer stage 7 (UG)	YES	YES	-	-	-	YES	-
	Transfer stage 8 (UH)	YES	YES	-	-	-	YES	-
	Transfer stage 9 (UI)	YES	YES	-	-	-	YES	-
	Transfer stage 10 (UJ)	YES	YES	-	-	-	YES	-
	Transfer stage 11 (UK)	YES	YES	-	-	-	YES	-
	Transfer stage 12 (UL)	YES	YES	-	-	-	YES	-

6-2 Other Positions

Other than the positions explained in "6-1 Position Data and Position Type", the following positions are registered in the Controller. The manipulator can be moved to each position by the Host or teaching pendant (hereinafter called "TP").

6-2-1 HOME Position

HOME position can be changed to any desired position by setting the appropriate position. The motion commnad **MHOM** will move the manipulator to this position.
The default setting for HOME position is Manipulator parameter R355 – R359.

6-2-2 Ready Position (RDY positin)

The Ready position is defined as the postion where the arm extension motion to a stage/loadport begins – for GET motion; or the position where arm extension motion from a stage/loadport has finished – PUT.

The Ready position is the final position for the following motion commands: MTRS, MGET, MPUT, MGT2, MPT2, MGTW, MPTW, MGWI, MPWI, MSWP,MSWI, MSP2, MXTW and MXWI.

For the motion path, refer to "6-5 Workpiece Transfer Motion Path."

6-2-3 MWRM Position

The motion commnad **MWRM** will move the manipulator to this position.

The default setting for MWRM position(Ready,B1,B2) is Manipulator parameter R450 – R459.

6-3   Slot Pitch and Slot Position

6-3-1 Slot Pitch

The slot pitch can be set individually for each loadport station. The slot pitch can be saved in volatile memory or non-volatile memory depending if the setting is temporary or permanent. If the setting is saved in volatile memory, the slot pitch will change to the previously permanently set value after rebooting the controller.

6-3-2 Slot Position

The manipulator can move to any arbitrary slot position for a loadport station by calculating the specified slot's position. This calculation is based on the defined slot pitch, the calculated Z distance to the specified slot and the lowest slot position data.

The maximum number of the cassette stage slots can be set by parameter. This prevents interference between the manipulator and the peripheral devices, which may result from improper user operation.

+---------------------------------+
| **Related Parameters** |

・Slot pitch parameter
・Maximum number of slots

6-4   Automatic Speed Changing Function

The following speeds are available for motion commands sen tfrom the Host. Furthermore, these speeds can be used for all motion commands.

The speed setting parameter is setup as a percent of a set maximum speed. The setting is saved in volatile memory, therefore rebooting the controller will cause the default value (saved in non-volatile memory) to be set.

Transfer Speed 1:   Motion speed when no wafer is being handled

Transfer Speed 2:   Motion speed when a wafer is being handled

Low speed: Motion speed for lifting wafer from stage/loadport
    (MGET, MPUT,MPNT, MGTW, MPTW, MGWI,
    MPWI,MSWP, MSWI ,MGT2,MPT2,MSP2,MXTW,MXWI);
     HOME motion (MHOM) speed

The Controller will automatically switch between Transfer Speed 1 and Transfer Speed 2 based on the status of the grip sensor and motion pattarn.

Related Parameters

・Host transfer speed setting parameter

6-5 Wafer GET/PUT Motion Pattern

Wafer GET's and PUT's are carried out by the manipulator moving between robot generated Transfer Points. Transfer points are calculated based on the taught position of a stage/loadport and the respective transfer offsets (set by parameter).   The transfer offset is available at each transfer station and any value can be set by the Host or T.P.

Transfer Point differ depending on the type of manipulator End-Effector. The motion pattarns for the different types of End-Effectors is described in this section.

The following types of End-Effectors can be provided:

- Vacuum End-Effector: Holds the wafer by vacuum suction.

- Active Edge Grip End-   A force is applied to the wafer to maintain its
  Effector: position during transfer

6-5-1 Transfer Offset

As mentioned above, transfer offset parameters are used inconjunction with teaching positions to generate the transfer points. The following transfer offsets are available at each stage/loadport:

Related Parameters

- ・Upward offset parameter
- ・Downward offset parameter
- ・Extending direction offset parameter
- ・Contracting direction offset parameter
- ・Downward offset for Put parameter
- ・Contracting direction offset for GET parameter
- ・Extending direction offset for GET parameter
- ・Upward offset for GET parameter
- ・Offset for PUT parameter

6-5-1 Wafer GET Motion Pattern for Vacuum End-Effector (MGET,MGT2,MSP2)

The following diagram shows a Wafer GET motion pattern (MGET,MGT2,MSP2) for a vacuum End-Effector.
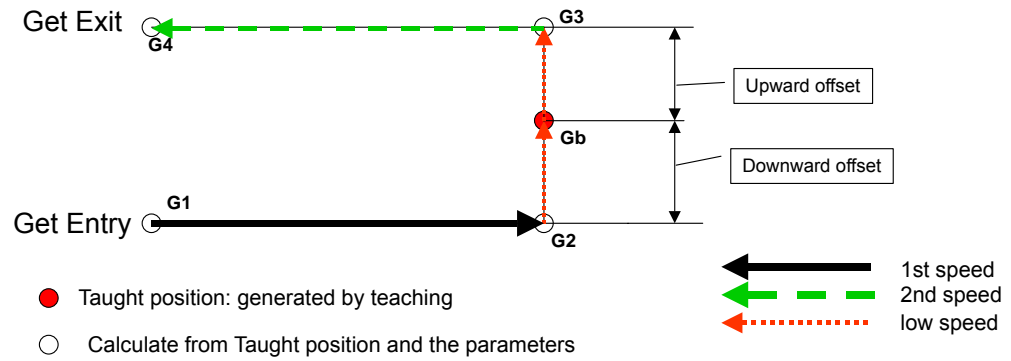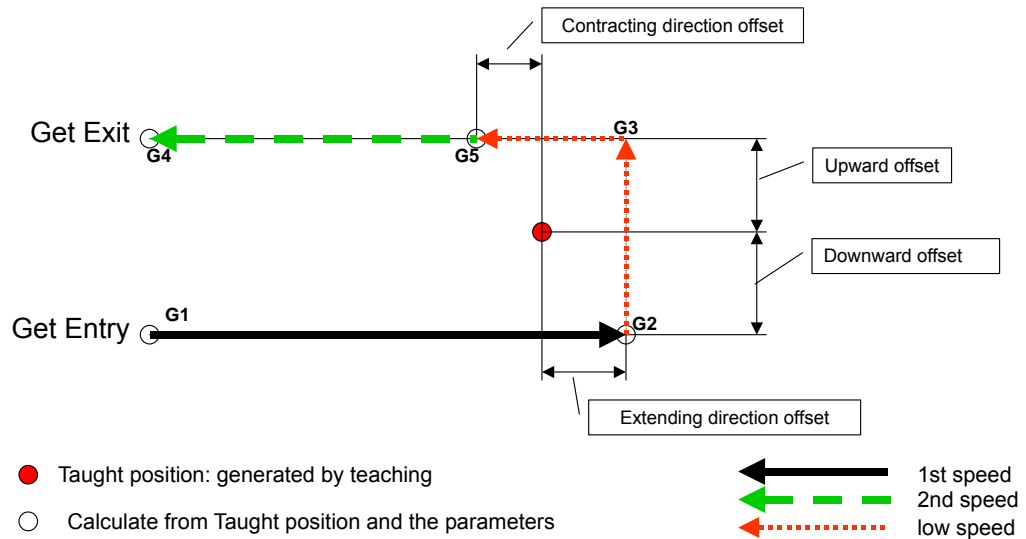

Fig. 2 Wafer GET Motion Pattern (MGET,MGT2,MSP2) for Vacuum End-Effector

Wafer GET Motion

(Wafer absence) → G1 (GET ready position) → G2 → (Solenoid valve ON) → Gb → G3 → (Wafer presence) → G4

6-5-2 Wafer GET Motion Pattern for Vacuum End-Effector

(MGTW,MGWI,MSWP,MSWI,MXTW,MXWI)

The following diagram shows a Wafer GET motion pattern
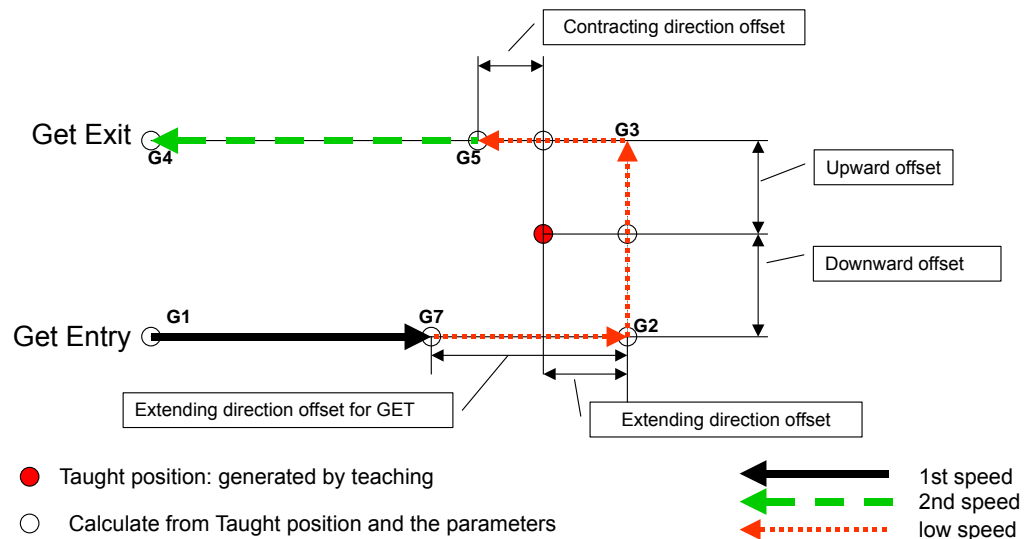(MGTW,MGWI,MSWP,MSWI,MXTW,MXWI) for a vacuum End-Effector.
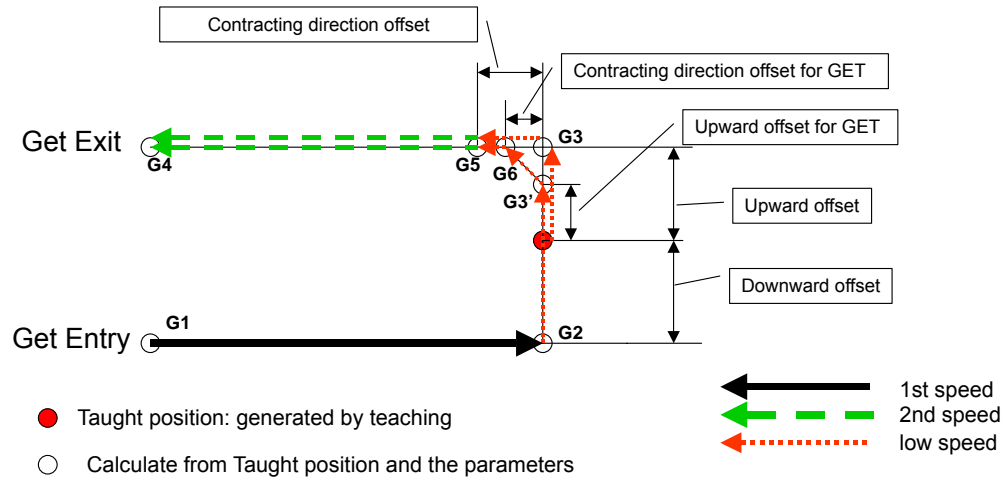
Fig. 3 Wafer GET Motion Pattern (MGTW,MGWI,MSWP,MSWI,MXTW,MXWI) for Vacuum
End-Effector

1) Wafer GET Motion (Trapenzoidal)

(Wafer absence) → G1 (GET ready position) → G2 → (Solenoid valve
ON) → G3' → (Wafer presence) → G6 → G5 → G4

2) Wafer GET Motion (Rectangular)

(Wafer absence) → G1 (GET ready position) → G2 → (Solenoid valve
ON) → G3 → (Wafer presence) → G5 → G4

6-5-3 Wafer PUT Motion Pattern for Vacuum End-Effector
(MPUT,MPT2,MSP2)

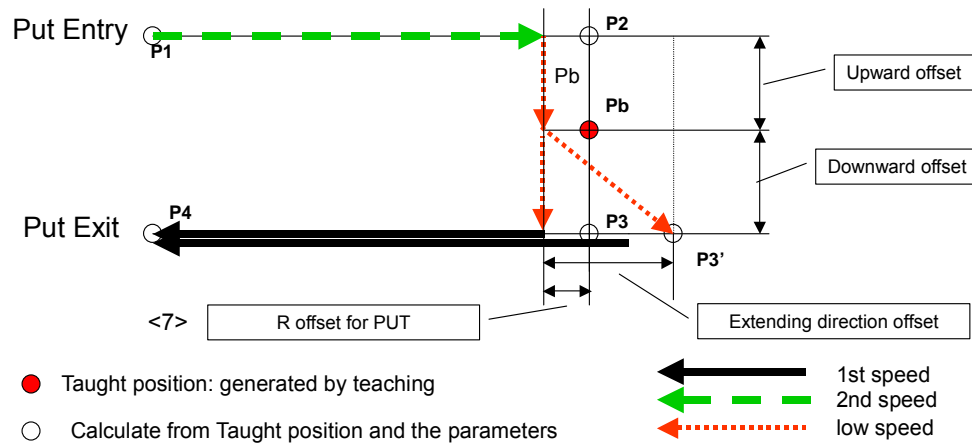The following diagram shows a Wafer PUT motion pattern (MPUT,MPT2,MSP2) for a vacuum End-Effector.
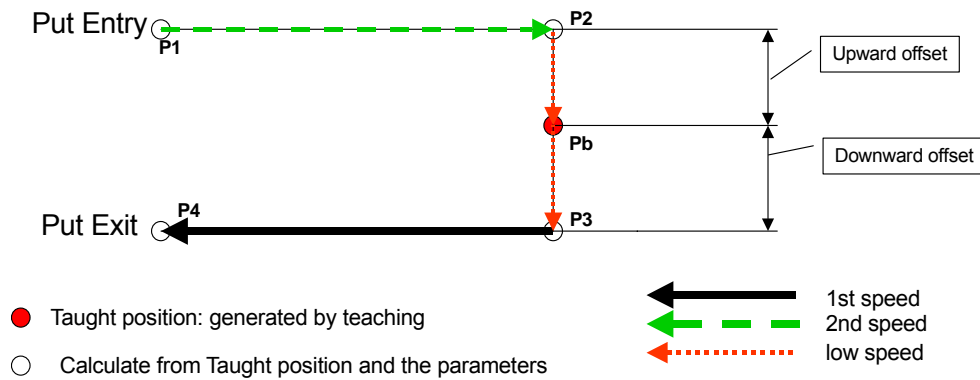
Fig. 4 Wafer PUT Motion Pattern (MPUT,MPT2,MSP2) for Vacuum End-Effector

Wafer PUT Motion

(Wafer presence) → P1 (Put ready position) → P2 → (Solenoid valve OFF + Wafer absence) → Pb → P3 (Wafer absence) → P4

6-5-4   Wafer PUT Motion Pattern for Vacuum End-Effector

(MPTW,MPWI,MSWP,MSWI,MXTW,MXWI)

The following diagram shows a Wafer PUT motion pattern
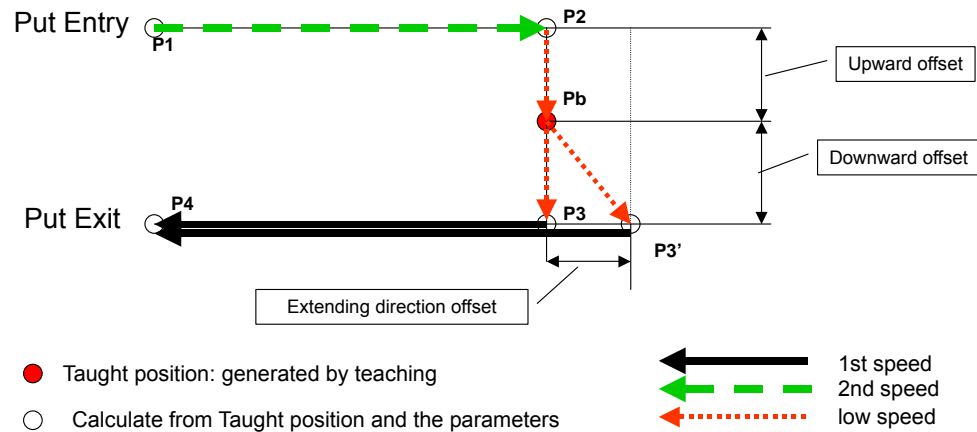(MPTW,MPWI,MSWP,MSWI,MXTW,MXWI) for a vacuum End-Effector.
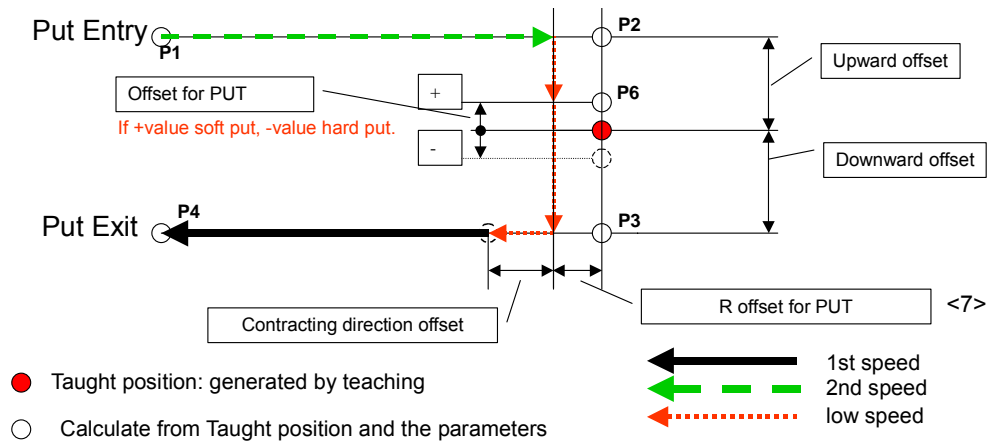
Fig. 5 Wafer PUT Motion Pattern (MPTW,MPWI,MSWP,MSWI,MXTW,MXWI) for Vacuum End-Effector

Wafer PUT Motion

(Wafer presence) → P1 (Put ready position) → P2 → P6 (Solenoid valve OFF)
→ P3 (Wafer absence) → P4

6-5-5   Wafer GET Motion Pattern for Edge Grip End-Effector
(MGET,MGT2,MSP2)

The following diagram shows a Wafer GET motion pattern (MGET,MGT2,MSP2) for an edge grip End-Effector.

Fig. 6 Wafer GET Motion Pattern (MGET,MGT2,MSP2) for Edge Grip End-Effector

Wafer GET Motion

Manipulator parameter i070=1

> (Wafer absence) → G1 (GET ready position) → (Solenoid valve OFF) → G2→G3 → G5 (Solenoid valve ON + Wafer presence*) → G4

Manipulator parameter i070=0

> (Wafer absence) → G1 (GET ready position) → (Solenoid valve OFF) → G2→G3 → (Solenoid valve ON) → G4 (Wafer presence*)

6-5-6 Wafer GET Motion Pattern for Edge Grip End-Effector (MGTW,MGWI,MSWP,MSWI,MXTW,MXWI)

The following diagram shows a Wafer GET motion pattern
(MGTW,MGWI,MSWP,MSWI,MXTW,MXWI) for
an edge grip End-Effector.

Fig. 7 Wafer GET Motion Pattern (MGTW,MGWI,MSWP,MSWI,MXTW,MXWI) for Edge Grip End-Effector

Wafer GET Motion

(Wafer absence) → G1 (GET ready position) → (Solenoid valve OFF) →
G7→G2 → G3 → G5 (Solenoid valve ON + Wafer presence) → G4

6-5-7 Wafer PUT Motion Pattern for Edge Grip End-Effector
(MPUT,MPT2,MSP2)

The following diagram shows a Wafer PUT motion pattern (MPUT,MPT2,MSP2) for an edge grip End-Effector.

Put motion for Edge grip EE (Cassette/Scan stage)

Fig. 8 Wafer PUT Motion Pattern (MPUT,MPT2,MSP2) for Edge Grip End-Effector

Wafer PUT Motion

(Wafer presence) → P1 (Put ready position) → P2 → (Solenoid valve OFF + Wafer absence) → Pb → (P5) → P3 (P3') → P4

6-5-8 Wafer PUT Motion Pattern for Edge Grip End-Effector
(MPTW,MPWI,MSWP,MSWI,MXTW,MXWI)

The following diagram shows a Wafer PUT motion pattern
(MPTW,MPWI,MSWP,MSWI,MXTW,MXWI) for
an edge grip End-Effector.

Fig. 9 Wafer PUT Motion Pattern (MPTW,MPWI,MSWP,MSWI,MXTW,MXWI) for Edge Grip End-Effector

Wafer PUT Motion

(wafer presence) → P1 (Put ready position) → P2 → (Solenoid valve
OFF + wafer absence) → Pb → P3 (P3') → P4

6-5 Mapping Function

Mapping function specifies whether the wafer is present at the loadport. The mapping function can identify: Presence, Double Slot or Cross Slot. The manipulator has different postures depending on the mounting position of the mapping sensor, etc. (See the following diagram.) The mapping function is not available for manipulators without the a vertical axis.
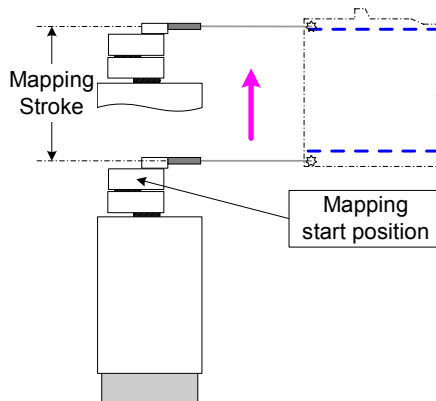
Fig. 4 Example of Mapping Operation

6-6-1 Mapping in Cassette Stage

For mapping at the cassette stage, the vertical axis rises from registered mapping start position by the mapping stroke set by parameter. Wafer presence, double slot and cross slot will be checked for each slot position.

> **Related Parameters**
>
> · Cassette stage pitch distance between slots
> · Cassette stage distance of registration position from mapping starting position
> · Cassette stage Mapping start position height
> · Cassette stage Mapping end position height
> · Cassette stage number of slot

6-7 Alignment Function

With the alignment function, positioning of the center of the wafer and the notch or orientation flat are performed. During alignment calibration, the refernce angle for the notch or orientation flat should be specified. Alignment calibration can be carried out by the Host or T.P. The setting corresponding to each P/A stage can be made so that the refernce angle can be specified at alignment.

> **Related Parameters**
>
> · Alignment mode
> · Alignment positioning mode

6-8 Master Registration (Teaching) Function

When a position registration is performed at a transfer station, other manipulator positions for the same station can be automatically calculated and registered using the master registration (teaching) function. This is used to teach both End-Effector (B1 and B2) positions when only teaching one End-Effector.

> Note: Due to assembly errors of peripheral devices, etc., fine adjustment may be necessary after confirming the automatically calculated position.

Registered Position	Position where Master Registration is Available
Lowest slot position	Lowest slot position of the other End-Effector

6-9 "AUTO"/"TEACH" Mode

During normal tool uptime "AUTO" mode is used for the Host to communicate with the Controller. While "AUTO" mode is enabled the teach pendant cannot be used.
For manually teaching positions (i.e., via the Teach Pendant), the controller must be switched to "TEACH" mode. "TEACH" mode is also used during maintenance time.

The switch on the Teach Pendant changes modes from "AUTO" to "TEACH".

6-10 ISYS/MHOM Function

6-10-1 Purpose of ISYS/MHOM

The purpose of ISYS/MHOM is automatic recovery during a sequenced operation .
ISYS for Manipulator will:

1. Clear the alarm
2. Servo Power ON
3. Move back to minimum sweep position

ISYS for Pre-aligner will:

1. Move back to Ready position
2. Alignment (only when wafer presence)

MHOM will:

1. Move back to minimum sweep position/ home position

6-10-2 Recoverable Position

It is possible to move back to safe posture from the following positions.

- Positions inside the Recoverable Area and close to a Minimum Sweep Position
  (See 6-10-5-1)
- Positions between a Ready Position and a Station Position (See 6-10-5-2)
- Positions between a Minimum Sweep Position and a Ready Position (See 6-10-5-3)

Above condition works for:

. Recovery from Teach Point
. Recovery from Position halted during a sequenced transfer
. Recovery from Position emergency stopped during a sequenced transfer

6-10-3 Exceptions to ISYS for Manipulator/MHOM (Non-recoverable)

It is not possible to move back to safe posture from the following positions.

The Manipulator has moved away from teach point.

- Relative motion with Host and Teach Pendant
- Absolute motion with Host and Teach Pendant
- JOG operation with Teach Pendant

In the above-mentioned case, requires manual recovery.

ex. Expansin axis:-250deg, End-Effector 1 axis:0deg, End-Effector 2 axis:0deg

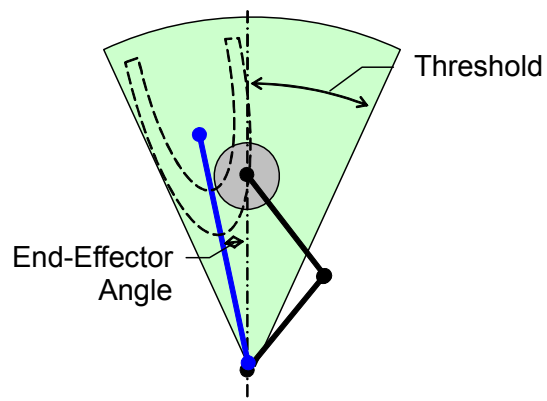Please move manually to recoverable posture (See 6-10-4) and initiate ISYS/MHOM
command.

6-10-4 Recoverable Area/Posture

The recoverable posture is posture that satisfies the following conditions.

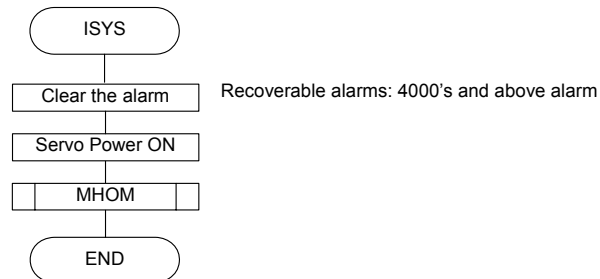. Wafer center is inside the recoverable area.
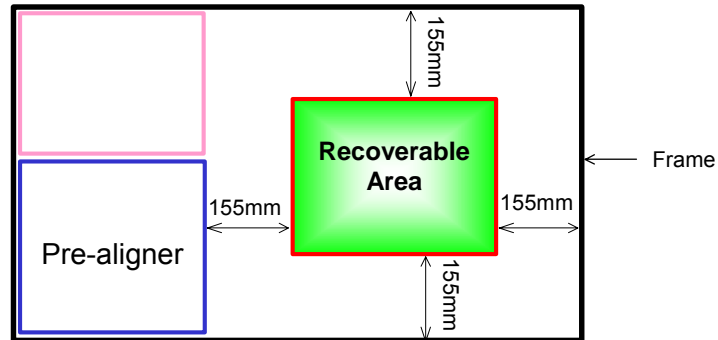
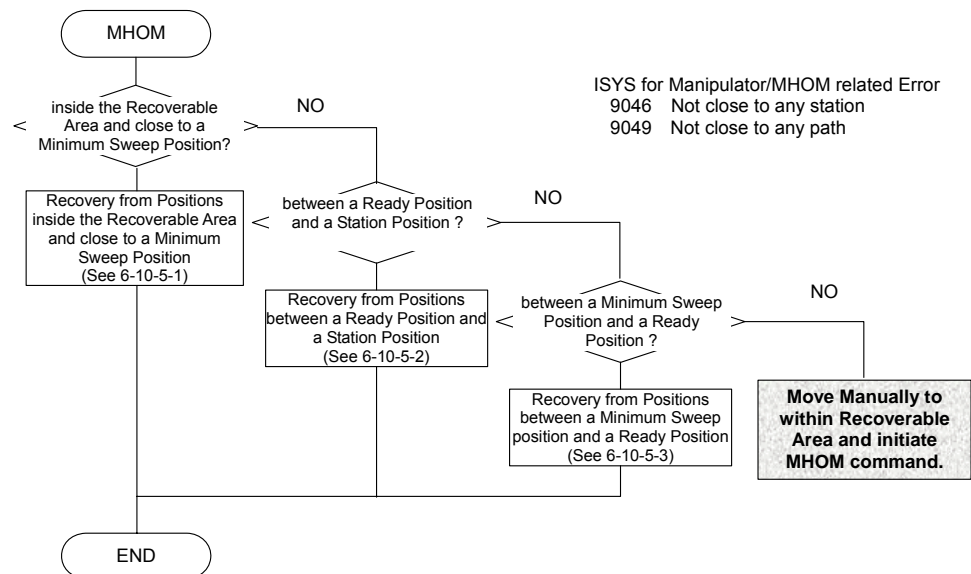. End-Effector angle is smaller than threshold value (default is 20 degrees)

6-10-5 ISYS for Manipulator/MHOM flow

The outline flow of ISYS for Manipulator is shown below.

```
        ┌─────────┐
        │  ISYS   │
        └────┬────┘
    ┌────────┴────────┐
    │ Clear the alarm │   Recoverable alarms: 4000's and above alarm
    └────────┬────────┘
    ┌────────┴────────┐
    │  Servo Power ON │
    └────────┬────────┘
   ┌─┬───────┴───────┬─┐
   │ │     MHOM      │ │
   └─┴───────┬───────┴─┘
        ┌────┴────┐
        │   END   │
        └─────────┘
```

The outline flow of MHOM is shown below.

ISYS for Manipulator/MHOM related Error
9046   Not close to any station
9049   Not close to any path

inside the Recoverable Area and close to a Minimum Sweep Position?

Recovery from Positions inside the Recoverable Area and close to a Minimum Sweep Position (See 6-10-5-1)

NO

between a Ready Position and a Station Position ?

Recovery from Positions between a Ready Position and a Station Position (See 6-10-5-2)

NO

between a Minimum Sweep Position and a Ready Position ?

Recovery from Positions between a Minimum Sweep position and a Ready Position (See 6-10-5-3)

NO

Move Manually to within Recoverable Area and initiate MHOM command.

6-10-5-1 Recovery from Positions inside the Recoverable Area
and close to a Minimum Sweep Position

It is determined that the current position is close to a Minimum Sweep Posture, if the current position satisfies the following conditions.

When End-Effector angle is smaller than threshold value
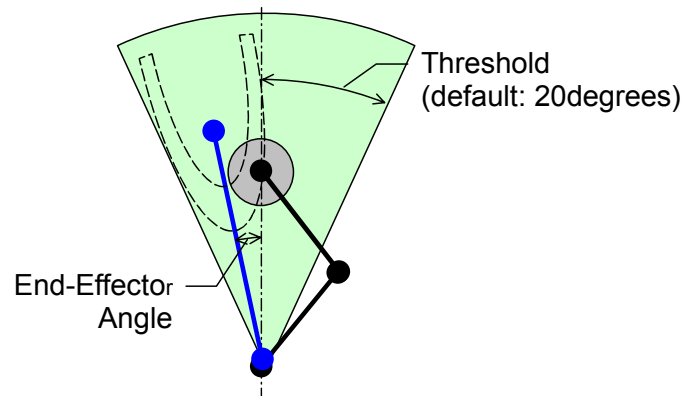
When End-Effector angle is smaller than threshold value

If extension is extended rather than minimum sweep position, retract arm first and fold the End-Effector.

If extension is retracted more than minimum sweep position, fold the End-Effector first and move extension axis to minimum sweep position.

When End-Effector angle is larger than than threshold value
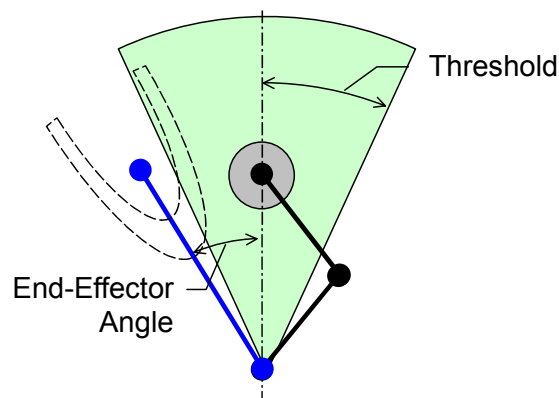
Go to "Recovery from Positions between a Minimum Sweep Position and a Ready Position".

6-10-5-2 Recovery from Positions between a Ready Position and a Station Position

It is determined that the current position is close to positions between a Ready Position and a Station Position, if the current position satisfies the following conditions.
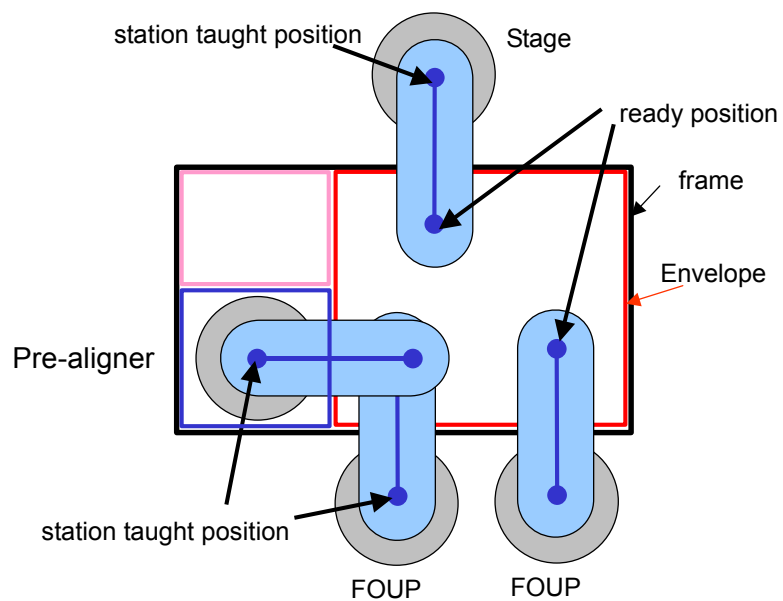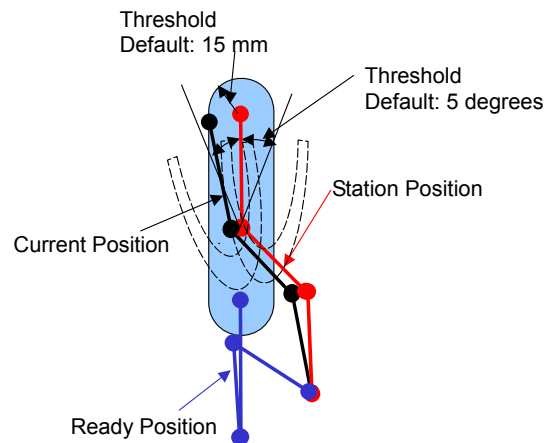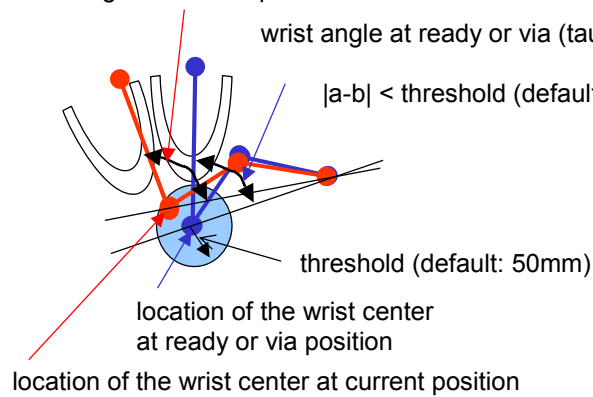
6-10-5-3 Recovery from Positions between a Minimum Sweep Position and a Ready Position

It is determined that the current position is close to positions between a Minimum Sweep Position and a Ready Position, if the current position satisfies the following conditions.

wrist angle at current pos = a

wrist angle at ready or via (taught pos) = b

$|a-b|$ < threshold (default: 5 degrees)

threshold (default: 50mm)

location of the wrist center
at ready or via position

location of the wrist center at current position

B

current position

A

position on the path

a

$|a-b|$ < threshold (default: 5 degrees)
$|A-B|$ < threshold (default: 5 degrees)

b
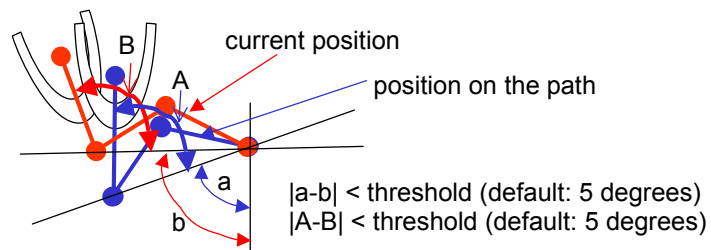
6-10-5-4 Z-axis Recovery
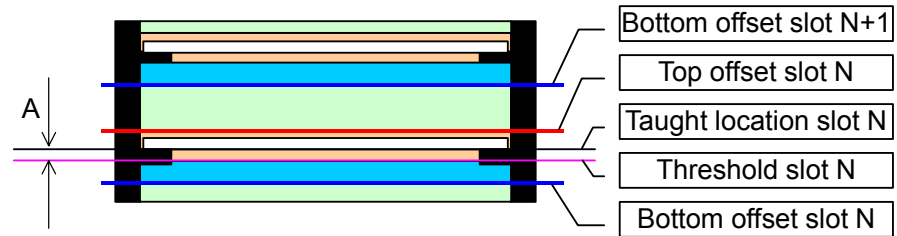
Top offset slot N ~ Bottom offset slot N+1: Leave height and pull out.
Threshold slot N ~ Top offset slot N: Move up to top offset slot N and pull out.
Bottom offset slot N ~ Threshold slot N: Move to bottom offset slot N and pull out.
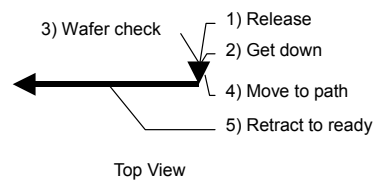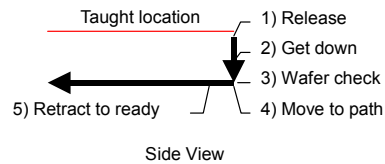
Value of 'A' (default: -1mm) can be changed.
Note: We recommend -1mm as value of 'A' , because manipulator locates in lower
slot and does not grip wafer locating in upper slot. In this case robot can be
get down safely.
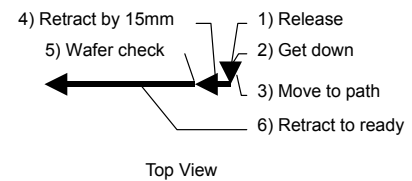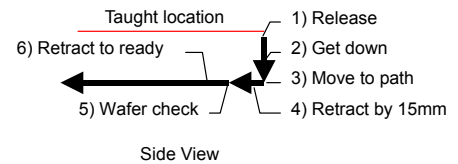
6-10-5-5 Recovery trajectory and chucking control

Recovery trajectory and chucking control is shown below.
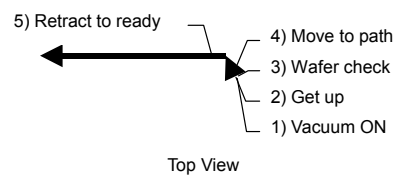
* Case of current Z is lower than Taught Z - 1mm

< Vacuum >

Taught location
1) Release
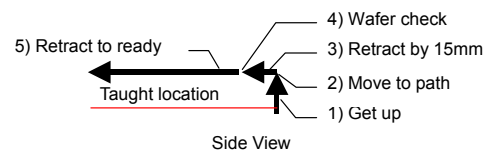2) Get down
3) Wafer check
5) Retract to ready
4) Move to path

Side View

< Edge grip >

Taught location
1) Release
6) Retract to ready
2) Get down
3) Move to path
5) Wafer check
4) Retract by 15mm

Side View

3) Wafer check
1) Release
2) Get down
4) Move to path
5) Retract to ready

Top View

4) Retract by 15mm
1) Release
5) Wafer check
2) Get down
3) Move to path
6) Retract to ready

Top View

* Case of current Z is higher than Taught Z - 1mm

< Vacuum >

5) Retract to ready
4) Move to path
3) Wafer check
Taught location
2) Get up
1) Vacuum ON

Side View

< Edge gip >

4) Wafer check
5) Retract to ready
3) Retract by 15mm
Taught location
2) Move to path
1) Get up

Side View

5) Retract to ready
4) Move to path
3) Wafer check
2) Get up
1) Vacuum ON

Top View

4) Wafer check
5) Retract to ready
3) Retract by 15mm
2) Move to path
1) Get up

Top View
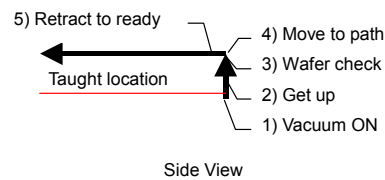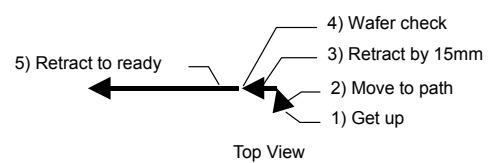
7   Command Lists

7-1   Motion Commands

	Command	Name	Manipultor	Pre-aligner	Macro Modification
1	MHOM	Move to HOME position	Available	-	Available
2	MTRS	Move to Ready position	Available	Available	Available
3	MGET	Workpiece Get motion	Available	-	Available
4	MPUT	Workpiece Put motion	Available	-	Available
5	MGT2	Workpiece Get motion (MTRS + MGET)	Available	-	Available
6	MPT2	Workpiece Put motion (MTRS + MPUT)	Available	-	Available
7	MSP2	Workpiece Swap motion (MGT2 + MPT2)	Available	-	Available
8	MPNT	Motion between transfer points	Available	-	Available
9	MMAP	Mapping	Available	-	Available
10	MALN	Alignment	-	Available	Available
11	MTCH	Move to registered position	Available	-	Available
12	MABS	Move to specified coordinate position	Available	Available	Available
13	MRLK	Move to specified relative position (LNK)	Available	Available	Available
14	MRLN	Move to specified relative position (LNR)	Available	-	Available
15	MACA	Alignment calibration	-	Available	Available
16	MMCA	Mapping calibration	Available	-	Available
17	MGTW	Workpiece Get Motion	Available	-	Available
18	MPTW	Workpiece Put Motion	Available	-	Available
19	MGWI	Workpiece Get Motion	Available	-	Available
20	MPWI	Workpiece Put Motion	Available	-	Available
21	MSWP	Workpiece Swap Motion	Available	-	Available
22	MSWI	Workpiece Swap Motion	Available	-	Available
23	MXTW	Workpiece Exchange Motion	Available	-	Available
24	MXWI	Workpiece Exchange Motion	Available	-	Available
24a	MCDT	Collision detection threshold calibration	Available	-	Available
25	ISYS	System Inisialization	Available	Available	Available
26	MWRM	Warm up opration	Available	-	Available

<7>

7-2   Control Commands

	Command	Name	Manipulator	Pre-aligner	Macro modification
27	CHLT	Motion interruption	Available	-	Available
28	CRSM	Restart from interruption	Available	-	Available
29	CEMG	Deceleration to a stop and servo OFF	Available	Available	Available
30	CSRV	Servo command	Available	Available	Available
31	CCLR	Error clear	Available	Available	Available
32	CSOL	Solenoid valve Control Command	Available	Available	Available
33	CCHK	Chucking Control Command	Available	Available	Available
34	CLFT	Lifter control command	-	Available	Available

7-3   Setting Commands

	Command	Name	Manipulator	Pre-aligner	Macro Modification
35	SSPP	Motion speed setting (in [%])	Available	Available	Available
36	SPOS	Registration of current position	Available	-	Available
37	SABS	Registration of coordinate position	Available	-	Available
38	SPSV	Position data save	Available	-	Available
39	SOFS	Transfer offset setting	Available	-	Available
40	SPIT	Setting of pitch between slots	Available	-	Available
41	SSLT	Setting of number of slots	Available	-	Available
42	SRSV	Parameter save	Available	-	Available
43	SMSK	Setting of interlock monitoring enabled/disabled	Available	Available	Available
44	SPRM *	Setting of parameter	Available	Available	-
45	SALM	Setting alignment mode	-	Available	Available
46	SSTD *	Setting standard position	Available	Available	-
47	SWSZ	Setting wafer size	Available	Available	Available

Note)   * User cannot change SPRM and SSTD macro.

7-4 Reference Commands

	Command	Name	Manipulator	Pre-aligner	Macro modification
48	RSPP	Reference motion speed	Available	Available	Available
49	RPOS	Reference current position	Available	Available	Available
50	ROFS	Reference Transfer offset	Available	-	Available
51	RCST	Reference information on cassette stage	Available	-	Available
52	RMAP	Reference mapping result	Available	-	Available
53	RSTS	Reference various statuses	Available	Available	Available
54	RERR	Reference error history	Available	Available	Available
55	RMSK	Reference information on interlock monitoring	Available	Available	Available
56	RVER	Reference software version	Available		Available
57	RCFG	Reference configuration information	Available	-	Available
58	RSTT	Reference status information	Available	-	Available
59	RPRM *	Reference parameters	Available	Available	-
60	RCCD	Reference CCD data	-	Available	Available
61	RALM	Reference alignment mode	-	Available	Available
62	RSTD *	Reference standard position	Available	Available	-
63	RWSZ	Reference wafer size	Available	Available	Available
64	RIOS	Reference I/O status	Available	Available	-

Note)　　* User cannot change RPRM, RSTD and RIOS macro.

7-5 Execution Completion Acknowledgement Command

	Command	Name	Manipulator	Pre-aligner	Macro Modification
65	ACKN *	Acknowledgement of execution completion	Available	Available	-

Note)　　* User cannot change ACKN macro.

7-6 Maintenance Commands

	Commands	Name	Manipulator	Pre-aligner	Macro Modification
66	UPOS *	Registered position data uploading	Available	Available	-
67	UPRM *	Internal parameter uploading	Available	Available	-
68	DPOS *	Registered position data downloading	Available	Available	-
69	DPRM *	Internal parameter downloading	Available	Available	-

Note)　　* User cannot change UPOS, UPRM, DPOS and DPRM macro.

7-7   Reset Command

	Commands	Name	Manipulator	Pre-aligner	Macro modification
70	HRST	Software Reset Command	Available	-	Available

7-8   Macro Command

	Commands	Name	Manipulator	Pre-aligner	Macro modification
71	MACR	User macro communication command	Available	Available	Available

1. MHOM

[Name]   Move to HOME Position

[Function]   Moves each axis of the specified unit to the home position.

[Operation]   Each axis moves to the home position in the following order.
1. Arm (R and B axes) is moved to the home position.
2. If specified (i.e., Mmode is 'F') all other axes are moved to the designated Home Position.

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode

[Command Format]

$ <u>\<UNo></u> <u>MHOM</u> <u>\<MMode></u> <u>\<Sum></u> <u>\<CR></u>

· UNo: Unit number (1 byte)
    · '1':   Manipulator
· MMode: Motion mode (1 byte)
    · 'F':   All axes
    · 'A':   Expansion axis only

[Response Format]

(Response Message)

@ <u>\<UNo></u> <u>\<Sts></u> <u>\<Ackcd></u> <u>\<Subcd></u> <u>\<Sum></u> <u>\<CR></u>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <u>\<UNo></u> <u>\<Sts></u> <u>\<Errcd></u> <u>\<Subcd></u> <u>MHOM</u> <u>\<Sum></u> <u>\<CR></u>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

[Note]   MHOM is not possible from certain manipulator positions.

2. MTRS

[Name]   Move to Ready Position

[Function]   Moves to the ready position where the workpiece Get/Put operations are enabled. Creates a transfer point according to the specified transfer station and slot number (only when the cassette stage is specified). Moves to the alignment start position when a three-axis Pre-aligner is used.

[Operation]   <Manipulator>
        Each axis moves to the ready position in the following order.
        1. Identifies the workpiece state.
        2. Moves the arm to the ready position.
    <Three-axis Pre-aligner>
        Moves to the alignment start position .

[Conditions]   · The specified unit is stopped.
        · The specified unit is under servo ON state.
        · "Auto" mode
        · Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
    $ <UNo> MTRS <TrsSt> <SlotNo> <NextMtn> <Sum> <CR>
    · UNo: Unit number (1 byte)
      · '1' to '2': Unit specified
    · TrsSt: Transfer station (2 bytes)
      <Manipulator>
      · "P1" to "P8"   : When cassette stage specified
      · "UA" to "UL"   : When transfer stage specified
      <Three-axis Pre-aligner>
      · "G1": Workpiece Get position for 300mm wafer
      · "G2": Workpiece Get position for 200mm wafer
    · SlotNo: Slot number (2 bytes)
      <Manipulator>
      · "01" to "XX": When cassette stage specified by <TrsST>
        (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
      · "00": When transfer stage stage specified by <TrsSt>
      <Three-axis Pre-aligner>
      · "00": Fixed value
    · NextMtn: Next motion mode (2 bytes)
      <Manipulator>
      · "GA": Next motion is workpiece Get motion at End-Effector 1.
      · "PA": Next motion is workpiece Put motion at End-Effector 1.
      · "GB": Next motion is workpiece Get motion at End-Effector 2.
      · "PB": Next motion is workpiece Put motion at End-Effector 2.
      <Three-axis Pre-aligner>
      · "AL": Next motion is alignment (fixed value).

[Response Format]
    (Response Message)
      @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1' or '2': Unit specified
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
      · At normal reception: Responds "0000".
      · Reception rejected: Responds with the error code.
    · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
      · When either alarm does not occur, sub code is '0000'.

2.MTRS

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> MTRS <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1' or '2':   Unit specified
- Sts: Status (2 byte)
- Errcd: Error code (4 bytes)
  - At normal completion: Responds "0000".
  - At erroneous completion: Responds with the error code.
- Subcd: Sub code (4 bytes)
  - Detailed alarm information is shown.
  - When either alarm does not occur, sub code is '0000'.

3. MGET

[Name]   Workpiece Get Motion

[Function]   Performs the workpiece Get motion. With YASKAWA standard interlock.
Note: This command is used by combining with "MTRS".

[Operation]   Performs the workpiece Get motion for the transfer station specified by "MTRS" parameter that is located immediately before this command. For the motion sequence, refer to "6.5 Workpiece Transfer Motion Path".

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· The "MTRS" command immediately before this command has been completed normally.

[Command Format]
$ <UNo> MGET <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Manipulator

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · At normal reception: Responds "0000".
   · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MGET <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
   · At normal completion: Responds "0000".
   · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

4. MPUT

[Name]   Workpiece Put Motion

[Function]   Performs the workpiece Put motion. With YASKAWA standard interlock
Note: This command is used by combining with "MTRS."

[Operation]   Performs the workpiece Put motion for the transfer station specified by "MTRS" parameter immediately before this command.   For the motion sequence, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· The "MTRS" command immediately before this command has been completed normally.

[Command Format]
$ <UNo> MPUT <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Manipulator

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
· At normal reception: Responds "0000".
· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MPUT <Sum> <CR>

· Uno: Unit number (1 byte)
· '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
· At normal completion: Responds "0000".
· At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

5. MGT2

[Name]   Workpiese Get Motion

[Function]   This command is a combination command of "MTRSxxxxGx" and "MGET". This get motion trajectory is same as MGET. Performs the workpiece Get motion. With YASKAWA standard interlock.

[Operation]   Performs the workpiece Get motion for the transfer station specified.
With YASKAWA standard interlock.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
$ <UNo> MGT2 <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Manipulator
· TrsSt: Transfer station (2 bytes)
   <Manipulator>
   · "P1" to "P8"   : When cassette stage specified
   · "UA" to "UL"   : When transfer stage specified
· SlotNo: Slot number (2 bytes)
   <Manipulator>
   · "01" to "XX": When cassette stage specified by <TrsST>
     (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
   · "00": When transfer stage specified by <TrsSt>
· End-Effector: End-Effector specified (1 byte)
   · 'A': When End-Effector 1 specified
   · 'B': When End-Effector 2 specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · At normal reception: Responds "0000".
   · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MGT2 <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
   · At normal completion: Responds "0000".
   · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

6. MPT2

[Name]   Workpiese Put Motion

[Function]   This command is a combination command of "MTRSxxxxPx" and "MPUT".
This put motion trajectory is same as MPUT.
Performs the workpiece Put motion.　With YASKAWA standard interlock.

[Operation]   Performs the workpiece Put motion for the transfer station specified.
With YASKAWA standard interlock.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
$ <UNo> MPT2 <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo:　Unit number (1 byte)
   · '1':　Manipulator
· TrsSt:　Transfer station (2 bytes)
   <Manipulator>
   · "P1" to "P8"　　　　　:　When cassette stage specified
   · "UA" to "UL"　　　　　:　When transfer stage specified
· SlotNo: Slot number (2 bytes)
   <Manipulator>
   · "01" to "XX":　When cassette stage specified by <TrsST>
     (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
   · "00":　When transfer stage specified by <TrsSt>
· End-Effector:　End-Effector specified (1 byte)
   · 'A':　When End-Effector 1 specified
   · 'B':　When End-Effector 2 specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo:　　Unit number (1 byte)
   · '1':　Unit specified
· Sts:　　Status (2 byte)
· Ackcd:　Response code (4 bytes)
   · At normal reception:　Responds "0000".
   · Reception rejected:　Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MPT2 <Sum> <CR>

· UNo:　　Unit number (1 byte)
   · '1':　Unit specified
· Sts:　　Status (2 byte)
· Errcd:　Error code (4 bytes)
   · At normal completion:　　Responds "0000".
   · At erroneous completion:　Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

7. MSP2

[Name]   Workpiese Swap Motion

[Function]   Performs the workpiece Swap motion.   With YASKAWA standard interlock.
    The get motion trajectory is same as MGET, and the put motion trajectory is same as MPUT.

[Operation]   Performs the workpiece Swap motion (Get with empty end effector and Put with full end effector) for
    the transfer station specified.   With YASKAWA standard interlock.
    For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]   · The specified unit is stopped.
    · The specified unit is under servo ON state.
    · "Auto" mode
    · Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
    $ <u><UNo></u> <u>MSP2</u> <u><TrsSt></u> <u><SlotNo></u> <u><Sum></u> <u><CR></u>

    · UNo: Unit number (1 byte)
       · '1': Manipulator
    · TrsSt: Transfer station (2 bytes)
       <Manipulator>
       · "P1" to "P8"   :   When cassette stage specified
       · "UA" to "UL"   :   When transfer stage specified
    · SlotNo: Slot number (2 bytes)
       <Manipulator>
       · "01" to "XX": When cassette stage specified by <TrsST>
         (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
       · "00": When transfer stage specified by <TrsSt>

[Response Format]
    (Response Message)
    @ <u><UNo></u> <u><Sts></u> <u><Ackcd></u> <u><Subcd></u> <u><Sum></u> <u><CR></u>

    · UNo: Unit number (1 byte)
       · '1':   Unit specified
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
       · At normal reception: Responds "0000".
       · Reception rejected: Responds with the error code.
    · Subcd: Sub code (4 bytes)
       · Detailed alarm information is shown.
       · When either alarm does not occur, sub code is '0000'.

    (Message for the Completion of Execution)
    $ <u><UNo></u> <u><Sts></u> <u><Errcd></u> <u><Subcd></u> <u>MSP2</u> <u><Sum></u> <u><CR></u>

    · UNo: Unit number (1 byte)
       · '1' : Unit specified
    · Sts: Status (2 byte)
    · Errcd: Error code (4 bytes)
       · At normal completion: Responds "0000".
       · At erroneous completion: Responds with the error code.
    · Subcd: Sub code (4 bytes)
       · Detailed alarm information is shown.
       · When either alarm does not occur, sub code is '0000'.

8. MPNT

[Name]   Motion between Transfer Points

[Function]   Moves to the specified transfer point of the transfer station.
Note: This command is used by comibining with "MTRS."

[Operation]   Moves to the transfer point for the transfer station specified by "MTRS" or "MPNT" immediately before this command.   For the motion sequence, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· The "MTRS" command immediately before this command has been completed normally.

[Command Format]

$ <UNo> MPNT <TrsPnt> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Manipulator
· TrsPnt: Transfer point (2 bytes)
· G1:   Transfer point G1
· G2:   Transfer point G2
· Gb: Transfer point Gb
· G3:   Transfer point G3
· G4:   Transfer point G4
· G5:   Transfer point G5
· P1:   Transfer point P1
· P2:   Transfer point P2
· Pb: Transfer point Pb
· P3:   Transfer point P3
· P4:   Transfer point P4
Note: For details on the transfer points, refer to "6.5 Workpiece Transfer Motion Path."

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
· At normal reception: Responds "0000".
· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> MPNT <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
· At normal completion: Responds "0000".
· At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

9. MMAP

[Name]   Mapping

[Function]   Performs mapping for the specified transfer station.
For the mapping result, refer to the "33. RMAP" command that is described later.

[Operation]   Mapping for the specified transfer station is performed in the following order.
1. Moves the arm to the least rotating range.
2. Moves any axis other than the expansion axis to the mapping start position.
3. Moves the expansion axis to the mapping start position.
4. Moves the lifting axis to search for the workpiece.
5. Moves the expansion axis to the smallest sweep diameter.

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station.

[Command Format]
$ <UNo> MMAP <TrsSt> <Sum> <CR>

· UNo: Unit number (1 byte)
　　· '1':   Manipulator
· TrsSt: Transfer station (2 bytes)
　　· "P1" to "P8"   :   When cassette stage specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
　　· '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
　　· At normal reception: Responds "0000".
　　· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
　　· Detailed alarm information is shown.
　　· When either alarm does not occur, sub code is '0000'.
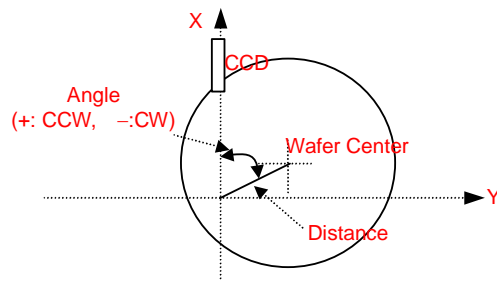
(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MMAP <Sum> <CR>

· UNo: Unit number (1 byte)
　　· '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
　　· At normal completion: Responds "0000".
　　· At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
　　· Detailed alarm information is shown.
　　· When either alarm does not occur, sub code is '0000'.

10. MALN

[Name]   Alignment

[Function]   Aligns the workpiece on the Pre-aligner.

[Operation]
1) The lifter will down when the down signal turn off
2) Wait to turn on the down signal and report the error to host when time-out
3) Chuck on
4) Wait to turn on the vacuum sensor signal, then chuck off and report the error to host when time-out
5) Move the rotary axis and samples the data for alignment
6) Detect the position data for center of the wafer and notch/orientation flat direction
7) Moves to the alignment position.

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode

[Command Format]

$ <UNo> MALN <TUNo> <Angle> <Sum> <CR>
- UNo: Unit number (1 byte)
  - '2':   Pre-aligner
- TUNo: Unit number to be compensated (1 bytes)
  - '1':   Manipulator (Constant Value)
- Angle: Positioning angle (6 bytes)
  - Relative angle with the position for workpiece positoning that is set by calibration as the reference
  - Specified in the range between "000000" and "035999" (resolution: 0.01 [deg])
  - If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>
- UNo: Unit number (1 byte)
  - '2':   Unit specified
- Sts: Status (2 byte)
- Ackcd: Response code (4 bytes)
  - At normal reception: Responds "0000".
  - Reception rejected: Responds with the error code.
- Subcd: Sub code (4 bytes)
  - Detailed alarm information is shown.
  - When either alarm does not occur, sub code is '0000'.

MALN (Cont'd)

(Opsitive Message for the Completion of Execution)
   $ <UNo> <Sts> <Errcd> <Subcd> MALN <Distance> <Angle> <Sum> <CR>
   · UNo: Unit number (1 byte)
      · '2':   Unit specified
   · Sts: Status (2 byte)
   · Errcd: Error code (4 bytes)
      · Responds "0000".
   · Subcd: Sub code (4 bytes)
      · Responds "0000".
   · Distance : The distance between rotation center of PA(Pre-aligner) and wefer center(4 byte)
      · Specified in the range between "0000" and "9999" (resolution: 0.01 [mm])
      · If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.
      · If a value is more than 10mm, Distance is "9999".
   · Angle: Direction of wafer center when putting the wafer on PA (6 byte)
      · Specified in the range between "-18000" and "018000" (resolution: 0.01 [deg])
      · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
      · The sign must be added to the highest digit.

(Negative Message for the Completion of Execution)
   $ <UNo> <Sts> <Ackcd> <Subcd> MALN <Sum> <CR>

   · UNo: Unit number (1 byte)
      · '2':   Unit specified by command
   · Sts: Status (2 byte)
   · Ackcd: Response code (4 bytes)
      · Responds with the error code.
   · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

11. MTCH

[Name]   Move to Registered Position

[Function]   Moves to the specified position (registered position / waiting positon of the registered position).

[Operation]   Moves to the registered position or ready position of the specified transfer station, considering the offset in the lifting direction or expanding direction specified by command parameter, in the following order.
1. Moves the arm to the least rotating range.
2. Moves any other axis other than the expansion axis to the ready position (the position considering the offset in the
   lifting direction for the lifting axis).

In addition, the following motions are performed when the registered position is specified by specifying the position mode.
3. Moves the expansion axis from the registered position to the position where the offset in the expanding direction is
   considered.
4. Moves the expansion axis and the lifting axis to the registered position at the same time.

[Conditions] · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station.

[Command Format]
   $ <UNo> MTCH <End-Effector> <TrsSt> <PMode> <ZOffset> <ROffset> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Manipulator
· End-Effector: End-Effector specified (1 byte)
   · 'A':   When End-Effector 1 specified
   · 'B':   When End-Effector 2 specified
· TrsSt: Transfer station (2 bytes)
   · "P1" to "P8"   :   When the lowest-layer position of the cassette stage specified
   · "UA" to "UL"   :   When transfer stage specified
   · "M1" to "M8"   :   When Mapping for Loadport 1 or 2 is specified
· PMode: Position mode (1 byte)
   · 'S':   Registered position
   · 'R: Ready position
· ZOffset: Offset in the lifting direction (6 bytes)
   · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm])
   · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
   · A sign is added to the highest digit.
   · Approach of the lifting axis differs depending on the offset sign.
      Offset value < 0:   Accesses from the lower side of the registered position.
      Offset value > 0:   Accesses from the higher side of the registered position.
      Offset value = 0:   Accesses at the height of the registered position.
· ROffset: Offset in the expanding direction (6 bytes)
   · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm])
   · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
   · A sign is added to the highest digit.
   · Approach of the expansion axis differs depending on the offset sign.
      Offset value < 0:   Accesses from the contracting direction of the registered position.
      Offset value > 0:   Accesses from the expanding direction of the registered position.
      Offset value = 0:   Accesses at the registered position.

MTCH (Cont'd)

[Response Format]
    (Response Message)
        <u>@</u> <u>\<UNo></u> <u>\<Sts></u> <u>\<Ackcd></u> <u>\<Subcd></u> <u>\<Sum></u> <u>\<CR></u>

        · UNo: Unit number (1 byte)
            · '1':   Unit specified
        · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
            · At normal reception: Responds "0000".
            · Reception rejected: Responds with the error code.
        · Subcd: Sub code (4 bytes)
            · Detailed alarm information is shown.
            · When either alarm does not occur, sub code is '0000'.

    (Message for the Completion of Execution)
        <u>$</u> <u>\<UNo></u> <u>\<Sts></u> <u>\<Errcd></u> <u>\<Subcd></u> <u>MTCH</u> <u>\<Sum></u> <u>\<CR></u>

        · UNo: Unit number (1 byte)
            · '1':   Unit specified
        · Sts: Status (2 byte)
        · Errcd: Error code (4 bytes)
            · At normal completion: Responds "0000".
            · At erroneous completion: Responds with the error code.
        · Subcd: Sub code (4 bytes)
            · Detailed alarm information is shown.
            · When either alarm does not occur, sub code is '0000'.

12. MABS

[Name]   Move to Specified Coordinate Position

[Function]   Moves the specified axis to a specified coordinate position.

[Operation]   Same as above.

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode

[Command Format]

$ <UNo> MABS <Axis> <Value> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' or '2':   Unit specified
· Axis: Axis (1 byte)
    <Manipulator>
    · 'S':   Rotary axis
    · 'E':   Expansion axis
    · 'A':   End-Effector 1
    · 'B':   End-Effector 2
    · 'Z':   Lifting axis
    · 'T':   Travel axis (with travel axis only)
    <Three-axis Pre-aligner>
    · 'X':   X-axis
    · 'Y':   Y-axis
· Value: Coordinate specified (6 bytes)
    · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm] or [deg])
    · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
    · A sign is added to the highest digit.
    · The moving directions are as follows.
    <Manipulator>

· Rotary axis:	+: Counterclockwise (CCW)	−:	Clockwise (CW)
· Expansion axis	+: End-Effector expanding direction	−:	End-Effector contracting direction
· Wrist axes 1 and 2:	+: Counterclockwise (CCW)	−:	Clockwise (CW)
· Lifting axis:	+: Upward	−:	Downward
· Travel axis:	+: Opposite direction to home position	−:	Home position direction

    <Three-axis Pre-aligner>

· X-axis:	+: Direction approaching the sensor viewed from the sensor		
	−: Direction aparting from the sensor viewed from the sensor		
· Y-axis:	+: Left viewed from the sensor	−:	Right viewed from the sensor

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' or '2':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

MABS (Cont'd)

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MABS <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
    - · At normal completion: Responds "0000".
    - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
    - · Detailed alarm information is shown.
    - · When either alarm does not occur, sub code is '0000'.

13. MRLK

[Name] Moves to Specified Relative Position

[Function] Moves the specified axis to the specified relative position. (Link (Radial) motion)

[Operation] Same as above.

**[Conditions]** · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode

[Command Format]

$ <UNo> MRLK <Axis> <Value> <Sum> <CR>

· UNo: Unit number (1 byte)
 · '1' or '2': Unit specified
· Axis: Axis (1 byte)
  <Manipulator>
  · 'S': Rotary axis, · 'E': Expansion axis
  · 'A': End-Effector 1, · 'B': End-Effector 2
  · 'Z': Lifting axis, · 'T': Travel axis (with travel axis only)
  <Three-axis Pre-aligner>
  · 'S': Rotary axis, · 'X': X-axis
  · 'Y': Y-axis

· Value: Coordinate specified (6 bytes)
 · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm] or [deg])
   However, the angle is to be specified in the range between "-35999" and "035999."
 · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
 · A sign is added to the highest digit.
 · The moving directions are as follows.
  <Manipulator>
  · Rotary axis: +: Counterclockwise (CCW) −: Clockwise (CW)
  · Expansion axis: +: End-Effector expanding direction −: End-Effector contracting direction
  · End-Effector 1 and 2: +: Counterclockwise (CCW) −: Clockwise (CW)
  · Lifting axis: +: Upward −: Downward
  · Travel axis: +: Opposite direction to home position −: Home position direction
  <Three-axis Pre-aligner>
  · Rotary axis: +: Counterclockwise (CCW) −: Clockwise (CW)
  · X-axis: +: Direction approaching the sensor viewed from the sensor
    −: Direction aparting from the sensor viewed from the sensor
  · Y-axis: +: Left viewed from the sensor −: Right viewed from the sensor

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
 · '1' or '2': Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
 · At normal reception: Responds "0000".
 · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
 · Detailed alarm information is shown.
 · When either alarm does not occur, sub code is '0000'.

MRLK (Cont'd)

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MREL <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
  - · At normal completion: Responds "0000".
  - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

14. MRLN

[Name]   Moves to Specified Relative Position

[Function]   Moves the specified axis to the specified relative position. (Linear (Off-center) motion)

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
              · The specified unit is under servo ON state.
              · "Auto" mode

[Command Format]

$ <UNo> MRLN <Axis> <Value> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Manipulator
· Axis: Axis (2 byte)
      <Manipulator>
    · "X1" : Horizontal linear moition (End-Effector 1) -LEFT/RIGHT Motion
    · "X2" : Horizontal linear moition (End-Effector 2) -LEFT/RIGHT Motion
    · "Y1" : Vertical linear motion (End-Effector 1) -FORE/AFT Motion
    · "Y2" : Vertical linear motion (End-Effector 2) -FORE/AFT Motion
· Value: Coordinate specified (6 bytes)
    · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm] or [deg])
      However, the angle is to be specified in the range between "-35999" and "035999."
    · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
    · A sign is added to the highest digit.
    · The moving directions are as follows.
    <Manipulator>
        · Horizontal motion: +:   Right to the direction that the End-Effector faces
         (Linear motion)   −:   Left to the direction that the End-Effector faces
        · Vertical motion: +:   Expanding direction to the direction that the End-Effector faces
         (Linear motion)   −:   Contracting direction to the direction that the End-Effector faces
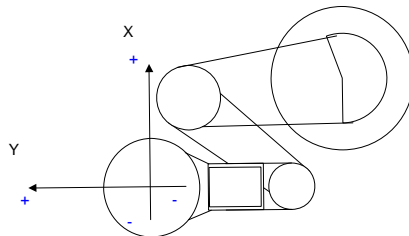
[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

MRLN (Cont'd)

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MREL <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1':   Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
  - · At normal completion: Responds "0000".
  - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

15. MACA

[Name]   Alignment Calibration

[Function]   Performs calibration for alignment.

[Operation]   1) The lifter will down when the down signal turn off
    2) Wait to turn on the down signal, then send the error to host when time-out
    3) Chuck on
    4) Wait to turn on the vacuum sensor signal, then chuck off and send the error to host when time-out
    5) Move the rotary axis and samples the data for alignment
    6) Detect the position data for center of the wafer and notch (orientation flat)
    7) Retains the reference angle for positioning of the workpiece in the Controller.

[Conditions]   · The specified unit is stopped.
    · The specified unit is under servo ON state.
    · "Auto" mode

[Command Format]
    $ <UNo> MACA <TUno> <Sum> <CR>

    · UNo: Unit number (1 byte)
        · '2':   Pre-aligner
    · TUNo: Unit number to be compensated (1 bytes)
        · '1':   Manipulator (Constant Value)

[Response Format]
(Response Message)
    @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

    · UNo: Unit number (1 byte)
        · '2':   Unit specified
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
        · At normal reception: Responds "0000".
        · Reception rejected: Responds with the error code.
    · Subcd: Sub code (4 bytes)
        · Detailed alarm information is shown.
        · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
    $ <UNo> <Sts> <Errcd> <Subcd> MACA <Sum> <CR>

    · UNo: Unit number (1 byte)
        · '2':   Unit specified
    · Sts: Status (2 byte)
    · Errcd: Error code (4 bytes)
        · At normal completion: Responds "0000".
        · At erroneous completion: Responds with the error code.
    · Subcd: Sub code (4 bytes)
        · Detailed alarm information is shown.
        · When either alarm does not occur, sub code is '0000'.

16. MMCA

[Name]   Mapping Calibration

[Function]   Performs calibration for mapping.

[Operation]   Performs calibration for the specified cassette in the following order.
1. Moves the expansion axis to the least rotating range.
2. Moves any axis other than the expansion axis to the mapping start position.
3. Moves the expansion axis to the mapping start position.
4. Moves the lifting axis to search for the workpiece.
5. Moves the expansion axis to the smallest sweep diameter.

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station.

[Command Format]

$ <UNo> MMCA <TrsSt> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Manipulator
· TrsSt: Transfer station (2 bytes)
    · "P1" to "P8":   When cassette stage specified

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> MMCA <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

17. MGTW

[Name]	Workpiese Get Motion

[Function]   Performs the workpiece Get motion.
With YASKAWA standard interlock.

[Operation]   Performs the workpiece Get motion for the transfer station specified.
With YASKAWA standard interlock.
Get and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]

$ <UNo> MGTW <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1': Manipulator
· TrsSt: Transfer station (2 bytes)
    <Manipulator>
    · "P1" to "P8"   : When cassette stage specified
    · "UA" to "UL"   : When transfer stage specified
· SlotNo: Slot number (2 bytes)
    <Manipulator>
    · "01" to "XX": When cassette stage specified by <TrsST>
        (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
    · "00": When transfer stage specified by <TrsSt>
· End-Effector: End-Effector specified (1 byte)
    · 'A': When End-Effector 1 specified
    · 'B': When End-Effector 2 specified

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

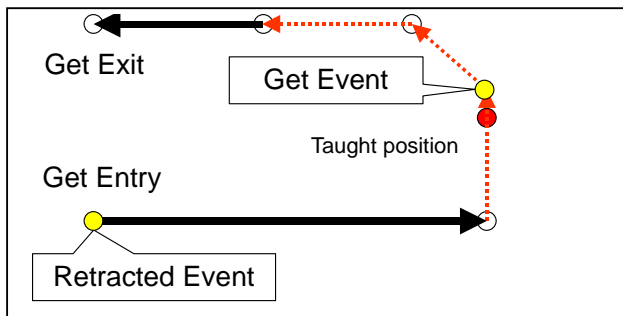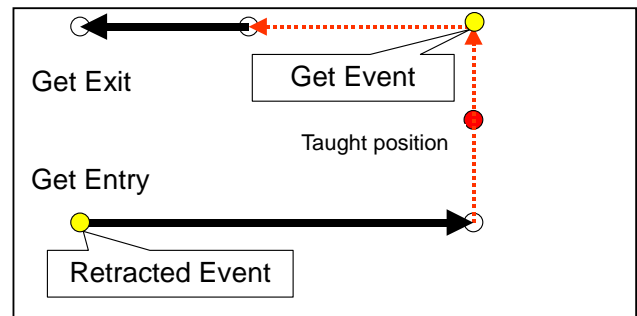(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> MGTW <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Retracted Event	"ARET"		UA to UL Station: "00"

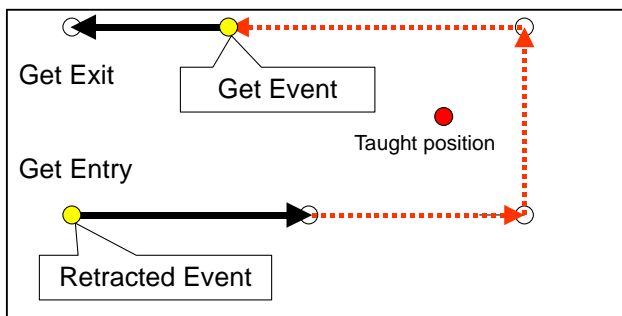Manipulator parameter I014=1 : Asynchronous information enable / =0 : desable
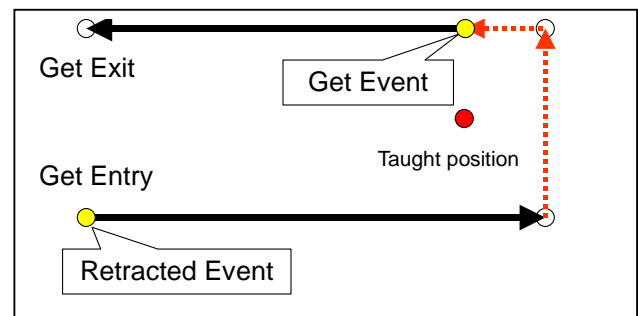
Vacuum Chuck

Trapezoid

Rectangle

Edge Grip

For cassette

For scan stage

18. MPTW

[Name]	Workpiese Put Motion

[Function]   Performs the workpiece Put motion.
With YASKAWA standard interlock.

[Operation]   Performs the workpiece Put motion for the transfer station specified.
With YASKAWA standard interlock.
Get and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]

$ <UNo> MPTW <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Manipulator
· TrsSt: Transfer station (2 bytes)
    <Manipulator>
    · "P1" to "P8"   :   When cassette stage specified
    · "UA" to "UL"   :   When transfer stage specified
· SlotNo: Slot number (2 bytes)
    <Manipulator>
    · "01" to "XX":   When cassette stage specified by <TrsST>
       (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
    · "00":   When transfer stage specified by <TrsSt>
· End-Effector: End-Effector specified (1 byte)
    · 'A':   When End-Effector 1 specified
    · 'B':   When End-Effector 2 specified

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
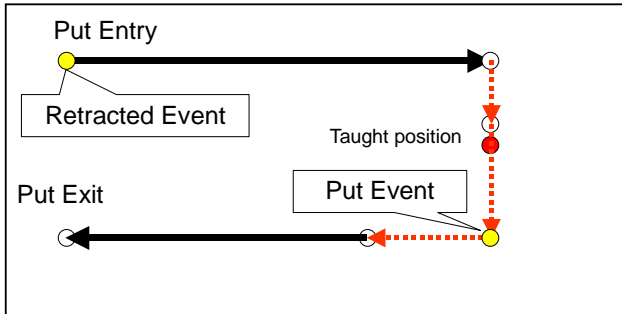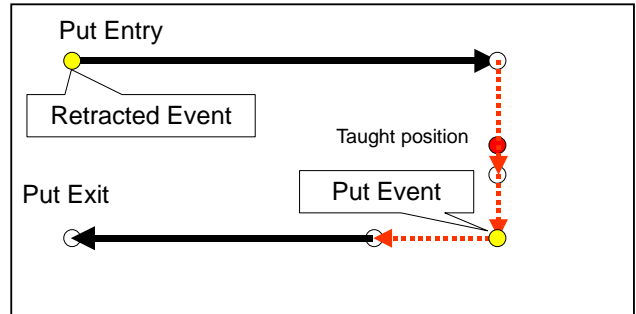
$ <UNo> <Sts> <Errcd> <Subcd> MPTW <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Put Event	"WPUT"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Retracted Event	"ARET"		UA to UL Station: "00"

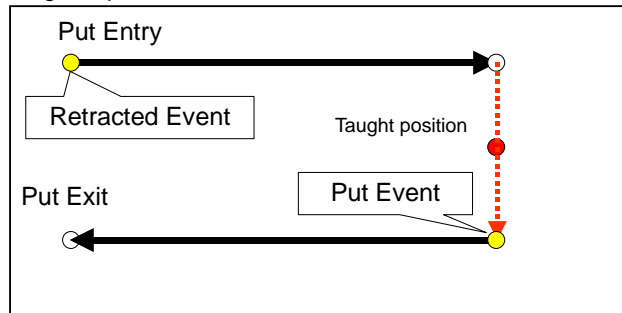Manipulator parameter I014=1 : Asynchronous information enable / =0 : desable
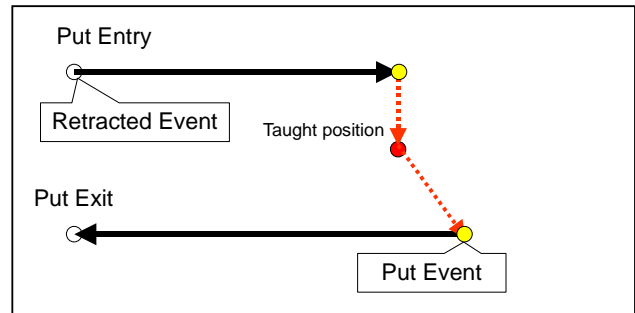
Vacuum Chuck

Soft put

Hard put

Edge Grip

For cassette

For scan stage

19. MGWI

[Name]   Workpiese Get Motion

[Function]   Performs the workpiece Get motion.
With Customized handshake.

[Operation]   Performs the workpiece Get motion for the transfer station specified.
With customized handshake.
Get and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
$ <UNo> MGWI <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1': Manipulator
· TrsSt: Transfer station (2 bytes)
   <Manipulator>
   · "P1" to "P8"   : When cassette stage specified
   · "UA" to "UL"   : When transfer stage specified
· SlotNo: Slot number (2 bytes)
   <Manipulator>
   · "01" to "XX": When cassette stage specified by <TrsST>
     (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
   · "00": When transfer stage specified by <TrsSt>
· End-Effector: End-Effector specified (1 byte)
   · 'A': When End-Effector 1 specified
   · 'B': When End-Effector 2 specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · At normal reception: Responds "0000".
   · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

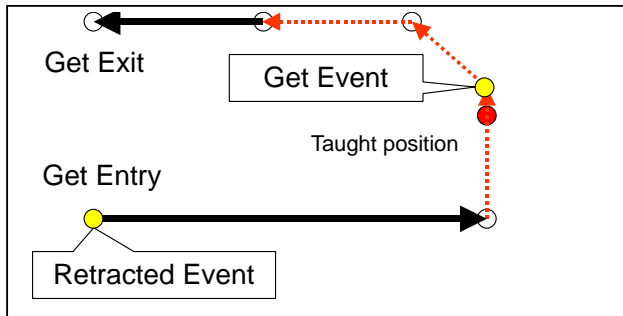(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> MGWI <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
   · At normal completion: Responds "0000".
   · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
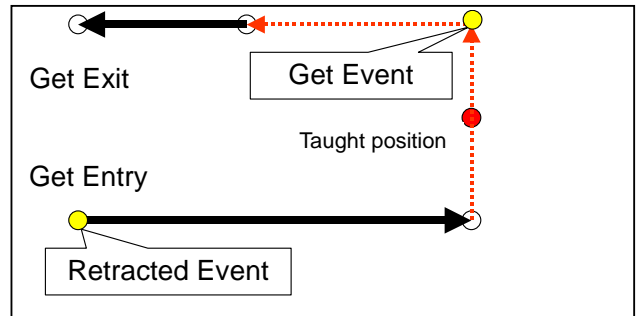   · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Retracted Event	"ARET"		UA to UL Station: "00"

Manipulator parameter I014=1 : Asynchronous information enable /   =0 : desable

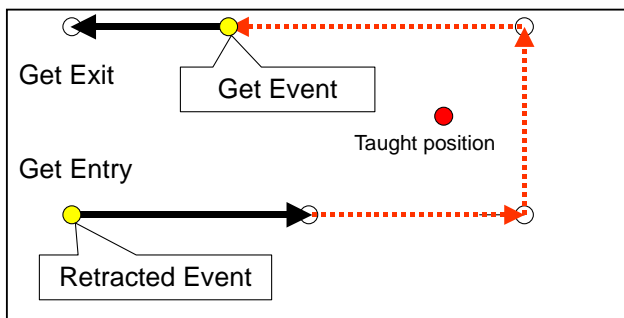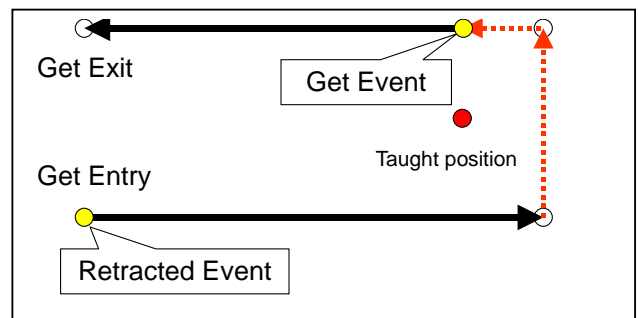Vacuum Chuck

Trapezoid

Rectangle

Edge Grip

For cassette

For scan stage

20. MPWI

[Name]   Workpiese Put Motion

[Function]   Performs the workpiece Put motion.
With Customized handshake.

[Operation]   Performs the workpiece Put motion for the transfer station specified.
With customized handshake.
Get and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]

$ <UNo> MPWI <TrsSt> <SlotNo> <Frok> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1': Manipulator
· TrsSt: Transfer station (2 bytes)
    <Manipulator>
    · "P1" to "P8"   : When cassette stage specified
    · "UA" to "UL"   : When transfer stage specified
· SlotNo: Slot number (2 bytes)
    <Manipulator>
    · "01" to "XX": When cassette stage specified by <TrsST>
      (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
    · "00": When transfer stage specified by <TrsSt>
· End-Effector: End-Effector specified (1 byte)
    · 'A': When End-Effector 1 specified
    · 'B': When End-Effector 2 specified

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

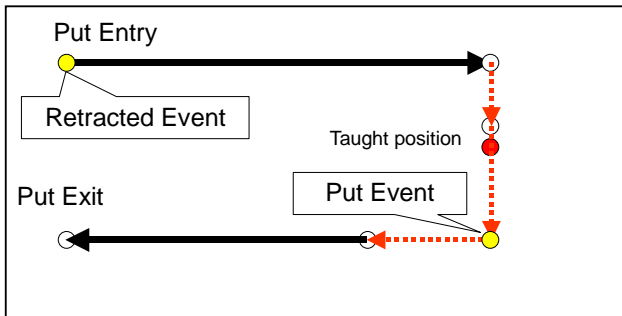$ <UNo> <Sts> <Errcd> <Subcd> MPWI <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Put Event	"WPUT"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Retracted Event	"ARET"		UA to UL Station: "00"

Manipulator parameter I014=1 : Asynchronous information enable /   =0 : disable

Vacuum Chuck

Soft put

Hard put

Edge Grip

For cassette

For scan stage

21. MSWP

[Name]   Workpiese Swap Motion

[Function]   Performs the workpiece Swap motion.
With YASKAWA interlock.

[Operation]   Performs the workpiece Swap motion (Get with empty End-Effector and Put with full End-Effector/ at Pre-aligner stage, Manipulator get with empty End-Effector -> Pre^aligner move to the alignment start position -> Manipulator put with full End-Effector) for the transfer station specified.
With YASKAWA standard interlock.
Get and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]

$ <UNo> MSWP <TrsSt> <SlotNo> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1': Manipulator
· TrsSt: Transfer station (2 bytes)
   <Manipulator>
   · "P1" to "P8"   :   When cassette stage specified
   · "UA" to "UL"   :   When transfer stage specified
· SlotNo: Slot number (2 bytes)
   <Manipulator>
   · "01" to "XX":   When cassette stage specified by <TrsST>
      (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
   · "00":   When transfer stage specified by <TrsSt>

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · At normal reception: Responds "0000".
   · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
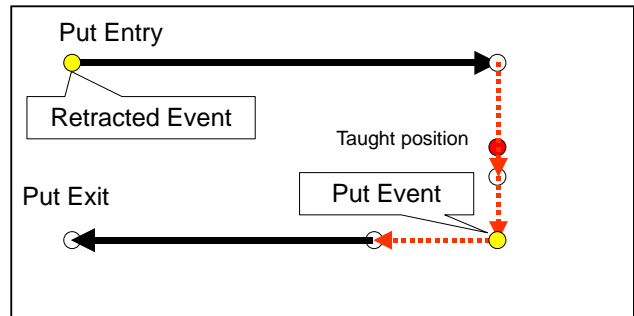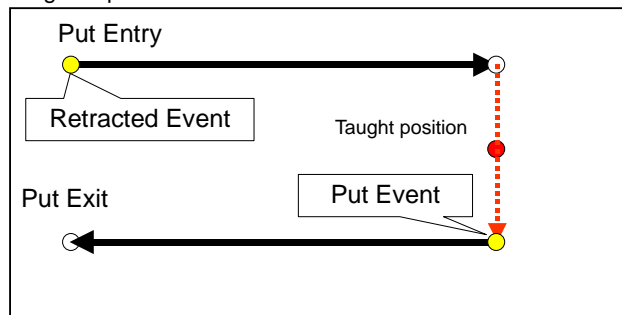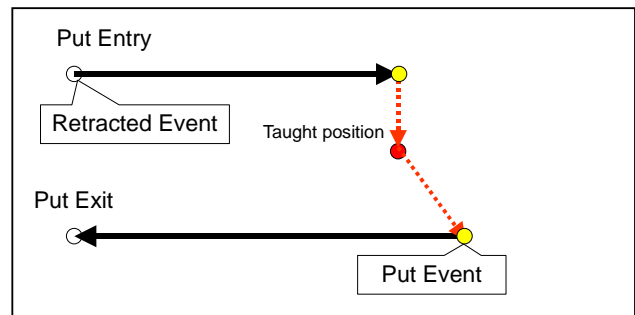   · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> MSWP <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' :   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
   · At normal completion: Responds "0000".
   · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Put Event	"WPUT"		UA to UL Station: "00"
Retracted Event	"ARET"		

Manipulator parameter I014=1 : Asynchronous information enable /   =0 : disable

Vacuum Chuck

Trapezoid

Rectangle

Soft put

Hard put

Edge Grip

For cassette

For scan stage

For cassette

For scan stage

22.. MSWI

[Name]   Workpiese Swap Motion

[Function]   Performs the workpiece Swap motion.
            With Customized handshake.

[Operation] Performs the workpiece Swap motion (Get with empty end effector and Put with full end effector/ at Pre-aligner stage, Manipulator get with empty End-Effector -> Pre^aligner move to the alignment start position -> Manipulator put with full End-Effector) for the transfer station specified.
            With customized handshake.
            Get and Retracted event information is sent according to the timing shown in the next page.
            For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions] · The specified unit is stopped.
            · The specified unit is under servo ON state.
            · "Auto" mode
            · Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
            $ <UNo> MSWI <TrsSt> <SlotNo> <Sum> <CR>

            · UNo: Unit number (1 byte)
                · '1':   Manipulator
            · TrsSt: Transfer station (2 bytes)
                <Manipulator>
                · "P1" to "P8"   :   When cassette stage specified
                · "UA" to "UL"   :   When transfer stage specified
            · SlotNo: Slot number (2 bytes)
                <Manipulator>
                · "01" to "XX":   When cassette stage specified by <TrsST>
                    (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
                · "00":   When transfer stage specified by <TrsSt>

[Response Format]
            (Response Message)
            @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

            · UNo: Unit number (1 byte)
                · '1':   Unit specified
            · Sts: Status (2 byte)
            · Ackcd: Response code (4 bytes)
                · At normal reception: Responds "0000".
                · Reception rejected: Responds with the error code.
            · Subcd: Sub code (4 bytes)
                · Detailed alarm information is shown.
                · When either alarm does not occur, sub code is '0000'.

            (Message for the Completion of Execution)
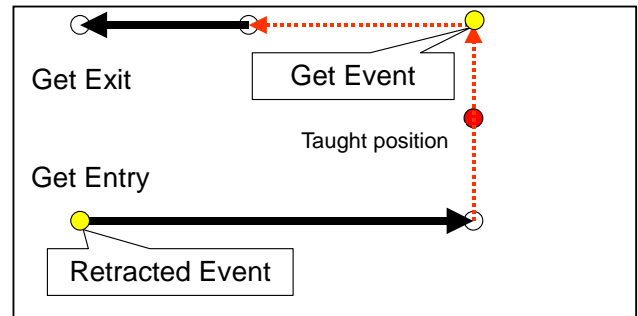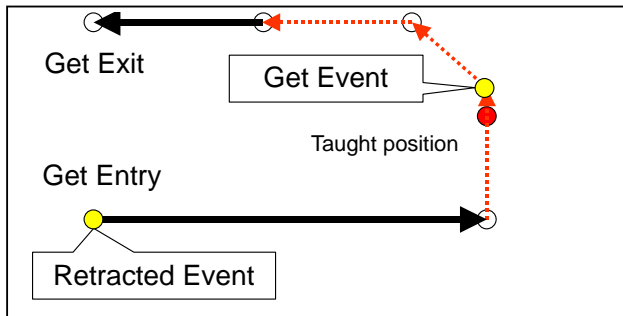 $ <UNo> <Sts> <Errcd> <Subcd> MSWI <Sum> <CR>

            · UNo: Unit number (1 byte)
                · '1':   Unit specified
            · Sts: Status (2 byte)
            · Errcd: Error code (4 bytes)
                · At normal completion: Responds "0000".
                · At erroneous completion: Responds with the error code.
            · Subcd: Sub code (4 bytes)
                · Detailed alarm information is shown.
                · When either alarm does not occur, sub code is '0000'

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Put Event	"WPUT"		UA to UL Station: "00"
Retracted Event	"ARET"		

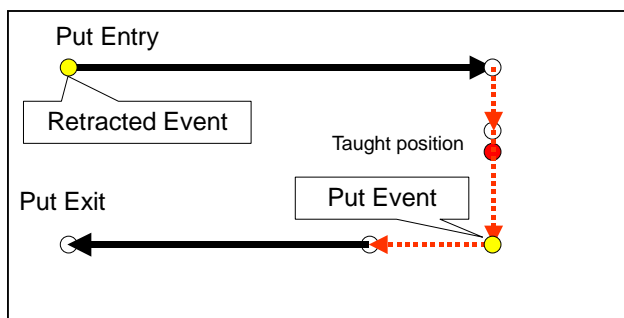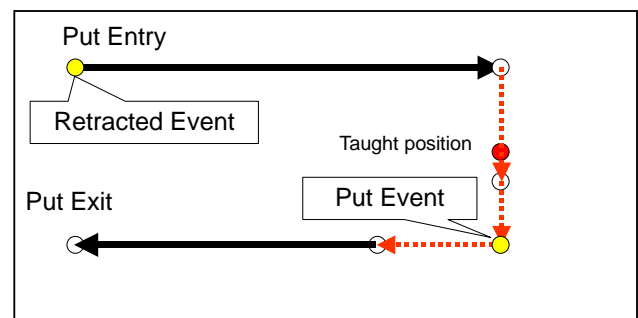Manipulator parameter I014=1 : Asynchronous information enable /   =0 : disable
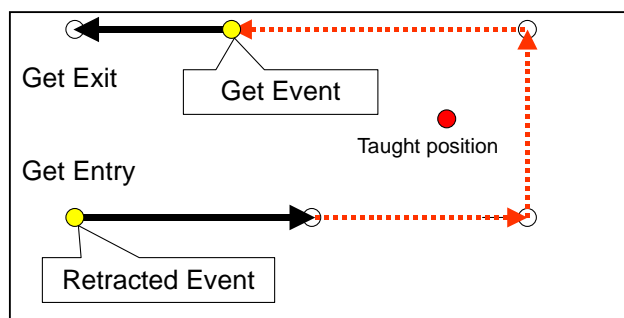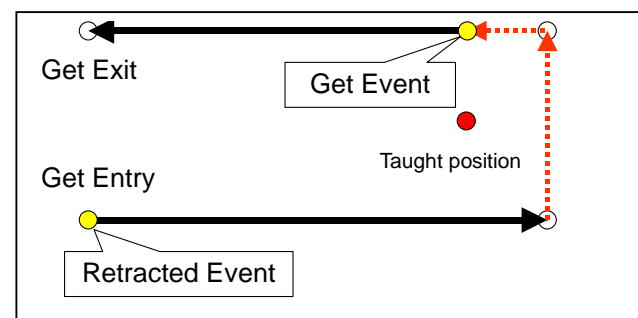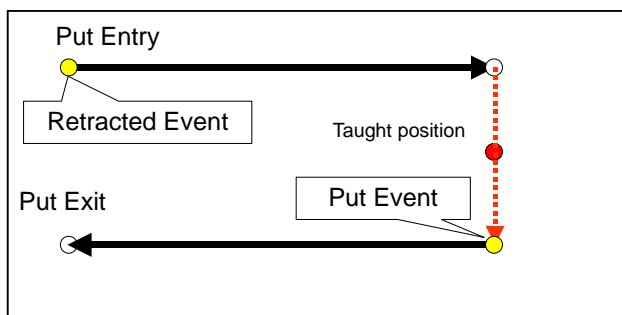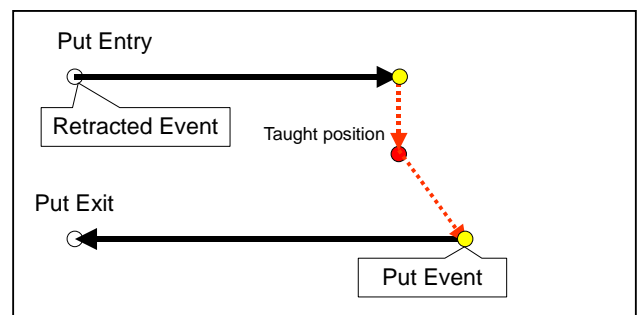
Vacuum Chuck

Trapezoid

Rectangle

Soft put

Hard put

Edge Grip

For cassette

For scan stage

For cassette

For scan stage

22.MSWI

23. MXTW

[Name]   Wafer Exchange Motion (w/ YASKAWA standard interlock)

[Function]   This command is a combination command of "MGTW" and "MPTW".
This get/put motion trajectory is same as MGTW / MPTW.
Wafer exchange between specified stations / slots is performed.
With YASKAWA standard interlock.

[Operation]   Wafer exchange between specified stations / slots is performed.
With YASKAWA standard interlock.
Get, Put and Retracted event information is sent according to the timing shown in the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]
$ <UNo> MXTW <1stMtn> <TrsSt1> <SlotNo1> < End-Effector 1> <TrsSt2> <SlotNo2>
< End-Effector 2> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Manipulator
· 1stMtn: First motion type (1 bytes)
· "G":   The first access is Get operation. (The second access Put-operates.)
· "P":   The first access is Put operation. (The second access Get-operates.)
· TrsSt1/2:   Transfer station (2 bytes)
· "P1" to "P8"   :   When cassette stage specified
· "UA" to "UL"   :   When transfer stage specified
· SlotNo1/2: Slot number (2 bytes)
· "01" to "XX":   When cassette stage specified by <TrsSt1/2>
(The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
· "00":   When transfer stage specified by <TrsSt1/2>
· End-Effector1/2:   End-Effector specified (1 byte)
· 'A':   When End-Effector 1 specified
· 'B':   When End-Effector 2 specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
· At normal reception: Responds "0000".
· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

MXTW (Cont'd)

(Message for the Completion of Execution)
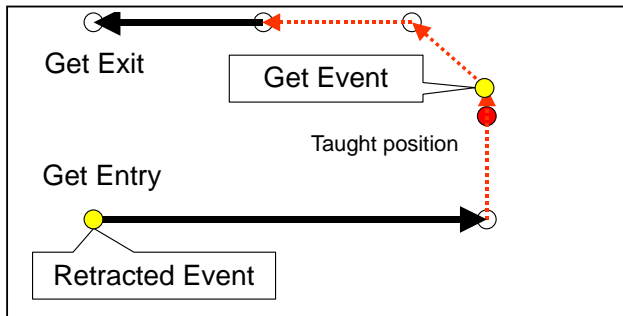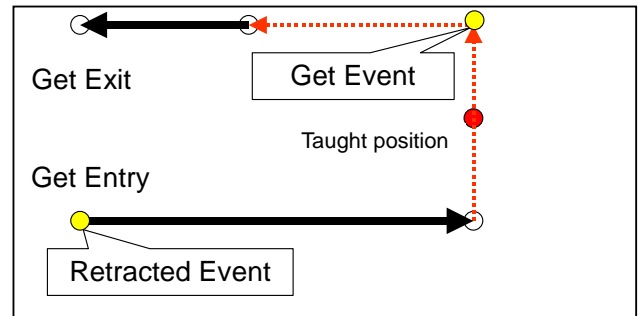$ <UNo> <Sts> <Errcd> <Subcd> MXTW <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1': Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
  - · At normal completion: Responds "0000".
  - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25" UA to UL Station: "00"
Put Event	"WPUT"		
Retracted Event	"ARET"		

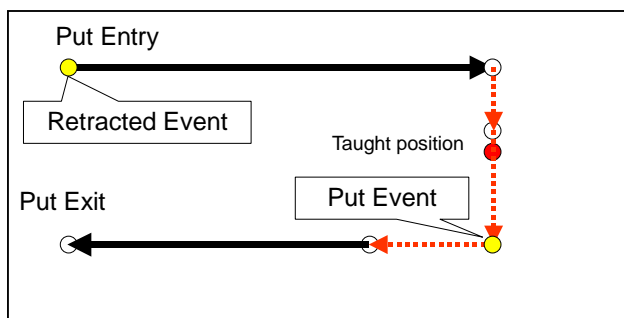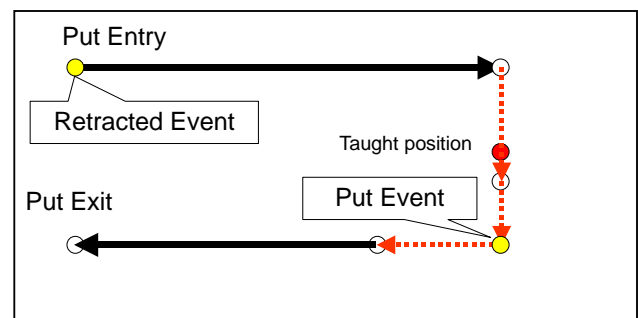Manipulator parameter I014=1 : Asynchronous information enable / =0 : disable

Vacuum Chuck
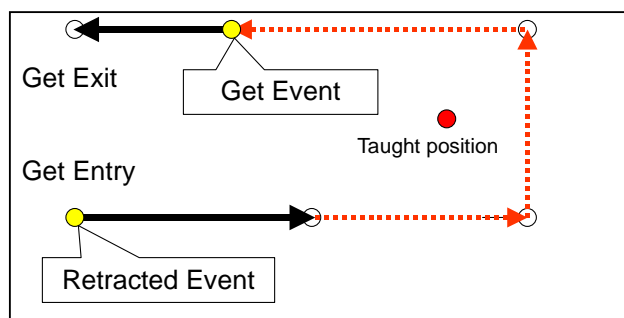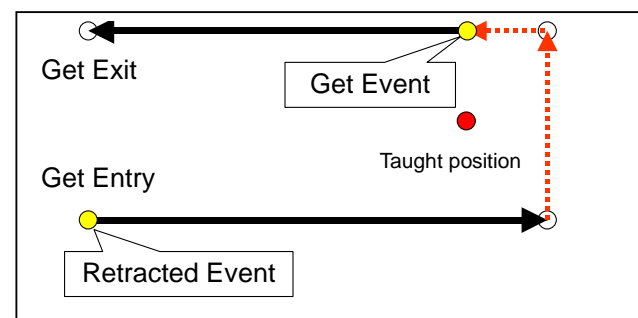

Trapezoid
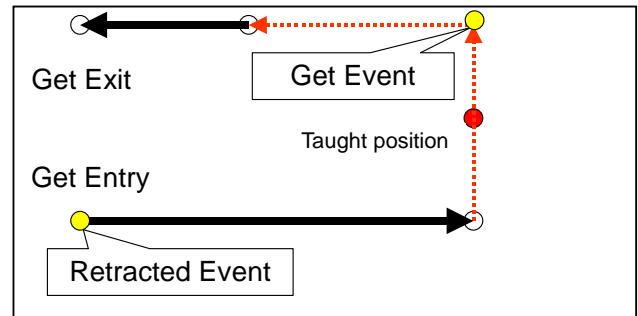
Rectangle
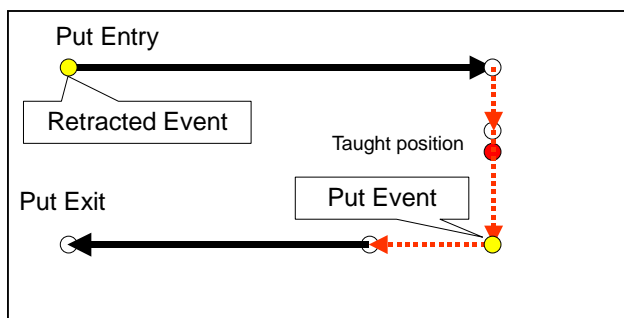
Soft put
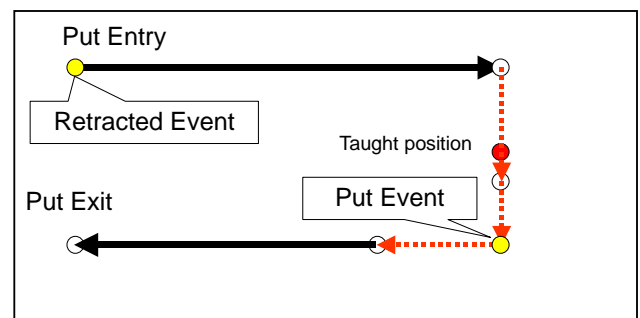
Hard put

Edge Grip
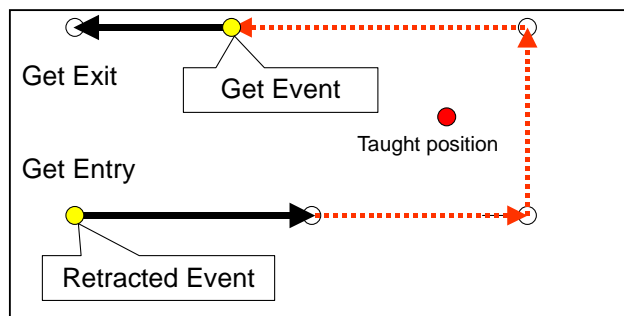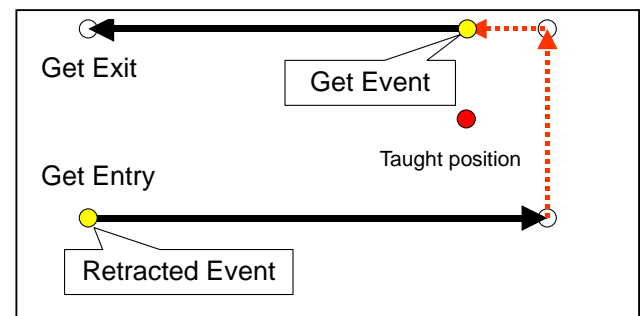
For cassette
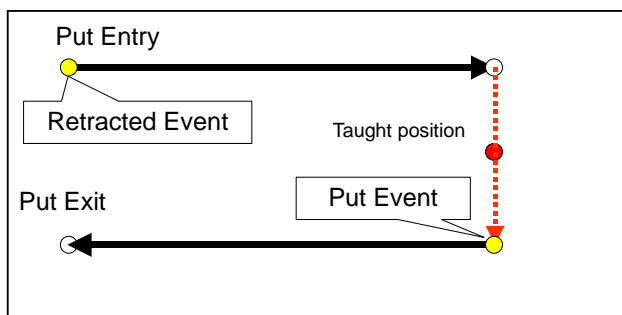
For scan stage

For cassette
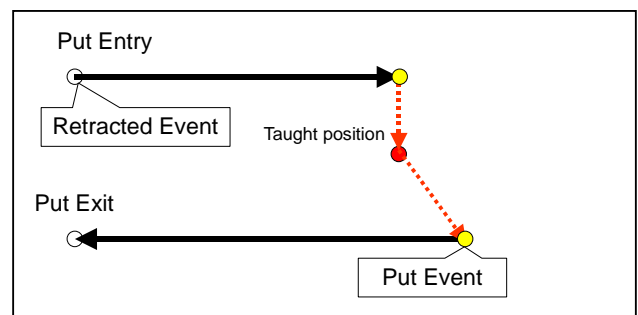
For scan stage

23.MXTW

24. MXWI

[Name]   Wafer Exchange Motion (w/ Customized handshake)

[Function]   This command is a combination command of "MGWI" and "MPWI".
This get/put motion trajectory is same as MGWI / MPWI.
Wafer exchange between specified stations / slots is performed.
With Customized handshake.

[Operation]   Wafer exchange between specified stations / slots is performed.
With Customized handshake.
Get, Put and Retracted event informations are sent according to the timing shown on the next page.
For the motion pattern, refer to "6.5 Workpiece Transfer Motion Path."

[Conditions]
· The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station (for the manipulator only).

[Command Format]

$ <UNo> MXWI <1stMtn> <TrsSt1> <SlotNo1> <Frok1> <TrsSt2> <SlotNo2> <Frok2> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1': Manipulator
· 1stMtn: First motion type (1 bytes)
    · "G": The first access is Get operation. (The second access Put-operates.)
    · "P": The first access is Put operation. (The second access Get-operates.)
· TrsSt1/2: Transfer station (2 bytes)
    · "P1" to "P8"   : When cassette stage specified
    · "UA" to "UL"   : When transfer stage specified
· SlotNo1/2: Slot number (2 bytes)
    · "01" to "XX": When cassette stage specified by <TrsSt1/2>
       (The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)
    · "00": When transfer stage specified by <TrsSt1/2>
· End-Effector1/2: End-Effector specified (1 byte)
    · 'A': When End-Effector 1 specified
    · 'B': When End-Effector 2 specified

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

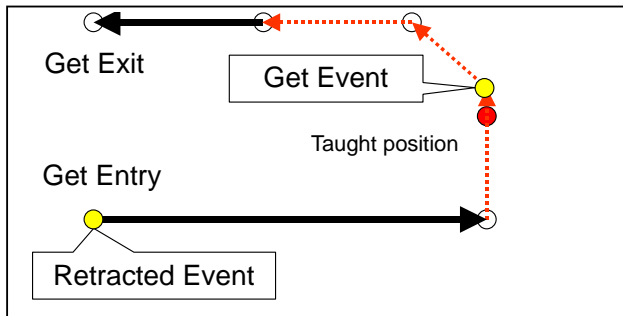$ <UNo> <Sts> <Errcd> <Subcd> MXWI <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1': Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
    - · At normal completion: Responds "0000".
    - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
    - · Detailed alarm information is shown.
   - · When either alarm does not occur, sub code is '0000'.

Event Information	Information Massage		
	Event Code (4 Bytes)	Station Name (2 Bytes)	Slot Number (2 Bytes)
Get Event	"WGET"	"P1" to "P8", "UA" to "UL"	P1 to P2 Station: "01" to "25"
Put Event	"WPUT"		UA to UL Station: "00"
Retracted Event	"ARET"		

Manipulator parameter I014=1 : Asynchronous information enable /　=0 : disable

Vacuum Chuck

Trapezoid

Rectangle


Soft put

Hard put

Edge Grip

For cassette

For scan stage

For cassette

For scan stage

24.MXWI

<7>

24a MCDT

[Name] Manipulator collision detection threshold setting command

[Function] Measures
Performs wafer-get and wafer–put motion.
The trajectories are the same as "MGTW" and "MPTW" command.
No YASKAWA standard or SFS interlock available

[Operation] 1) Temporary threshold value (6000) set for End-Effector1 and End-Effector2
2) Maximum disturbance value collection

<Exercise motion>
-MGTW and MPTW motion with End-Effector1 to <TrsSt1> at <SlotNo1> 5 times each.
<Measurement>
1. Wafer-get and Wafer-put motion with End-Effector1 to <TrsSt1> at <SlotNo1> for 5 times each.
2. Wafer-get and Wafer-put motion with End-Effector2 to <TrsSt1> at <SlotNo1> for 5 times each.
3. Wafer-get and Wafer-put motion with End-Effector1 to <TrsSt2> at <SlotNo2> for 5 times each.
4. Wafer-get and Wafer-put motion with End-Effector2 to <TrsSt2> at <SlotNo2> for 5 times each.

Maximum disturbance value is cleared at Entry RDY position and is obtained at Exit RDY position for each wafer-get/wafer-put motion.
Maximum disturbance value is averaged for each 1-4.

<Collision detection threshold value calculation>
Collision detection threshold value is calculated by multiplying by a safety coefficient with maximum number from four averaged numbers.   This calculation is done for End-Effector1 and End-Effector2 separately.

<Collision detection threshold value update>
If the calculated threshold value is greater than 5% and less than 30 % of current threshold value, it is applied as the threshold value.

[Conditions] · The specified unit is stopped.
· The specified unit is under servo ON state
· "Auto" mode
· Registration of positions has already been completed in the specified transfer station(for the manipulator only).
· Specified load ports are open.

[Command Format]
$ <UNo> MCDT <TrsSt1> <SlotNo1> <TrsSt2> <SlotNo2> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' :   Unit specified
· TrsSt1:   Transfer station (2 bytes)
· "P1" to "P8" :   Station assignment (P1~P8 parameter i272~279) must be "Left".
· TrsSt2:   Transfer station (2 bytes)
· "P1" to "P8" :   Station assignment (P1~P8 parameter i272~279) must be "Right".
· SlotNo1/2: Slot number (2 bytes)
· "01" to "XX" :   When cassette stage specified by <TrsSt1/2>
(The maximum value of "XX" is in the ASCII code of the number of slots specified by parameter.)

<7>

MCDT (Cont'd)

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' :   Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · At normal reception: Responds "0000".
  - · Reception rejected: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> SCDT <Value1> <Value2> <Sum> <CR>
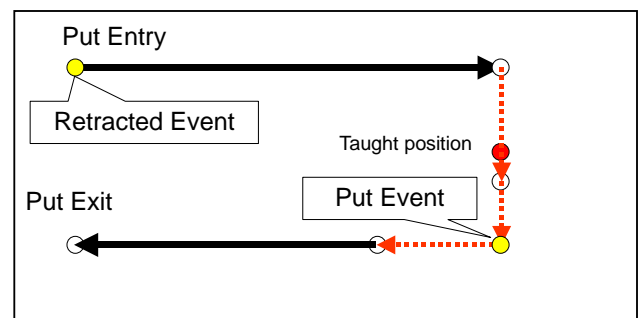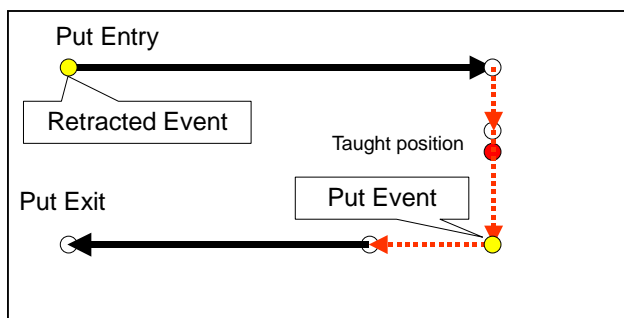
- · UNo: Unit number (1 byte)
  - · '1':   Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
  - · At normal completion: Responds "0000".
  - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.
- · Value: Coordinate (6 bytes each)
  - · Specified in the range between "-99999" and "999999" (resolution: 0.001 [mm], [deg])
  - · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
  - · The sign is added to the highest digit.
    - Value 1: Threshold of collision detection (End-Effector1).
    - Value 2: Threshold of collision detection (End-Effector2).

[Note]   As this command is to set optimized unique collision detection threshold for each manipulator,
execute this command in case of followings

- ·   Robot replacement
- ·   Controller replacement
- ·   Load port station location change
- ·   Manipulator belt tension adjustment

25. ISYS

[Name] System Inisialization

[Function] The system is initialized in the following order.
<Manipulator>
    1) The error is cleared.
    2) Servo ON is done.
    3) Chuck ON. (In vaccum type, if there is a no wafer, chuck OFF.)
    4) The robot moves to the minimum sweep position.
<Pre-aligner>
    1) The error is cleared.
    2) Servo ON is done.
    3) Chuck ON. (In vaccum type, if there is a no wafer, chuck OFF.)
    4) The Pre-aligner moves to the alignment start position.
    5) Alignes the workpiece, if there is a wafer on the Pre-aligner. (alignment angle is 0.0 degree.)

[Operation] Same as above.

[Conditions]
· The specified unit is stopped.
· "Auto" mode
· In case of Pre-aligner, The "ISYS" command of the robot had to execute beforehand.

[Command Format]
$ <UNo> ISYS <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' or '2': Unit specified

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' or '2': Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

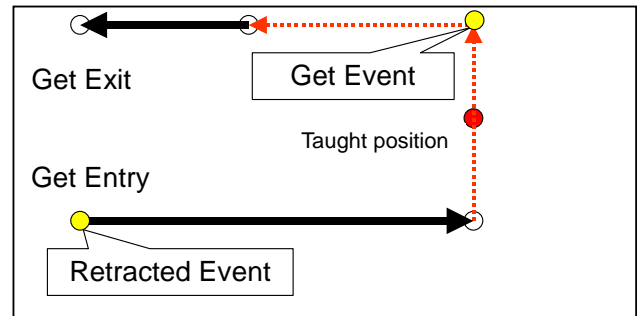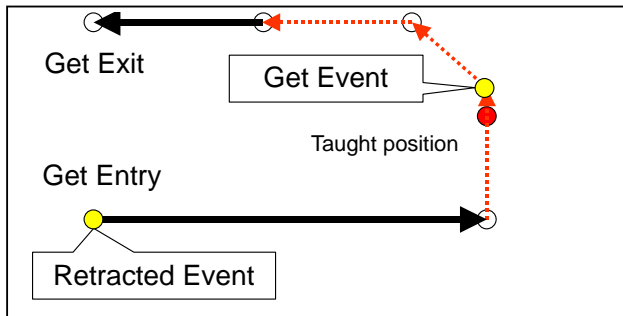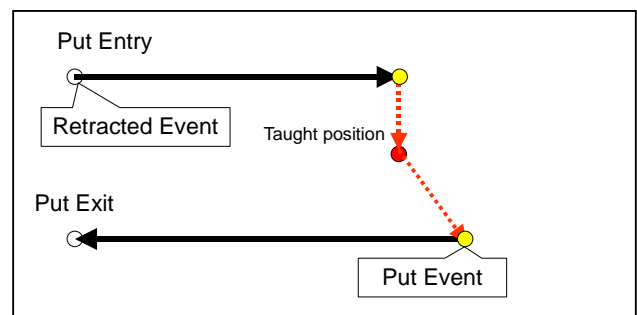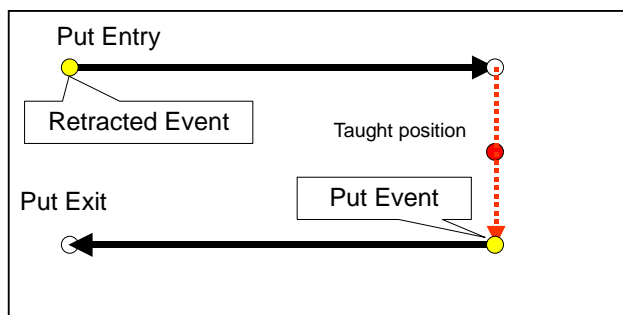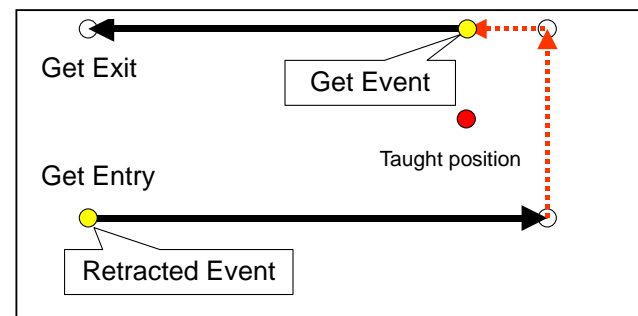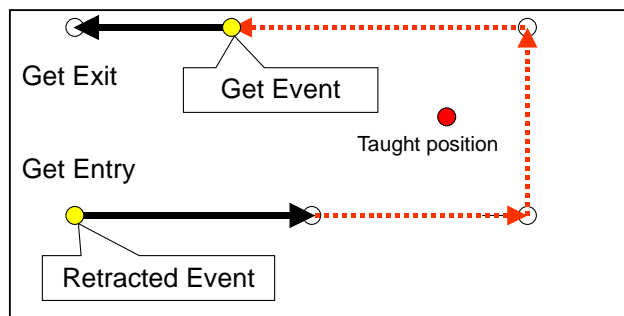(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> ISYS <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' or '2': Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
    · At normal completion: Responds "0000".
    · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

[Note] IF Reference command is performed while executing ISYS command, the response may get delayed.

26. MWRM

[Name]   Warm Up operation

[Function]   After moving to the minimum revolution posture, only the appointed time repeats Warm Up   operation and moves to the minimum revolution posture again.

[Operation]   Same as above

[Conditions] · The specified unit is stopped.
          · The specified unit is under servo ON state.
          · "Auto" mode

[Command Format]
        $ <UNo MWRM <Time> <Sum> <CR>

       · UNo: Unit number (1 byte)
         · '1':   Manipulator
       · Time: Warm Up time (4 byte)
         · '0001' to '9999':   Warm Up time。 Unit is sec.

[Response Format]
   (Response Message)
       @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

       · UNo: Unit number (1 byte)
         · '1':   Manipulator
       · Sts: Status (2 byte)
       · Ackcd: Response code (4 bytes)
         · At normal reception:Responds "0000".
         · Reception rejected: Responds with the error code.
       · Subcd: Sub code (4 bytes)
         · Detailed alarm information is shown.
         · When either alarm does not occur, sub code is '0000'.

   (Message for the Completion of Execution)
       $ <UNo> <Sts> <Errcd> <Subcd> MWRM <Sum> <CR>

       · UNo: Unit number (1 byte)
         · '1':   Manipulator
       · Sts: Status (2 byte)
       · Errcd: Error code (4 bytes)
         · At normal completion: Responds "0000".
         · At erroneous completion: Responds with the error code.
       · Subcd: Sub code (4 bytes)
         · Detailed alarm information is shown.
         · When either alarm does not occur, sub code is '0000'.

MWRM (Cont'd)

A sequence of operation
- ① MHOM of only an arm
- ② Move to (a) position.
- ③ Move to (b) position.
- ④ Move to (c) position.
- ⑤ Only the number of times of specification repeats   Move to (b)(c) position.
  The number of times of operation = The appointed time   * The number of times of operation per minute (it sets up by i variable) / 60
- ⑥ Move to (a) position


```
(a)  position default data
     and setting
     Manipulator parameter
TH: 90 [deg]          :R450
EX: -250 [mm]         :R451
B1: 0 [deg]
B2: 0 [deg]
Z: 50 [mm]            :R454
```



```
(b)  position default data
     and setting
     Manipulator parameter
TH: 90 [deg]          :R450
EX: -250 [mm]         :R451
B1: 40 [deg]          :R452
B2: -60 [deg]         :R453
Z: 50 [mm]            :R454
```



```
(c)  position default data
     and setting
     Manipulator parameter
TH: 115 [deg]         :R455
EX: -270 [mm]         :R456
B1: -60 [deg]         :R457
B2: 40 [deg]          :R458
Z: 100 [mm]           :R459
```

26.MWRM

27. CHLT

[Name]	Motion Interruption (Deceleration to a Stop)
[Function]	Decelerates the specified unit to a stop during moving by using a drive command. Interruption of Customized handshake release waiting.
[Operation]	Same as above.
[Conditions]	· "Auto" mode

[Command Format]

$ <UNo> CHLT <Sum> <CR>

· UNo:　Unit number (1 byte)
　　· '1' :　Manipulator

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo:　　Unit number (1 byte)
　　· '1':　　Unit specified
· Sts:　　Status (2 byte)
· Ackcd:　Response code (4 bytes)
　　· At normal reception: Responds "0000".
　　· Reception rejected: Responds with the error code.
· Subcd:　Sub code (4 bytes)
　　· Detailed alarm information is shown.
　　· When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> CHLT <Sum> <CR>

· UNo:　　Unit number (1 byte)
　　· '1' :　　Unit specified
· Sts:　　Status (2 byte)
· Errcd:　Error code (4 bytes)
　　· At normal completion:　　Responds "0000".
　　· At erroneous completion:　Responds with the error code.
· Subcd:　Sub code (4 bytes)
　　· Detailed alarm information is shown.
　　· When either alarm does not occur, sub code is '0000'.

28. CRSM

[Name]   Restart from Interruption

[Function]   Restarts the motion interrupted by deceleration stop ("CHLT") command.

[Operation]   Same as above.
Restarts the motion. After completion of the motion, sends the response message for the completion of execution using the drive command immediately before this command.

[Conditions]   · The specified unit is stopped.
· The specified unit is under servo ON state.
· "Auto" mode

[Command Format]

$ <UNo> CRSM <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1' :   Manipulator

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
    · '1':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
    · At normal reception: Responds "0000".
    · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
Follows to the message for the completion of execution of each drive command.

29. CEMG

[Name]   Deceleration to a stop and servo OFF

[Function]   Decelerates the specified unit to a stop and turns OFF the servo power supply.

[Operation]   Same as above

[Conditions]   None

[Command Format]

    $ <UNo> CEMG <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1' or '2':   Unit specified

[Response Format]

    @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1' or '2':   Unit specified
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
      · At normal reception: Responds "0000".
      · Reception rejected: Responds with the error code.
    · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
      · When either alarm does not occur, sub code is '0000'.

[Note]   This command cannot be used as Safty Interlock.
This command has no message sent from the Controller for the completion of execution. Therefore, the acknowledgement message for the completion of execution for this command from the Host is not needed. However, when this command is issued during motion executed by a drive command, the message for the completion of execution (termination with error) of the drive command is informed to the Host from the Controller. Therefore, the acknowledgement message for the completion of execution is needed.

30. CSRV

[Name]   Servo Command

[Function]   Turns ON/OFF the servo of the specified unit.

[Operation]   Same as above.

[Conditions]   <At Servo ON>
              · The specified unit is stopped.
              · "Auto" mode
              · No alarm is occurring.
             <At Servo OFF>
              · The specified unit is stopped.
              · "Auto" mode

[Command Format]
              $ <UNo> CSRV <Sw> <Sum> <CR>

              · UNo: Unit number (1 byte)
                 · '1' or '2':   Unit specified
              · Sw: Servo command (1 byte)
                 · '0':   Servo OFF
                 · '1':   Servo ON

[Response Format]
        (Response Message)
              @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

              · UNo: Unit number (1 byte)
                 · '1' or '2':   Unit specified
              · Sts: Status (2 byte)
              · Ackcd: Response code (4 bytes)
                 · At normal reception: Responds "0000".
                 · Reception rejected: Responds with the error code.
              · Subcd: Sub code (4 bytes)
                 · Detailed alarm information is shown.
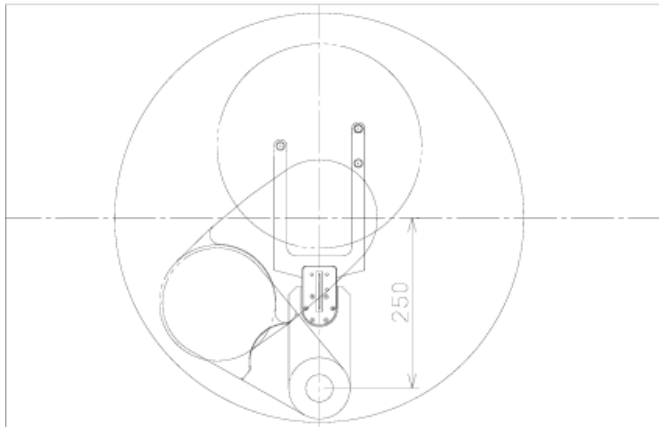   · When either alarm does not occur, sub code is '0000'.
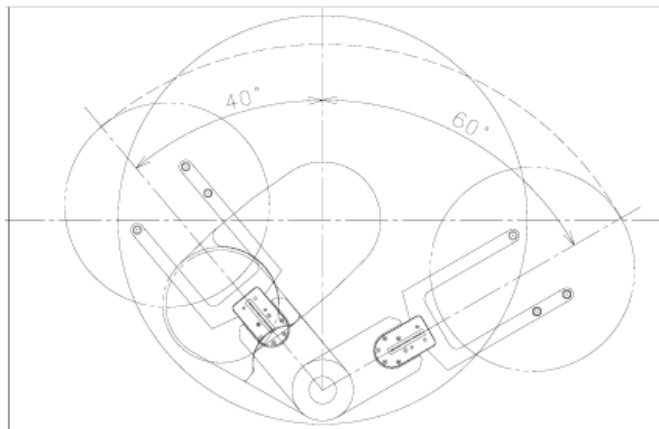
        (Message for the Completion of Execution)
              $ <UNo> <Sts> <Errcd> <Subcd> CSRV <Sum> <CR>

              · UNo: Unit number (1 byte)
                 · '1' or '2':   Unit specified
              · Sts: Status (2 byte)
              · Errcd: Error code (4 bytes)
                 · At normal completion: Responds "0000".
                 · At erroneous completion: Responds with the error code.
              · Subcd: Sub code (4 bytes)
                 · Detailed alarm information is shown.
                 · When either alarm does not occur, sub code is '0000'.

[Note]   IF Reference command is performed while executing CSRV command, the response may get delayed.

31. CCLR

[Name]	Error Release (Clear)

[Function] Clears the error status or error history of the specified unit.
Note: The error factor is not released with this command.   Also, some errors cannot be released.
The error with error code of less than 4000 cannot be released.
4000th alarms can be cleared only when all the alarm causes have been removed.

[Operation] Same as above.

[Conditions] <Releasing the error status>
· The specified unit is stopped.
· "Auto" mode
<Clearing the error history>
· The specified unit is stopped.
· "Auto" mode
· No error is occurring.

[Command Format]

$ <UNo> CCLR <CMode> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' to '2':   Unit specified
· CMode: Clear mode (1 byte)
· 'E'   :   Releases the error status.
· 'H':   Clears the error history.

[Response Format]
(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' or '2':   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
· At normal reception: Responds "0000".
· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> CCLR <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' or '2':   Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
· At normal completion: Responds "0000".
· At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

[Note]   IF Reference command is performed while executing CCLR command, the response may get delayed.

32. CSOL

[Name]	Solenoid Control Command for Workpiece Chucking
[Function]	Gives a solenoid control command for workpiece chucking to the specified unit.
[Operation]	Same as above.
[Conditions]	· The specified unit is stopped.
	· "Auto" mode

[Command Format]

$ <UNo> CSOL <End-Effector> <Sw> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2':   Unit specified
- · End-Effector: End-Effector specified (1 byte)
  - · 'A':   End-Effector 1, Pre-aligner
  - · 'B':   End-Effector 2
- · Sw: Chucking command (1 byte)
  - · '0':   Chucking OFF
  - · '1':   Chucking ON

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · At normal reception: Responds "0000".
  - · Reception rejected: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> CSOL <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Errcd: Error code (4 bytes)
  - · At normal completion: Responds "0000".
  - · At erroneous completion: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

33. CCHK

[Name]   Chucking Control Command

[Function]   Gives a chucking control command to the specified unit.

[Operation]   Chuck or Release instruction is output.
Chuck or Release completion is confirmed, and the execution complete message is returned.
The error is notified when operation is not completed while it is regulations time (i030/i031 parameter).

[Conditions]   · The specified unit is stopped.
· "Auto" mode

[Command Format]

$ <UNo> CCHK <End-Effector> <Sw> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2': Unit specified
· End-Effector: End-Effector specified (1 byte)
   · 'A':   End-Effector 1, Pre-aligner
   · 'B':   End-Effector 2
· Sw: Chucking command (1 byte)
   · '0':   Release the wafer
   · '1':   Chuck the wafer

[Response Format]

(Response Message)

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2': Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · At normal reception: Responds "0000".
   · Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
     -Control group data type
   · When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)

$ <UNo> <Sts> <Errcd> <Subcd> CCHK <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2': Unit specified
· Sts: Status (2 byte)
· Errcd: Error code (4 bytes)
   · At normal completion: Responds "0000".
   · At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

34. CLFT

[Name]   Lifter Control Command

[Function]   Gives a lifter control command to Pre-aligner

[Operation]   <In case of lift up>
        1) Unchuck wafer if the wafer is chucked.
        2) Raise the lifter
        3) Wait to turn on the Up signal , then send the abnormal motion end when time-out
        4) Confirm the Up signal and send the normal motion end

        <In case of Downing the lifter>
        1) Down the lifter
        2) Wait to turn on the Down signal, then send the abnormal motion end when time-out
        3) Confirm the Up signal and send the normal motion end

[Conditions]   · The specified unit is stopped.
          · "Auto" mode
          · No alarm is occurring.

[Command Format]
        $ <UNo> CLFT <Sw> <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '2':   Pre-aligner
        · Sw: Lifter Control(1 byte)
           · 'U':   Up
           · 'D':   Down

[Response Format]
    (Response Message)
        @ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '2':   Pre-aligner
         · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
           · At normal reception: Responds "0000".
           · Reception rejected: Responds with the error code.
        · Subcd: Sub code (4 bytes)
           · Detailed alarm information is shown.
           · When either alarm does not occur, sub code is '0000'.

    (Message for the Completion of Execution)
        $ <UNo> <Sts> <Errcd> <Subcd> CLFT <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '2':   Pre-aligner
        · Sts: Status (2 byte)
        · Errcd: Error code (4 bytes)
           · At normal completion: Responds "0000".
           · At erroneous completion: Responds with the error code.
        · Subcd: Sub code (4 bytes)
           · Detailed alarm information is shown.
           · When either alarm does not occur, sub code is '0000'.

35. SSPP

[Name]   Motion Speed Setting (in [%])

[Function] Sets the motion speed (first transfer speed, second transfer speed, and low speed) in the ratio ([%]) for the maximum speed for the specified axis of the specified unit.   Speed data to be set by this command is only for the volatile memory.   When the power supply is turned ON (or the hardware is reset), the speed data (refer to the "Parameter Lists for Transfer Manipulator") in the non-volatile memory is used as the initial value.   For the motion speeds, refer to "6.4 Automatic Speed Changing Function."

[Operation]   Same as above.

[Conditions] The specified unit is stopped.

[Command Format]

$ <UNo> SSPP <Axis> <SMode> <Value> <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1' or '2':   Unit specified
- · Axis: Axis (1 byte)
    - \<Manipulator>
    - · 'S':   Rotary axis
    - · 'E':   Expansion axis
    - · 'A':   End-Effector 1
    - · 'B':   End-Effector 2
    - · 'Z':   Lifting axis
    - · 'T':   Travel axis (with travel axis only)
    - · 'R':   Linear motion speed
    - · 'W':   Elbow switching motion speed
    - \< Pre-aligner>
    - · 'S':   Rotary axis
    - · 'X':   X-axis (for three-axis Pre-aligner only)
    - · 'Y':   Y-axis (for three-axis Pre-aligner only)
- · SMode: Speed mode (1 byte)
    - · 'H':   First transfer speed
    - · 'M':   Second transfer speed
    - · 'L':   Low speed
- · Value: Speed specified (4 bytes)
    - · Specified in the range between "0000" and "1000" (resolution: 0.1 [%])
    - · If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.

[Response Format]

(Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> SSPP <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
    - · At normal reception: Responds "0000".
    - · Reception rejected: Responds with the error code.
- · Subcd: Sub code (4 bytes)
    - · Detailed alarm information is shown.
    - · When either alarm does not occur, sub code is '0000'.

36. SPOS

[Name]   Registration of Current Position

[Function]   Registers the current position of the specified unit as the specified transfer station.   The memory type (volatile or non-volatile) to register the position data can be specified.
>
> · To register the data temporarily: Select the volatile memory.
> · To register the data permanently: Select the non-volatile memory.

The position data registered in the volatile memory is held until the power supply is turned OFF or the hardware is reset.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
> ·"Auto" mode
> · The specified unit is under servo ON status.

[Command Format]
>
> $ <UNo> SPOS <Mem> <TrsSt> <End-Effector> <Step> <Another End-Effector> <Mapp pos> <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1':   Manipulator
- Mem: Memory (1 byte)
  - 'V':   Volatile memory   (For temporary teaching)
  - 'N':   Non-volatile memory
- TrsSt: Transfer station (2 bytes)
  - "P1" to "P8"   : When the lowest-layer position of the cassette stage specified
  - "UA" to "UL"   : When transfer stage specified
  - "M1" to "M8"   : When Mapping Station specified
- End-Effector: End-Effector specified (1 byte)
  - 'A':   When End-Effector 1 specified
  - 'B':   When End-Effector 2 specified
- Step: Step specified (3 byte)
  - 'STA':   Station
  - 'RDY':   Ready
  - 'IM3':   Intermediate 3
  - 'IM2':   Intermediate 2
  - 'IM1':   Intermediate 1
  - 'MIN':   Minmum sweep
- Another End-Effector: Another End-Effector registration (1 byte)
  - 'A':   Update calculated data
  - 'N':   No update
- Mapp pos: Mapping position registration (1 byte)
  - 'Y':   Update calculated data
  - 'N':   No update

[Note]   Mapping position is valid only for cassette stage.

SPOS (Cont'd)

[Response Format]

   (Message for the Completion of Execution)

     <u>$</u> <u>\<UNo\></u> <u>\<Sts\></u> <u>\<Ackcd\></u> <u>\<Subcd\></u> <u>\<SPOS\></u> <u>\<Sum\></u> <u>\<CR\></u>

      · UNo: Unit number (1 byte)
       · '1': Unit specified
      · Sts: Status (2 byte)
      · Ackcd: Response code (4 bytes)
       · At normal reception: Responds "0000".
       · Reception rejected: Responds with the error code.
      · Subcd: Sub code (4 bytes)
       · Detailed alarm information is shown.
       · When either alarm does not occur, sub code is '0000'.

[Note]   Because the number of write-in times in the non-volatile memory is limited to 100,000 times, temporary registration must be made to the volatile memory.

37. SABS

[Name]   Registration of Coordinate Position

[Function]   Registers the coordinate position as the specified transfer station for the specified unit.   The memory type (volatile or non-volatile) to register the position data can be specified.
- · To register the data temporarily: Select the volatile memory.
- · To register the data permanently: Select the non-volatile memory.

The position data registered in the volatile memory is held until the power supply is turned OFF or the hardware is reset.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
· "Auto" mode

[Command Format]

$ <UNo> SABS <Mem> <TrsSt> <End-Effector> <Step>
                <Value1> . . .<ValueN> <Another End-Effector> <Mapp pos> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1':   Manipulator
- · Mem: Memory (1 byte)
  - · 'V':   Volatile memory,   · 'N':   Non-volatile memory
- · TrsSt: Transfer station (2 bytes)
  - · "P1" to "P8"   :   When the lowest-layer position of the cassette stage specified
  - · "UA" to "UL"   :   When transfer stage specified
  - · "M1" to "M8"   :   When Mapping Station specified
- · End-Effector: End-Effector specified (1 byte)
  - · 'A':   When End-Effector 1 specified
  - · 'B':   When End-Effector 2 specified
- · Step: Step specified (3 byte)
  - · 'STA':   Station,   · 'RDY':   Ready
  - · 'IM3':   Intermediate 3,   · 'IM2':   Intermediate 2
  - · 'IM1':   Intermediate 1,   · 'MIN':   Minmum sweep
- · ValueN: Coordinate (6 bytes each)
  - · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm], [deg])
  - · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
  - · The sign must be added to the highest digit.
  - · Be sure to specify as many as the number of the axes for the specified unit.
    - Value 1:   Rotary Axis   Value 2:   Expansion Axis   Value 3:   End-Effector 1
    - Value 2:   End-Effector 2   Value 5:   Lifting Axis
- · Another End-Effector: Another End-Effector registration (1 byte)
  - · 'A':   Update (Another End-Effector position)
  - · 'N':   No update
- · Mapp pos: Mapping position registration (1 byte)
  - · 'Y':   Update (mapping position)
  - · 'N':   No update

[Note]   Mapp pos is invalid excluding the cassette stage specification.

SABS (Cont'd)

[Response Format]

(Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> <SABS> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1':   Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · At normal reception: Responds "0000".
  - · Reception rejected: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

[Note]   Because the number of write-in times in the non-volatile memory is limited to 100,000 times, temporary registration must be made to the volatile memory.

38. SPSV

[Name]   Position Data Save

[Function]   Registers all the position data in the volatile memory to the non-volatile memory.

[Operation]   Same as above.

[Conditions] · The specified unit is stopped.
             · "Auto" mode

[Command Format]
        $ \<UNo> SPSV \<TrsSt> \<End-Effector> \<Sum> \<CR>

        · UNo: Unit number (1 byte)
           · '1':   Manipulator
        · TrsSt: Transfer station (2 bytes)
           · "FF":   Fixed value
        · End-Effector: End-Effector specified (1 byte)
           · 'F':   Fixed value

[Response Format]
      (Message for the Completion of Execution)
        $ \<UNo> \<Sts> \<Ackcd> \<Subcd> \<SPSV> \<Sum> \<CR>

        · UNo: Unit number (1 byte)
           · '1':   Unit specified
        · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
           · At normal reception:Responds "0000".
           · Reception rejected: Responds with the error code.
        · Subcd: Sub code (4 bytes)
           · Detailed alarm information is shown.
           · When either alarm does not occur, sub code is '0000'.

39. SOFS

[Name]	Transfer Offset Setting

[Function] Sets an aribitrary transfer offset for the workpiece Get/Put motion for each transfer station. The memory type (volatile or non-volatile) to register the data can be specified.
  · To register the data temporarily: Select the volatile memory.
  · To register the data permanently: Select the non-volatile memory.
The transfer offset data registered in the volatile memory is held until the power supply is turned OFF or the hardware is reset. When the power supply is turned ON (or the hardware is reset), the transfer offset value (refer to "Parameter Lists for Transfer Manipulator") in the non-volatile memory is used as the initial value.

[Operation] Same as above.

[Conditions] · The specified unit is stopped.
  · "Auto" mode

[Command Format]
  $ <UNo> SOFS <Mem> <TrsSt> <Offset1> <Offset2> . . . <Offset9> <Sum> <CR>

  · UNo: Unit number (1 byte)
    · '1': Manipulator
  · Mem: Memory (1 byte)
    · 'V':   Volatile memory
    · 'N':   Non-volatile memory
  · TrsSt: Transfer station (2 bytes)
    · "P1" to "P8"   : When cassette stage specified
    · "UA" to "UL"   : When transfer stage specified
  · Offset: Transfer offset (4 bytes)
    · Offset1:   Downward offset
    · Offset2:   Upward offset
    · Offset3:   Extending direction offset
    · Offset4:   Contracting direction offset
    · Offset5:   Downward offset for PUT
    · Offset6:   Contracting direction offset for GET
    · Offset7:   Extending direction offset for GET
    · Offset8:   Upward offset for GET
    · Offset9:   Offset for PUT

    · Specified in the range between "0000" and "9999" (resolution: 0.01 [mm])
    · If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.

[Response Format]
  (Message for the Completion of Execution)
      $ <UNo> <Sts> <Ackcd> <Subcd> SOFS <Sum> <CR>

  · UNo: Unit number (1 byte)
    · '1':   Manipulator
  · Sts: Status (2 byte)
  · Ackcd: Response code (4 bytes)
    · At normal reception:Responds "0000".
    · Reception rejected: Responds with the error code.
  · Subcd: Sub code (4 bytes)
    · Detailed alarm information is shown.
    · When either alarm does not occur, sub code is '0000'.

[Note] Because the number of write-in times in the non-volatile memory is limited to 100,000 times, temporary registration must be made to the volatile memory.

40. SPIT

[Name]	Setting of Pitch Distance between Slots

[Function]   Sets the pitch distance between slots for each cassette stage. The memory type (volatile or non-volatile) to register the data can be specified.

    · To register the data temporarily: Select the volatile memory.
    · To register the data permanently: Select the non-volatile memory.

The pitch distance between slots registered in the volatile memory is held until the power supply is turned OFF or the hardware is reset. When the power supply is turned ON (or the hardware is reset), the pitch distance between slots (refer to "Parameter Lists for Transfer Manipulator") in the non-volatile memory is used as the initial value.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
    · "Auto" mode
    · Mode to create the pitch distance between slots is the arbitrary setting mode (optional).

[Command Format]

    $ <UNo> SPIT <Mem> <TrsSt> <Value> <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1':   Manipulator
    · Mem: Memory (1 byte)
      · 'V':   Volatile memory
      · 'N':   Non-volatile memory
    · TrsSt: Transfer station (2 bytes)
      · "P1" to "P8":   When cassette stage specified
    · Value: Pitch distance between slots (6 bytes)
      · Specified in the range between "000000" and "999999" (resolution: 0.0001 [mm])
      · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.

[Response Format]

    (Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> SPIT <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1':   Unit specified
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
      · At normal reception: Responds "0000".
      · Reception rejected: Responds with the error code.
    · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
      · When either alarm does not occur, sub code is '0000'.

[Note]   Because the number of write-in times in the non-volatile memory is limited to 100,000 times, temporary registration must be made to the volatile memory.

41. SSLT

[Name]   Setting of Number of Slots

[Function]   Sets any number of slots for each cassette stage. The memory type (volatile or non-volatile) to register the data can be specified.
- To register the data temporarily: Select the volatile memory.
- To register the data permanently: Select the non-volatile memory.

The number of slots registered in the volatile memory is held until the power supply is turned OFF or the hardware is reset. When the power supply is turned ON (or the hardware is reset), the number of slots (refer to the "Parameter Lists for Transfer Manipulator") in the non-volatile memory is used as the initial value.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
               · "Auto" mode

[Command Format]

$ <UNo> SSLT <Mem> <TrsSt> <SlotNo> <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1':   Manipulator
- Mem: Memory (1 byte)
  - 'V':   Volatile memory
  - 'N':   Non-volatile memory
- TrsSt: Transfer station (2 bytes)
  - "P1" to "P8":   When cassette stage specified
- SlotNo: Number of slots (2 bytes)
  - Specified in the range between "01" and "30"
  - If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.

[Response Format]

(Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> SSLT <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1':   Unit specified
- Sts: Status (2 byte)
- Ackcd: Response code (4 bytes)
  - At normal reception:Responds "0000".
  - Reception rejected: Responds with the error code.
- Subcd: Sub code (4 bytes)
  - Detailed alarm information is shown.
  - When either alarm does not occur, sub code is '0000'

[Note]   Because the number of write-in times in the non-volatile memory is limited to 100,000 times, temporary registration must be made to the volatile memory.

42. SRSV

[Name]   Parameter Save

[Function]   The transfer offset and cassette information (mode to create pitch distance between slots (optional), pitch distance between slots, and the number of slots) in the volatile memory are preserved in the non-volatile memory.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
                · "Auto" mode

[Command Format]
        $ <UNo> SRSV <PrmNo> <TrsSt> <Sum> <CR>

        · UNo: Unit number (1 byte)
          · '1':   Manipulator
        · Prm: Parameter type (2 bytes)
          · "FF":   Fixed value
        · TrsSt: Transfer station (2 bytes)
          · "FF":   Fixed value

[Response Format]
    (Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> SRSV <Sum> <CR>

        · UNo: Unit number (1 byte)
          · '1':   Unit specified
        · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
          · At normal reception:Responds "0000".
          · Reception rejected: Responds with the error code.
        · Subcd: Sub code (4 bytes)
          · Detailed alarm information is shown.
          · When either alarm does not occur, sub code is '0000'.

43. SMSK

[Name]   Setting of Interlock Monitoring Enabled/Disabled

[Function]   Sets whether a monitoring function is enabled or disabled.   This setting is registered only in the volatile memory. All monitoring functions are enabled when the power supply is turned ON (or the hardware is reset).

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
               · "Auto" mode

[Command Format]
            $ <UNo> SMSK <Valid1> . . . <Valid4> <Sum> <CR>

       · UNo: Unit number (1 byte)
          · '1' or '2':   Unit specified
       · ValidN: Interlock information (4 bytes)
          · For details, refer to the next page.

[Response Format]
       (Message for the Completion of Execution)
          $ <UNo> <Sts> <Ackcd> <Subcd> SMSK <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '1' or '2':   Unit specified
        · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
          · At normal reception : Responds "0000".
          · Reception rejected : Responds with the error code.
        · Subcd: Sub code (4 bytes)
          · Detailed alarm information is shown.
          · When either alarm does not occur, sub code is '0000'.

<Interlock Information>
Set the interlock information using the ASCII code of the total value (hexadecimal value) of each status value.
<Manipulator>

	Bit No.	Contents	Setting
Valid1	Bit0	End-Effector 1 Wafer Presence Monitoring	
	Bit1	End-Effector 2 Wafer Presence Monitoring	
	Bit2	Customized handshake	
	Bit3	Open for future use	
Valid2	Bit0	Access Authorization 1 Interlock Signal	
	Bit1	Access Authorization 2 Interlock Signal	
	Bit2	Access Authorization 3 Interlock Signal	
	Bit3	Access Authorization 4 Interlock Signal	
Valid3	Bit0	Access Authorization 5 Interlock Signal	
	Bit1	Access Authorization 6 Interlock Signal	
	Bit2	Access Authorization 7 Interlock Signal	0: Enabled
	Bit3	Access Authorization 8 Interlock Signal	1: Disabled
Valid4	Bit0	Pre-aligner operating interlock (when the manipulator accesses to the Pre-aligner stage)	
	Bit1	Pre-aligner workpiece chucking interlock (when the manipulator accesses to the Pre-aligner stage)	
	Bit2	Pre-aligner lifter interlock (lifter up) (when the manipulator with straight type effector accesses to the Pre-aligner stage)	
	Bit3	Pre-aligner workpiece presence interlock (when the manipulator puts the wafer on the Pre-aligner stage)	

SMSK (Cont'd)

<Pre-aligner>

	Bit No.	Contents	Setting
Valid1	Bit0	Pre-aligner workpiece chucking interlock *1 (when alignment)	0: Enabled 1: Disabled
	Bit1	Open for future use	
	Bit2	Open for future use	
	Bit3	Open for future use	
Valid2	Bit0	Open for future use	
	Bit1	Open for future use	
	Bit2	Open for future use	
	Bit3	Open for future use	
Valid3	Bit0	Open for future use	
	Bit1	Open for future use	
	Bit2	Open for future use	
	Bit3	Open for future use	
Valid4	Bit0	Open for future use	
	Bit1	Open for future use	
	Bit2	Open for future use	
	Bit3	Open for future use	

When 1 is set, chuck check for pre-aligner is disabled; enabled when 0 is set.

44. SPRM

[Name]   Setting the parameters

[Function]   Setting the user parameters

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
                 · The specified unit is under servo OFF state.
                 · "Auto" mode

[Command Format]
        $ <UNo> SPRM <Type> <Sign> <Num> <Value> <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '1' or '2':   Unit specified
        · Type: Parameter type (1 byte)
           · 'I':   Integer type parameter
           · 'R':   Real-number type parameter
        · Sign: (1 byte)
           · 'U':   User parameter only   (System parameter is protected)
        · Num: Parameter number(3 byte)
           Refer to Parameter Lists for Wafer Transfer Manipulator
        · Value: Parameter value(10 byte)
           · Type = 'I':   Specified in the range between "-000032768" and "0000032767"
           · Type = 'R':   Specified in the range between "-999999999" and "9999999999"(resolution: 0.0001)
           * If a value is less than 10 digits, add '0' to higher digit value so that it can always be specified in 10 digits.

[Response Format]
      (Message for the Completion of Execution)
           $ <UNo> <Sts> <Ackcd> <Subcd> SPRM <Sum> <CR>

        · UNo: Unit number (1 byte)
           · '1' or '2':   Unit specified
        · Sts: Status (2 byte)
        · Ackcd: Response code (4 bytes)
           · At normal reception:Responds "0000".
           · Reception rejected: Responds with the error code.
        · Subcd: Sub code (4 bytes)
           · Detailed alarm information is shown.
           · When either alarm does not occur, sub code is '0000'.

45. SALM

[Name]   Setting alignment mode

[Function]   The alignment mode is selected from among Accuracy mode, Medium mode and Throughput mode.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
              · "Auto" mode

[Command Format]
$ <UNo> SALM <Mem> <PMode> <ALMode> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '2':   Pre-aligner (Not selectable)
- · Mem: Memory used (1 byte)
  - · 'V':   Volatile memory
  - · 'N':   Nonvolatile memory
- · PMode: Positioning mode (1 byte)
  - · 'F':   Non-shortcut positioning
- · ALMode: Alignment Mode (1 byte)
  - · 'A':   Accuracy mode
  - · 'M':   Medium mode
  - · 'S':   Throughput mode

[Response Format]
(Message for the Completion of Execution)
$ <UNo> <Sts> <Ackcd> <Subcd> SPRM <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2':   Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · At normal reception: Responds "0000".
  - · Reception rejected: Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

46. SSTD

[Name]   Setting standard position

[Function]   Setting Standard Position.

[Operation]   Set Standard position of Manipulator and Pre-aligner.

[Conditions]   · The specified unit is stopped.
                · "Auto" mode

[Command Format]
$ <UNo> SSTD <Value1> … <ValueN> <Sum><CR>

- UNo: Unit number (1 byte)
  - '1' or '2': Unit specified
- ValueN: Position data(11 bytes)

In case the value is less than 11 figures, set 0 in the head.
    <Manipulator>
    Value1: S axis   Value2: R axis   Value3: End-Effector1 axis
    Value4: End-Effector 2 axis   Value5: Z axis
    <Pre-aligner>
    Value1: S axis   Value2: X axis   Valu3: Y axis

If <Value1> is 'CURRENT POS', register the current position as Standard Position.
Ignore the other value <2-5>. However, the number of characters shall be prescribed.

[Response Format]
(Message for the Completion of Execution)
$ <UNo> <Sts> <Ackcd> <Subcd> SSTD <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1' or '2': Unit specified
- Sts: Status (2 byte)
- Ackcd: Response code (4 bytes)
  - At normal reception:Responds "0000".
  - Reception rejected: Responds with the error code.
- Subcd: Sub code (4 bytes)
  - Detailed alarm information is shown.
  - When either alarm does not occur, sub code is '0000'.

47. SWSZ

[Name]   Setting wafer size

[Function]   Setting the wafer size to be used.

[Operation]   Setting the wafer size among 150mm, 200mm, 300mm.
Manipulator moves according to the setting wafer wise.
Set to Manipulator parameter i102, i103/Pre-aligner parameter i102.

[Conditions]   · The specified unit is stopped.
                · "Auto" mode

[Command Format]
        $ <UNo> SWSZ <End-Effector> <Size> <Sum><CR>

        · UNo: Unit number(1 byte)
           · '1' or '2':   Unit specified
        · End-Effector: End-Effector(1 byte)
           · 'A':   End-Effector 1 /Pre-aligner
           · 'B':   End-Effector 2
        · Size: Diameter of the wafer(3 bytes)
           · '300':   300mm
           · '200':   200mm
           · '150':   150mm

[Response Format]
     (Positive Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> SWSZ <Sum> <CR>
       · UNo: Unit number (1 byte)
          · '1' or '2':   Unit specified
       · Sts: Status (2 bytes)
       · Ackcd: Response code (4 bytes)
          · At normal reception:Responds "0000".
          · Reception rejected: Responds with the error code.
       · Subcd: Sub code (4 bytes)
          · Detailed alarm information is shown.
          · When either alarm does not occur, sub code is '0000'.

48. RSPP

[Name]   Motion Speed Reference (in [%])

[Function]   Refers to the motion speed (first transfer speed, second transfer speed, and low speed) in the ratio ([%]) for the maximum speed for the specified axis of the specified unit. The speed data to be referred to by this command is the current value set in the volatile memory.
For the motion speed, refer to "6.4 Automatic Speed Changing Function."

[Operation]   Same as above.

[Conditions]   None

[Command Format]

    $ <UNo> RSPP <Axis> <SMode> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2': Unit specified
- · Axis: Axis (1 byte)
  - \<Manipulator\>
  - · 'S': Rotary axis
  - · 'E': Expansion axis
  - · 'A': End-Effector 1
  - · 'B': End-Effector 2
  - · 'Z': Lifting axis
  - · 'T': Travel axis (with travel axis only)
  - · 'R': Linear moition speed
  - · 'W': Elbow switching motion speed
  - \<Pre-aligner\>
  - · 'S': Rotary axis
  - · 'X': X-axis (for three-axis Pre-aligner only)
  - · 'Y': Y-axis (for three-axis Pre-aligner only)
- · SMode   : Speed mode (1 byte)
  - · 'H': First transfer speed
  - · 'M': Second transfer speed
  - · 'L': Low speed

[Response Format]

    (Positive Message for the Completion of Execution)

        $ <UNo> <Sts> <Ackcd> <Subcd> RSPP <Value> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2': Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds "0000".
- · Subcd: Sub code (4 bytes)
  - · Responds "0000".
- · Value: Speed [%] (4 bytes)
  - · Specified in the range between "0000" and "1000" (resolution: 0.1 [%])
  - · If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.

RSPP (Cont'd)

(Negative Message for the Completion of Execution)

    $ <u>UNo</u> <u>Sts</u> <u>Ackcd</u> RSPP <u>Sum</u> <u>CR</u>

    · UNo: Unit number (1 byte)

        · '1' or '2':   Unit specified by command

    · Sts: Status (2 byte)

    · Ackcd: Response code (4 bytes)

        · Responds with the error code.

    · Subcd: Sub code (4 bytes)

        · Detailed alarm information is shown.

        · When either alarm does not occur, sub code is '0000'.

49. RPOS

[Name]	Current Position Reference/Registered Position Reference
[Function]	Refers to the current position or registered position for the specified unit. The registered position to be referred to by this command is the value set in the volatile memory.
[Operation]	Same as above.
[Conditions]	<When referring to the registered position> · The position has been registered. (STA,RDY,and MIN must be registered)

[Command Format]

$ <UNo> RPOS <TrsSt> <Step> <End-Effector> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2'   : Unit specified
· TrsSt: Transfer station (2 bytes)
   · "P1" to "P8"   :   When the lowest-layer position of the cassette stage specified
   · "UA" to "UL"   :   When transfer stage specified
   · "FF"   :   When current position specified
   <Three-axis Pre-aligner>
   · "FF":   When current position specified
· Step: Step specified (3 byte)
   · 'STA':   Station
   · 'RDY':   Ready
   · 'IM3':   Intermediate 3
   · 'IM2':   Intermediate 2
   · 'IM1':   Intermediate 1
   · 'MIN':   Minmum sweep
   · 'FFF':   When current position specified
   <Three-axis Pre-aligner>
   · "FFF": When current position specified
· End-Effector: End-Effector specified (1 byte)
   · 'A':   When End-Effector 1 specified
   · 'B':   When End-Effector 2 specified
   Note: Fixed to 'A' when the current position is specified at <TrsSt>

[Response Format]

(Positive Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> RPOS <Value1> . . . <ValueN> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2':   Unit specified by command
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · Responds "0000".
· Subcd: Sub code (4 bytes)
   · Responds "0000".
· Value: Coordinate (6 bytes each)
   · Specified in the range between "-99999" and "999999" (resolution: 0.01 [mm], [deg])
   · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
   · The sign is added to the highest digit.
   · Be sure to respond to as many as the number of the axes for the specified unit.
   · Rotary axis of Pre-aligner is always "000000".
      Value 1: Rotary Axis   Value 2: Expansion Axis   Value 3: End-Effector 1
      Value 2: End-Effector 2   Value 5: Lifting Axis

RPOS (Cont'd)

(Negative Message for the Completion of Execution)
$ <UNo> <Sts> <Ackcd> <Subcd> RPOS <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1' or '2': Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

50. ROFS

[Name]   Transfer Offset Reference

[Function]   Refers to the transfer offset value for workpiece Get/Put motion for each transfer station of the specified unit.   The value to be referred to by this command is the value that is currently set in the volatile memory.

[Operation]   Save as above.

[Conditions] None

[Command Format]

$ <UNo> ROFS <TrsSt> <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1': Manipulator
- · TrsSt: Transfer station (2 bytes)
    - · "P1" to "P8"   :   When cassette stage specified
    - · "UA" to "UL"   :   When transfer stage specified

[Response Format]

(Positive Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> ROFS <Offset1> <Offset2> . . . <Offset9> <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1': Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
    - · Responds "0000".
- · Subcd: Sub code (4 bytes)
    - · Responds "0000".
- · OffsetN: Transfer offset (4 bytes each)
    - · Offset1:   Downward offset
    - · Offset2:   Upward offset
    - · Offset3:   Extending direction offset
    - · Offset4:   Contracting direction offset
    - · Offset5:   Downward offset for PUT
    - · Offset6:   Contracting direction offset for GET
    - · Offset7:   Extending direction offset for GET
    - · Offset8:   Upward offset for GET
    - · Offset9:   Offset for PUT
- * Specified in the range between "0000" and "9999" (resolution: 0.01 [mm])
- * If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> ROFS <Sum> <CR>

- · UNo: Unit number (1 byte)
    - · '1':   Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
    - · Responds with the error code.
- · Subcd: Sub code (4 bytes)
    - · Detailed alarm information is shown.
    - · When either alarm does not occur, sub code is '0000'.

51. RCST

[Name]   Reference to Information on Cassette Stage

[Function]   Refers to the information on the specified cassette stage (mode to create pitch distance between slots (optional), pitch distance between slots, and the number of slots). The value to be referred to by this command is the value that is currently set in the volatile memory.

[Operation]   Same as above.

[Conditions]   None

[Command Format]

$ <UNo> RCST <TrsSt> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Manipulator
· TrsSt: Transfer station (2 bytes)
   · "P1" to "P8":   When cassette stage specified

[Response Format]

(Positive Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> RCST <Value> <SlotNo> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified by command
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · Responds "0000".
· Subcd: Sub code (4 bytes)
   · Responds "0000".
· Value: Pitch distance between slots (6 bytes)
   · Specified in the range between "000000" and "999999" (resolution: 0.0001 [mm])
   · If a value is less than 6 digits, add '0' to higher digit value so that it can always be specified in 6 digits.
· SlotNo: Number of slots (2 bytes)
   · Responds in the range between "01" and "30."
   · If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> RCST <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1':   Unit specified by command
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
   · Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

52. RMAP

[Name]   Mapping Result Reference

[Function]   Refers to the result of mapping for the specified transfer station.

[Operation]   Same as above.

[Conditions] Mapping has been completed normally.

[Command Format]
> $ <UNo> RMAP <TrsSt> <SlotNo> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1': Manipulator
- · TrsSt: Transfer station (2 bytes)
  - · "P1" to "P8"   :   When cassette stage specified
- · SlotNo: Slot number (2 bytes)
  - · "FF": When all slots specified
  - · "01" to "XX": When cassette stage specified by <TrsSt>
    (The maximum value of "XX" is in ASCII code of the number of slots specified by parameter.)

[Response Format]
(Positive Message for the Completion of Execution)
> $ <UNo> <Sts> <Ackcd> <Subcd> RMAP 01 <Result> 02 <Result> . . . XX <Result> <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1': Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds "0000".
- · Subcd: Sub code (4 bytes)
  - · Responds "0000".
- · "01" to "XX": Slot number (2 bytes)
  - ·Specified in the range between "01" and "XX."
    (The maximum value of "XX" is in the ASCII code of the number of slots set in the volatile memory.)
  - · If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.
- · Result: Mapping result (2 bytes)
  - · "OK": Workpiece inserted correctly
  - · "ER": Workpiece inserted incorrectly (cross slot)
  - · "W2": Workpiece inserted incorrectly (double)
  - · "--":   No workpiece provided

(Negative Message for the Completion of Execution)
> $ <UNo> <Sts> <Ackcd> <Subcd> RMAP <Sum> <CR>

- · UNo: Unit number (1 byte)
  - · '1': Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

53. RSTS

[Name]   Reference to Various Statuses

[Function]   Refers to various statuses.

[Operation]   Same as above.

[Conditions]   None

[Command Format]

     $ <UNo> RSTS <Sum> <CR>

     · UNo: Unit number (1 byte)
       · '1' or '2':   Unit specified

[Response Format]

(Positive Message for the Completion of Execution)

     $ <UNo> <Sts> <Ackcd> <Subcd> RSTS <Errcd> <Subcd> <Status1> . . . <StatusN> <Sum> <CR>

     · UNo: Unit number (1 byte)
       · '1' or '2':   Unit specified by command
     · Sts: Status (2 byte)
     · Ackcd: Response code (4 bytes)
       · Responds "0000".
     · Subcd: Sub code (4 bytes)
       · Responds "0000".
     · Errcd: Error code (4 bytes)
       · Responds with the error code that is currently occurred for the specified unit ("0000" when no error occurs).
     · Subcd: Sub code (4 bytes)

(Negative Message for the Completion of Execution)

     $ <UNo> <Sts> <Ackcd> <Subcd> RSTS <Sum> <CR>

     · UNo: Unit number (1 byte)
       · '1' or '2':   Unit specified by command
     · Sts: Status (2 byte)
     · Ackcd: Response code (4 bytes)
       · Responds with the error code.
     · Subcd: Sub code (4 bytes)
       · Detailed alarm information is shown.
       · When either alarm does not occur, sub code is '0000'.

<Status Information>
  Set the status information using the ASCII code of the total value (hexadecimal value) of each status value.

<Manipulator>

	Bit No.	Contents	Setting
Status1	Bit0	Workpiece status for the End-Effector 1	0: Workpiece provided 1: Workpiece not provided
	Bit1	Workpiece status for the End-Effector 2	0: Workpiece provided 1: Workpiece not provided
	Bit2	Chucking status for the End-Effector 1	0: Workpiece released 1: Holding workpiece
	Bit3	Chucking status for the End-Effector 2	0: Workpiece released 1: Holding workpiece
Status2	Bit0	Access Authorization 1 Interlock Signal	0: Close 1: Open
	Bit1	Access Authorization 2 Interlock Signal	0: Close 1: Open
	Bit2	Access Authorization 3 Interlock Signal	0: Close 1: Open
	Bit3	Access Authorization 4 Interlock Signal	0: Close 1: Open
Status3	Bit0	Access Authorization 5 Interlock Signal	0: Close 1: Open
	Bit1	Access Authorization 6 Interlock Signal	0: Close 1: Open
	Bit2	Access Authorization 7 Interlock Signal	0: Close 1: Open
	Bit3	Access Authorization 8 Interlock Signal	0: Close 1: Open
Status4	Bit0	Customized handshake signal IN1 (Di28)	Refer to bellow
	Bit1	Customized handshake signal IN2 (Di29)	Refer to bellow
	Bit2	Customized handshake signal IN3 (Di30)	Refer to bellow
	Bit3	Open for future use	0

<Pre-aligner>

	Bit No.	Contents	Setting
Status1	Bit0	Workpiece status for the Pre-aligner(judge by Vacuum sensor)	0: Workpiece provided 1: Workpiece not provided
	Bit1	Open for future use	0
	Bit2	Chucking status for the Pre-aligner	0: Workpiece released 1: Holding workpiece
	Bit3	Open for future use	0
Status2	Bit0	Open for future use	0
	Bit1	Open for future use	0
	Bit2	Open for future use	0
	Bit3	Open for future use	0
Status3	Bit0	Open for future use	0
	Bit1	Open for future use	0
	Bit2	Open for future use	0
	Bit3	Open for future use	0
Status4	Bit0	Pre-aligner Lifter UP	0: No 1: Yes
	Bit1	Pre-aligner Lifter DOWN	0: No 1: Yes
	Bit2	Workpiece presence on the Pre-aligner(judge by CCD)	0: Presence, 1: Absence
	Bit3	Open for future use	0

54. RERR

[Name]   Error History Reference

[Function]   This command can refer to error history with time stamp data for each.

[Operation]   When NXC100 receives this command, 8 error codes with sub code and time stamp will be sent back to host according to specified block number.   Total 32 error codes in history can be referred by RERR.

[Conditions]   None

[Command Format]

$ <UNo> RERR <BlkNo> <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1' or '2': Unit specified
- BlkNo: Block Number (1 byte)
  - 0 ~ 3   (4 blocks for 32 error codes in alarm history)

[Response Format]

(Positive Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> RERR <BlkNo> <Error 1> ... <Error 8> <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1' to '2': Unit specified
- Sts: Status (2 byte)
- Ackcd: Response code (4 bytes)
  - Responds "0000"
- Subcd: Response code (4 bytes)
  - Responds "0000"
- BlkNo: Block number (1 byte)
  - Responds in the range between "0" and "3"
- Error X: Error code data <28 bytes>

  MM/DD/YY HH:MM:SS <Errcd>[<Subcd>]

  Ex.: 02/24/04 12:34:56 1234[5678]

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> RERR <Sum> <CR>

- UNo: Unit number (1 byte)
  - '1' or '2': Unit specified by command
- Sts: Status (2 byte)
- Ackcd: Response code (4 bytes)
  - Responds with the error code.
- Subcd: Sub code (4 bytes)
  - Detailed alarm information is shown.
  - When either alarm does not occur, sub code is '0000'.

55. RMSK

[Name]   Reference to Information on Interlock Monitoring

[Function]   Refers to the information on the current interlock monitoring for the specified unit.

[Operation]   Same as above.

[Conditions]   None

[Command Format]
> $ <UNo> RMSK <Sum> <CR>

> · UNo: Unit number (1 byte)
> > · '1' or '2':   Unit specified

[Response Format]
> (Positive Message for the Completion of Execution)
> > $ <UNo> <Sts> <Ackcd> <Subcd> RMSK <Valid1> . . . <Valid4> <Sum> <CR>

> > · UNo: Unit number (1 byte)
> > > · '1' or '2':   Unit specified
> > · Sts: Status (2 byte)
> > · Ackcd: Response code (4 bytes)
> > > · Responds "0000".
> > · Subcd: Sub code (4 bytes)
> > > · Responds "0000".
> > · ValidN: Information on interlock monitoring (4 bytes)
> > > · For details, refer to "35. SMSK."

> (Negative Message for the Completion of Execution)
> > $ <UNo> <Sts> <Ackcd> <Subcd> RMSK <Sum> <CR>

> > · UNo: Unit number (1 byte)
> > > · '1' or '2':   Unit specified by command
> > · Sts: Status (2 byte)
> > · Ackcd: Response code (4 bytes)
> > > · Responds with the error code.
> > · Subcd: Sub code (4 bytes)
> > > · Detailed alarm information is shown.
> > > · When either alarm does not occur, sub code is '0000'.

56. RVER

[Name]   Software Version Reference

[Function]   Refers to the Controller's software version.

[Operation]   Same as above.

[Conditions]   None

[Command Format]

        $ <UNo> RVER <Sum> <CR>

     · UNo: Unit number (1 byte)
        · '1' or '2':   Unit specified

[Response Format]

   (Positive Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> RVER <Version> <Sum> <CR>

     · UNo: Unit number (1 byte)
        · '1' or '2':   Unit specified by command
     · Sts: Status (2 byte)
     · Ackcd: Response code (4 bytes)
        · Responds "0000".
     · Subcd: Sub code (4 bytes)
        · Responds "0000".
     · Version: Version information (16 bytes)
        · "NXC100 V1.00   "

   (Negative Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> RVER <Sum> <CR>

     · UNo: Unit number (1 byte)
        · '1' or '2':   Unit specified by command
     · Sts: Status (2 byte)
     · Ackcd: Response code (4 bytes)
        · Responds with the error code.
     · Subcd: Sub code (4 bytes)
        · Detailed alarm information is shown.
        · When either alarm does not occur, sub code is '0000'.

57. RCFG

[Name]   Reference to configuration information

[Function]   Refers to the configuration information.

[Operation]   Same as above.

[Conditions] None

[Command Format]

        $ <UNo> RCFG <Sum> <CR>

    · UNo: Unit number (1 byte)
       · '1':   Manipulator

[Response Format]

  (Positive Message for the Completion of Execution)

      $ <UNo> <Sts> <Ackcd> <Subcd> RCFG <Ver> <RB-Modl> <RB-Uno> <RB-AxisN> <RB-LEE> <RB-UEE>
                    <RB-LEEType> <RB-UEEType> <RB-LEESize> <RB-UEESize> <RB-Map>
                    <RB-MapNum> <RB-Track> <RB-PA> <PA-Modl> <PA-Uno> <PA-AxisN>
                    <PA-Type> <PA-Size> <Sts-Attr> <Sum> <CR>

    · UNo: Unit number (1 byte)
       · '1':   Unit specified by command
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
       · Responds "0000".
    · Subcd: Sub code (4 bytes)
       · Responds "0000".
    · Ver: Firmware Version (16 bytes)
    · RB-Modl: Robot Model (16 bytes)
    · RB-Uno: Robot Unit Number. (1 bytes)
       · '1': Robot
    · RB-AxisN: Robot number of axis (1 bytes)
    · RB-LEE: Robot lower end effector (1 bytes)
       · '0': Disable, '1': Enable
    · RB-UEE: Robot upper End-Effector (1 bytes)
       · '0': Disable, '1': Enable
    · RB-LEEType: Robot lower End-Effector type (1 bytes)
       · 'E': Edge Grip, 'F': End-Effector Type Vaccum, 'S': Straight Vacuum
    · RB-UEEType: Robot upper End-Effector type (1 bytes)
       · 'E': Edge Grip, 'F': End-Effector Type Vaccum, 'S': Straight Vacuum
    · RB-LEESize: Robot lower End-Effector size (3 bytes)
       · '300': 300mm, '200': 200mm, '150': 150mm
    · RB-UEESize: Robot upper End-Effector size (3 bytes)
       · '300': 300mm, '200': 200mm, '150': 150mm
    · RB-Map: Robot mapping presence (1 bytes)
       · '0': Disable, '1': Enable
    · RB-MapNum: Robot holding mapping number (1 bytes)
       · '1': Lower End Effector, '2': Upper End Effector
    · RB-Track: Robot Track axis presence (1 bytes)
       · '0': Disable, '1': Enable
    · RB-PA: Robot Pre-aligner presence (1 bytes)
       · '0': Disable, '1': Enable

]

    · PA-Modl: Pre-aligner Model (16 bytes)
    · PA-Uno: Pre-aligner Unit Number. (1 bytes)
       · '2': Pre-aligner
    · PA-AxisN: Pre-aligner number of axis (1 bytes)
    · PA-Type: Pre-aligner Type (1 bytes)
       · '1': Yaskawa Edge Grip,   '2': 1 axis Vacuum,   '3': 3 axes w/o lift,   '4': 3 axes w/ lift
    · PA-Size: Pre-aligner Size (3 bytes)
       · '300': 300mm,   '200': 200mm,   '150': 150mm
    · Sts-Attr: Station Attribute (P1---UL) (20 bytes)
        · 'N': None,   'R': Radial,   'L': Linear

(Negative Message for the Completion of Execution)
    $ <UNo> <Sts> <Ackcd> <Subcd> RCFG <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '1':   Unit specified by command
    · Sts: Status (2 byte)
    · Ackcd: Response code (4 bytes)
      · Responds with the error code.
    · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
      · When either alarm does not occur, sub code is '0000'.

[Note]   If Pre-aligner does not exist, <PA-Model>---<PA-Size> are all '0'.

      Each status is set by the following parameters.

· RB-LEE	: (Manipulator)IS006[1]	· RB-UEE	: (Manipulator)IS007[1]
· RB-LEEType	: (Manipulator)I100	· RB-UEEType	: (Manipulator)I101
· RB-LEESize	: (Manipulator)I102	· RB-UEESize	: (Manipulator)I103
· RB-Map	: (Manipulator)IS008[1]	· RB-Track	: (Manipulator)IS009[1]
· RB-PA	: (Manipulator)I249	· PA-AxisN	: (Pre-aligner)I100[2]
· PA-Type	: (Pre-aligner)I100[2]	· PA-Size	: (Pre-aligner)I102
· Sts-Attr	: (Manipulator)I170-I189		

    *1 These parameters are system parameters .   The user cannot access these parameters.

    *2 PA-AxisN is set with PA-Type.

Yaskawa Edge Grip : 4 axis	1 axis Vacuum	: 1 axis
3 axes w/o lift : 3 axis	3 axes w/ lift	: 3 axis

58. RSTT

[Name]   Reference to status information

[Function]   Refers to the status information.
]
[Operation]   Same as above.

[Conditions]   None

[Command Format]
> $ <UNo> RSTT <Sum> <CR>

> · UNo: Unit number (1 byte)
>> · '1':   Manipulator

[Response Format]
> (Positive Message for the Completion of Execution)
>> $ <UNo> <Sts> <Ackcd> <Subcd> RSTT <RB-Rdy> <RB-Isys> <RB-MapAct> <RB-MapSign>
>>> <RB-Pos1> ---<RB-Pos5> <RB-Mov> <RB-Srv> <RB-Int> <PA-Presence>
>>> <PA-Rdy> <PA-Isys> <PA-Pos> <PA-Lift> <PA-Mov> <PA-Srv>

> · UNo: Unit number (1 byte)
>> · '1':   Unit specified by command
> · Sts: Status (2 byte)
> · Ackcd: Response code (4 bytes)
>> · Responds "0000".
> · Subcd: Sub code (4 bytes)
>> · Responds "0000".
> · RB-Rdy: Robot Ready (1 bytes)
>> · '0': Busy (It is not possible to move.),   '1': Ready (It is possible to move.)
> · RB-Isys: Robot Need 'ISYS' command (1 bytes)
>> · '0': Not need 'ISYS',   '1': Need 'ISYS'
> · RB-MapAct: Robot Mapper active. (1 bytes)
>> · '0': Laser OFF,   '1': Laser ON,
> · RB-MapSign: Robot Mappre signal (1 bytes)
>> · '0': Sensor OFF,   '1': Sensor ON,
> · RB-Pos1-5:   Robot Current Position (6 bytes)   (resolution: 0.01 [mm], [deg])
>> · RB-Pos1: Rotation axis [deg],
>> · RB-Pos2: Extension axis [mmg],
>> · RB-Pos3: End-Effector 1 [deg],
>> · RB-Pos4: End-Effector 2 [deg],
>> · RB-Pos5: Elevation axis [mm],
> · RB-Mov: Robot Status of motion (1 bytes)
>> · '0': Stop,   '1': Moving
> · RB-Srv: Robot Servo status (1 bytes)
>> · '0': Servo OFF,   '1': Servo ON
> · RB-Int: Robot Interlock (4 bytes)
>> · For details, refer to "35. SMSK."

· PA-Presence:　Pre-aligner presence (1 bytes)
　　· '0': Absence,　'1': Presence
· PA-Rdy:　　　Pre-aligner Ready (1 bytes)
　　· '0': Busy (It is not possible to move.),　'1': Ready (It is possible to move.)
· PA-Isys:　　　Pre-aligner Need 'ISYS' command (1 bytes)
　　· '0': Not need 'ISYS',　'1': Need 'ISYS'
· PA-Pos1-3:　　Pre-aligner Current Position (6 bytes)　(resolution: 0.01 [deg], [mm])
　　· PA-Pos1: Rotation axis is always "000000".
　　· PA-Pos2: X-axis [deg] (for three-axis Pre-aligner only)
　　· PA-Pos3: Y-axis [deg] (for three-axis Pre-aligner only)
· PA-Lift :　　　Pre-aligner Lifter status (1 bytes)
　　· '0': None Pre-aligner,　'1': Down,　'2': Up
· PA-Mov:　　　Pre-aligner Status of motion (1 bytes)
　　· '0': Stop,　'1': Moving
· PA-Srv:　　　Pre-aligner Servo status (1 bytes)
　　· '0': Servo OFF,　'1': Servo ON

(Negative Message for the Completion of Execution)
　　$ <UNo> <Sts> <Ackcd> <Subcd> RSTT <Sum> <CR>

· UNo:　　Unit number (1 byte)
　　· '1':　　Unit specified by command
· Sts:　　Status (2 byte)
· Ackcd:　Response code (4 bytes)
　　· Responds with the error code.
· Subcd:　Sub code (4 bytes)
　　· Detailed alarm information is shown.
　　· When either alarm does not occur, sub code is '0000'.

[Note]　　　　If Pre-aligner does not exist, <PA-Rdy>---<PA-Srv> are all '0'.

59. RPRM

[Name]   Reference to the parameters

[Function]   Reference to the parameters

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
               · "Auto" mode

[Command Format]
        $ <UNo> RPRM <Type> <Sign> <Num> <Sum> <CR>

          · UNo: Unit number (1 byte)
            · '1' or '2': Unit specified
          · Type: Parameter type (1 byte)
            · 'I':   Integer type parameter
            · 'R':   Real-number type parameter
          · Sign: (1 byte)
            · 'U':   User parameter
            · 'S':   Syatem parameter
          · Num: Parameter number (3 byte)
            Refer to Parameter Lists for Wafer Transfer Manipulator

[Response Format]
    (Positive Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> RPRM <Value> <Sum> <CR>

          · UNo: Unit number (1 byte)
            · '1' or '2': Unit specified
          · Sts: Status (2 byte)
          · Ackcd: Response code (4 bytes)
            · Responds "0000".
          · Subcd: Sub code (4 bytes)
            · Responds "0000".
          · Value: Parameter value(10 byte)
            · Type = 'I':   Specified in the range between "-000032768" and "0000032767"
            · Type = 'R':   Specified in the range between "-999999999" and "9999999999"(resolution: 0.0001)
            * If a value is less than 10 digits, add '0' to higher digit value so that it can always be specified in 10 digits.

    (Negative Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> RPRM <Sum><CR>

          · UNo: Unit number (1 byte)
            · '1' or '2': Unit specified
          · Sts: Status (2 byte)
          · Ackcd: Response code (4 bytes)
           · Responds with the error code.
          · Subcd: Sub code (4 bytes)
           · Detailed alarm information is shown.
           · When either alarm does not occur, sub code is '0000'.

60. RCCD

[Name]	Reference to the CCD Sensor Data
[Function]	Reference to the position of the wafer edge And reference to brightness on the CCD sensor side
[Operation]	Same as above.
[Conditions]	· The specified unit is stopped. · "Auto" mode

[Command Format]

$ <u>UNo</u> <u>RCCD</u> <u>Sum</u> <u>CR</u>

- · UNo: Unit number (1 byte)
  - · '2': Unit specified (Pre-aligner)

[Response Format]

(Positive Message for the Completion of Execution)

$ <u>UNo</u> <u>Sts</u> <u>Ackcd</u> <u>Subcd</u> <u>RCCD</u> <u>CCD</u> <u>Light</u> <u>Sum</u> <u>CR</u>

- · UNo: Unit number (1 byte)
  - · '2': Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds "0000".
- · Subcd: Sub code (4 bytes)
  - · Responds "0000".
- · CCD: CCD data(4 byte)
  - · The pixel is counted from the outside of the wafer, and the position of the wafer edge (position which darkens first) responds by the number of pixels.   One pixel is 0.014mm.
  - · Specified in the range between "0000" and "2047"
  - * If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.
- · Light: Light volume(4 byte)
  - · This value shows brightness on the CCD sensor side.
  - · It is shown that this value is the small degree bright.
  - · Specified in the range between "0000" and "0255"
  - * If a value is less than 4 digits, add '0' to higher digit value so that it can always be specified in 4 digits.

(Negative Message for the Completion of Execution)

$ <u>UNo</u> <u>Sts</u> <u>Ackcd</u> <u>Subcd</u> <u>RCCD</u> <u>Sum</u><u>CR</u>

- · UNo: Unit number (1 byte)
  - · '2': Unit specified
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.
  - · When either alarm does not occur, sub code is '0000'.

61. RALM

[Name]   Reference alignment mode

[Function]   Obtain currently set alignment mode information

[Operation]   Returns any of Accuracy priority / middle / Speed priority as alignment information.

[Conditions]   · "Auto" mode

[Command Format]

        $ <UNo> RALM <Mem> <Sum><CR>

    · UNo: Unit number(1 byte)
       · '2':   Unit specified(Pre-aligner)
    · Mem: Memory (1 byte)
       · 'V': Volatile memory   (D variable：returns D430, D431)
       · 'N':   Non-volatile memory (D variable：returns D450, D451)

[Response Format]

        $ <UNo> <Sts> <Ackcd> <Subcd> RALM <PMode> <ALMode> <Sum> <CR>

    · UNo: Unit number (1 byte)
       · '2':   Unit specified
    · Sts: Status (2 bytes)
    · Ackcd: Response code (4 bytes)
       · Responds "0000".
    · Subcd: Sub code (4 bytes)
       · Responds "0000".
    · PMode: Positioning mode (1 byte)
       · 'F':   Non-shortcut positioning
    · ALMode: Alignment Mode (1 byte)
       · 'A':   Accuracy mode
       · 'M':   Medium mode
       · 'S':   Throughput mode

   (Negative Message for the Completion of Execution)
        $ <UNo> <Sts> <Ackcd> <Subcd> RALM <Sum><CR>

    · UNo: Unit number (1 byte)
       · '2':   Unit specified
    · Sts: Status (2 bytes)
    · Ackcd: Response code (4 bytes)
       · Responds with the error code.
    · Subcd: Sub code (4 bytes)
       · Detailed alarm information is shown.
       · When either alarm does not occur, sub code is '0000'.

62.. RSTD

[Name]   Reference standard position

[Function]   Obtain Standard Position value which is currently set.

[Operation]   Returns Standard position value of Manipulator and Pre-aligner.

[Conditions]   · "Auto" mode

[Command Format]

    $ <UNo> RSTD <Sum><CR>

    · UNo: Unit number(1 byte)
      · '2':   Unit specified(Pre-aligner)

[Response Format]

    $ <UNo> <Sts> <Ackcd> <Subcd> RSTD<Value1>…<ValueN> <Sum> <CR>

    · UNo: Unit number (1 byte)
      · '2':   Unit specified
    · Sts: Status (2 bytes)
    · Ackcd: Response code (4 bytes)
      · Responds "0000".
    · Subcd: Sub code (4 bytes)
      · Responds "0000".
    · Value: Position data(11 bytes each)
      · In case the value is less than 11 figures, set 0 in the head.
      <Manipulator>
          Value1: S axis Value2: R axis   Value3: End-Effector1 axis
          Value4: End-Effector 2 axis   Value5: Z axis
      <Pre-aligner>
          Value1: S axis Value2: X axis   Valu3: Y axis

  (Negative Message for the Completion of Execution)
    $ <UNo> <Sts> <Ackcd> <Subcd> RSTD <Sum><CR>

    · UNo: Unit number (1 byte)
      · '2':   Unit specified
    · Sts: Status (2 bytes)
    · Ackcd: Response code (4 bytes)
      · Responds with the error code.
    · Subcd: Sub code (4 bytes)
      · Detailed alarm information is shown.
      · When either alarm does not occur, sub code is '0000'.

63. RWSZ

[Name]   Reference wafer size

[Function]   Obtain the wafer size information which is currently set..

[Operation]   Returns any of the wafer size 150mm, 200mm, and 300mm.

[Conditions]   · The specified unit is stopped.
· "Auto" mode

[Command Format]

$ <UNo> RWSZ <End-Effector> <Sum><CR>

· UNo: Unit number(1 byte)
· '1' or '2'   : Unit specified
· End-Effector: End-Effector specified (1 byte)
· 'A':   When End-Effector 1 specified/Pre-aligner
· 'B':   When End-Effector 2 specified

[Response Format]

$ <UNo> <Sts> <Ackcd> <Subcd> RWSZ <Size> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' or '2'   : Unit specified
· Sts: Status (2 bytes)
· Ackcd: Response code (4 bytes)
· Responds "0000".
· Subcd: Sub code (4 bytes)
· Responds "0000".
· Size: Diameter of the wafer (3 bytes)
· '300' : 300mm
· '200' : 200mm
· '150' : 150mm

(Negative Message for the Completion of Execution)
$ <UNo> <Sts> <Ackcd> <Subcd> RWSZ <Sum><CR>

· UNo: Unit number (1 byte)
· '1' or '2'   : Unit specified
· Sts: Status (2 bytes)
· Ackcd: Response code (4 bytes)
· Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

64. RIOS

[Name]   Reference I/O status

[Function]   Obtain the Input/Output ports status information which is currently set..

[Operation]   Returns binary string of I/O status(OFF:0,ON:1).

[Conditions]   · "Auto" mode

[Command Format]

$ <UNo> RIOS <I/O Selection> <Sum><CR>

· UNo: Unit number(1 byte)
   · '1':   Unit specified
· I/O Selection: Input/Output port selected (1 byte)
   · '1':   When Input port specified
   · '2':   When Output port specified

[Response Format]

$ <UNo> <Sts> <Ackcd> <Subcd> RIOS <I/O Status> <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' :   Unit specified
· Sts: Status (2 bytes)
· Ackcd: Response code (4 bytes)
   · Responds "0000".
· Subcd: Sub code (4 bytes)
   · Responds "0000".
· I/O Status: Binary string of I/O status (28 bytes) OFF:0,ON:1
   <Input port>

Di00	Di01	Di02	Di03	Di04	Di05	Di06	Di07	Di08	Di09	Di10	Di11	Di20	Di21

Di22	Di23	Di24	Di25	Di26	Di27	Di28	Di29	Di30	Di31	Di32	Di33	Di34	Di35

   <Output port>

Do00	Do01	Do02	Do03	Do04	Do07	Do08	Do09	Do10	Do11	Do12	Do13	Do20	Do21

Do22	Do23	Do24	Do25	Do26	Do27	Do28	Do29	,Do30	Do31	Do32	Do33	Do34	Do35

(Negative Message for the Completion of Execution)
$ <UNo> <Sts> <Ackcd> <Subcd> RIOS <Sum><CR>

· UNo: Unit number (1 byte)
   · '1' or '2'   : Unit specified
· Sts: Status (2 bytes)
· Ackcd: Response code (4 bytes)
   · Responds with the error code.
· Subcd: Sub code (4 bytes)
   · Detailed alarm information is shown.
   · When either alarm does not occur, sub code is '0000'.

65. ACKN

[Name]   Acknowledgement of Execution Completion

[Function]   Informs the Controller of acknowledgement for the completion of execution.
Informs the Controller that the message for the completion of execution has been received from the Controller.
The Controller does not respond concerning this command.

[Operation]   Same as above.

[Conditions] None

[Command Format]
$ <UNo> ACKN <Sum> <CR>

· UNo: Unit number (1 byte)
   · '1' or '2':   Unit specified

[Response Format]
None

66. UPOS

[Name]   Registered Position Data Uploading

[Function]   Requests for all the registered position data in the non-volatile memory for the specified unit.

Note: This command is used for maintenance.   The data saved (backed up) by this command can be downloaded
to the Controller by using "DPOS" that is described later.

Also, the registered position data cannot be changed because it is of the Controller's internal data type.

[Operation]   When the Controller receives this command, it returns up to 10 sets of the registered position data (for all axes).
The Host monitors the block number of the message received from the Controller and repeats sending this
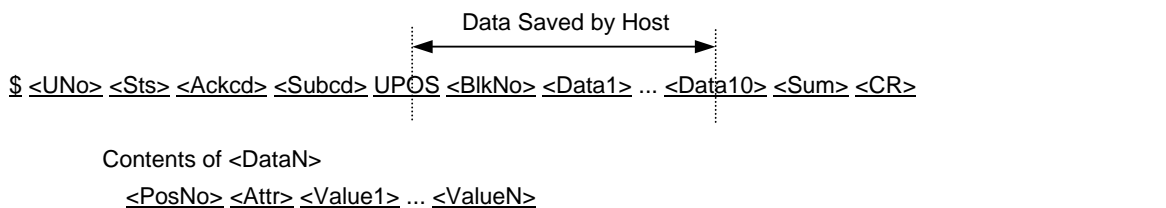command until it identifies "-1."

[Conditions]   · The specified unit is stopped.

· "Auto" mode

[Command Format]

$ <UNo> UPOS <Sum> <CR>

· UNo: Unit number (1 byte)

· '1' to '2':   Unit specified

[Response Format]

(Positove Message for the Completion of Execution)

Data Saved by Host

$ <UNo> <Sts> <Ackcd> <Subcd> UPOS <BlkNo> <Data1> ... <Data10> <Sum> <CR>

Contents of <DataN>

<PosNo> <Attr> <Value1> ... <ValueN>

· UNo: Unit number (1 byte)

· '1' to '2':   Unit specified

· Sts: Status (2 byte)

· Ackcd: Response code (4 bytes)

· Responds "0000".

· Subcd: Sub code (4 bytes)

· Responds "0000".

· BlkNo: Block number (2 bytes)

· Responds in the range between"00" and "99."

· If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.

· Responds "-1" as the final data.

· PosNo: Identification number (3 bytes)
  · Responds in the range between "000" and "999."
  · If a value is less than 3 digits, add '0' to higher digit value so that it can always be specified in 3 digits.
  · Responds "-99" as the final data.
· Attr: Attribute of position data (8 bytes)
  · Shows interpolation method etc.
· ValueN: Registered position data (8 bytes each)
  · Be sure to respond to as many as the number of the axes for the specified unit.
  · The data of identification number "-99" is a checksum value.

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> UPOS <Sum> <CR>

· UNo: Unit number (1 byte)
  · '1' to '2': Unit specified by command
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
  · Responds with the error code.
· Subcd: Sub code (4 bytes)
  · Detailed alarm information is shown.

[Note] · Up to 10 sets of the registered position data are returned at one transmission from the Controller.
· If any error occurs during execution of this command, or if any other unit is specified or any other command is sent, the Controller interrupts uploading.
· The registered position data to be sent to the Host by this command is of the Controller's internal data type. Therefore, the registered position data received by the Host cannot be changed.
· Because the registered position data in the non-volatile memory is transferred to the volatile memory when executing this command, the registered position data in the volatile memory is cleared.

67. UPRM

[Name]   Internal Parameter Uploading

[Function]   Requests for all the Controller's internal parameters for the specified unit.

              Note: This command is used for maintenance.   The data saved (backed up) by this command can be downloaded
                    to the Controller by using "DPRM" that is described later.

                    Also, the internal parameters cannot be changed because it is of the Controller's internal data type.

[Operation]   When the Controller receives this command, it returns up to 10 sets of parameters.   The Host monitors the block
              number of the message received from the Controller and repeats sending this command until it identifies "-1."

[Conditions]   · The specified unit is stopped.
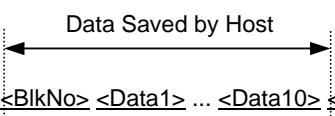   · "Auto" mode

[Command Format]

              $ <UNo> UPRM <Sum> <CR>

            · UNo: Unit number (1 byte)
               · '1' to '2':   Unit specified

[Response Format]

        (Positive Message for the Completion of Execution)

<div align="center">Data Saved by Host</div>

        $ <UNo> <Sts> <Ackcd> <Subcd> UPRM <BlkNo> <Data1> ... <Data10> <Sum> <CR>

                Contents of <DataN>
                    <Type> <Sign> <PrmNo> <Value>

      · UNo: Unit number (1 byte)
         · '1' to '2':   Unit specified
      · Sts: Status (2 byte)
      · Ackcd: Response code (4 bytes)
         · Responds "0000".
      · Subcd: Sub code (4 bytes)
         · Responds "0000".
      · BlkNo: Block number (2 bytes)
         · Responds in the range between"00" and "99."
         · If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.
         · Responds "-1" as the final data.
      · Type: Parameter identification (1 byte)
         · 'I':   Integral-type parameter
         · 'R':   Real-number-type parameter
      · Sign: (1 byte)
         · 'U':   User parameter only   (System parameter is protected)

· PrmNo: Identification number (3 bytes)
  · Responds in the range between "000" and "999."
  · If a value is less than 3 digits, add '0' to higher digit value so that it can always be specified in 3 digits.
  · Responds "-99" as the final data.
· Value: Parameter
  · In case of integtal type (8 bytes)   :   Data between "00000000" and "FFFFFFFF"
  · In case of real-number type (8 bytes):   Data between "00000000" and "FFFFFFFF"
  · Identification number "-99" data is a checksum value.:   Data between "00000000" and "FFFFFFFF"

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> UPRM <Sum> <CR>

· UNo: Unit number (1 byte)
  · '1' to '2':   Unit specified by command
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
  · Responds with the error code.
· Subcd: Sub code (4 bytes)
  · Detailed alarm information is shown.

[Note]   · Up to 10 sets of parameters are returned at one transmission from the Controller.
         · If any error occurs during execution of this command, or if any other unit is specified or any other command is sent, the Controller interrupts uploading.
         · The parameter to be sent to the Host by this command is of the Controller's internal data type.   Therefore, the parameters received by the Host cannot be changed.

68. DPOS

[Name] Registered Position Data Downloading

[Function] Downloads the registered position data that has been uploaded by "UPOS" onto the Controller's non-volatile memory for the specified unit. The Host repeats sending this command until the block number becomes "-1."

Note: After completion of downloading, the Controller's registered position data returns to the status at uploading.

[Operation] Same as above.

[Conditions] · The specified unit is stopped.
      · "Auto" mode

[Command Format]

    $ <UNo> DPOS <BlkNo> <Data1> ... <Data10> <Sum> <CR>

     Contents of <DataN>
      <PosNo> <Attr> <Value1> ... <ValueN>

Note: The response data parts of "UPOS" are used for <BlkNo> and <DataN> without changing.

 · UNo: Unit number (1 byte)
  · '1' to '2': Unit specified
 · BlkNo: Block number (2 bytes)
  · Responds in the range between"00" and "99."
  · If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.
  · Responds "-1" as the final data.
 · PosNo: Identification number (3 bytes)
  · Responds in the range between "000" and "999."
  · If a value is less than 3 digits, add '0' to higher digit value so that it can always be specified in 3 digits.
  · Responds "-99" as the final data.
 · Attr: Attribute of position data (8 bytes)
  · Shows interpolation method etc.
 · ValueN: Registered position data (8 bytes each)
  · Be sure to specify as many as the number of the axes for the specified unit.
  · The data of identification number "-99" is a checksum value.

[Response Format]

 (Positive Message for the Completion of Execution)
   $ <UNo> <Sts> <Ackcd> <Subcd> DPOS <Sum> <CR>

  · UNo: Unit number (1 byte)
   · '1' to '2': Unit specified
  · Sts: Status (2 byte)
  · Ackcd: Response code (4 bytes)
   · Responds "0000".
  · Subcd: Sub code (4 bytes)
   · Responds "0000".

(Negative Message for the Completion of Execution)

       $ <u>UNo</u> <u>Sts</u> <u>Ackcd</u> <u>Subcd</u> <u>DPOS</u> <u>Sum</u> <u>CR</u>

- · UNo: Unit number (1 byte)
  - · '1' to '2':   Unit specified by command
- · Sts: Status (2 byte)
- · Ackcd: Response code (4 bytes)
  - · Responds with the error code.
- · Subcd: Sub code (4 bytes)
  - · Detailed alarm information is shown.

[Note]   · If any error occurs during execution of this command, or if any other unit is specified or any other command is sent,

the Controller interrupts uploading.

· The data to be downloaded by this command is only the data that has been uploaded by "UPOS." If any incorrect data is sent, the Controller operations are not guaranteed.

· By turning ON the power supply of the Controller again after completion of this command, the downloaded data becomes effective.

69. DPRM

[Name]   Internal Parameter Downloading

[Function]   Downloads internal parameter that has been uploaded by "UPRM" onto the Controller's non-volatile memory for the specified unit.   The Host repeats sending this command until the block number becomes "-1."

Note: After completion of downloading, the Controller's internal parameter returns to the status at uploading.

[Operation]   Same as above.

[Conditions]   · The specified unit is stopped.
            · The specified unit is under servo OFF state.
            · "Auto" mode

[Command Format]

        $ <u>\<UNo\></u> <u>DPRM</u> <u>\<BlkNo\></u> <u>\<Data1\></u> ... <u>\<Data10\></u> <u>\<Sum\></u> <u>\<CR\></u>

            Contents of \<DataN\>
              <u>\<Type\></u> <u>\<Sign\></u> <u>\<PrmNo\></u> <u>\<Value\></u>

Note: The response data parts of "UPRM" are used for \<BlkNo\> and \<DataN\> without changing.

    · UNo: Unit number (1 byte)
        · '1' to '2':   Unit specified
    · BlkNo: Block number (2 bytes)
        · Responds in the range between"00" and "99."
        · If a value is less than 2 digits, add '0' to higher digit value so that it can always be specified in 2 digits.
        · Responds "-1" as the final data.
    · Type: Parameter identification (1 byte)
        · 'I':   Integral-type parameter
        · 'R':   Real-number-type parameter
    · Sign: (1 byte)
        · 'U':   User parameter only   (System parameter is protected)
    · PrmNo: Identification number (3 bytes)
        · Responds in the range between "000" and "999."
        · If a value is less than 3 digits, add '0' to higher digit value so that it can always be specified in 3 digits.
        · Responds "-99" as the final data.
    · Value: Parameter
        · In case of integtal type (8 bytes):   Data between "00000000" and "FFFFFFFF"
        · In case of real-number type (8 bytes):   Data between "00000000" and "FFFFFFFF"
        · Identification number "-99" data is a checksum value.:   Data between "00000000" and "FFFFFFFF"

[Response Format]

(Positive Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> DPRM <Sum> <CR>

· UNo: Unit number (1 byte)

· '1' to '2':   Unit specified

· Sts: Status (2 byte)

· Ackcd: Response code (4 bytes)

· Responds "0000".

· Subcd: Sub code (4 bytes)

· Responds "0000".

(Negative Message for the Completion of Execution)

$ <UNo> <Sts> <Ackcd> <Subcd> DPRM <Sum> <CR>

· UNo: Unit number (1 byte)

· '1' to '2':   Unit specified by command

· Sts: Status (2 byte)

· Ackcd: Response code (4 bytes)

· Responds with the error code.

· Subcd: Sub code (4 bytes)

· Detailed alarm information is shown.

[Note]

· If any error occurs during execution of this command, or if any other unit is specified or any other command is sent, the Controller interrupts uploading.

· The data to be downloaded by this command is only the data that has been uploaded by "UPRM."   If any incorrect data is sent, the Controller operations are not guaranteed.

· By turning ON the power supply of the Controller again after completion of this command, the downloaded data becomes effective.

70. HRST

[Name]	Software Reset Command

[Function]   Reset (Both the robot and Pre-aligner even if Uno is 1) and restart controller.
Modified data in volatile memory is lost in executing this command, since position data and parameter in Non-volatile memory are loaded to volatile memory.
Allows all modified parameters to take affect.
Servo motors are powered off once
I/O states are reset to their default value once

[Operation]   Same as above.

[Conditions]   · The robot and Prealignment are stopped.
· "Auto" mode

[Command Format]
$ <UNo> HRST <Sum> <CR>

· UNo: Unit number (1 byte)
· '1':   Manipulator

[Response Format]
(Response Message)
@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' :   Unit specified
· Sts: Status (2 byte)
· Ackcd: Response code (4 bytes)
· At normal reception: Responds "0000".
· Reception rejected: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

(Message for the Completion of Execution)
$ <UNo> <Sts> <Errcd> <Subcd> HRST <Sum> <CR>

· UNo: Unit number (1 byte)
· '1' :   Unit specified
· Sts: Status (2 byte)
· Errcd: Error Code (4 bytes)
· At normal completion: Responds "0000".
· At erroneous completion: Responds with the error code.
· Subcd: Sub code (4 bytes)
· Detailed alarm information is shown.
· When either alarm does not occur, sub code is '0000'.

71. MACR

[Name]	User macro communication command
[Function]	This command is used to communicate with user macro.
[Operation]	Same as above.
[Conditions]	User definition

[Command Format]

$ <UNo> MACR <ComData> <Sum> <CR>

- ・ UNo: Unit number (1 byte)
  - ・ '1':   Manipulator
  - ・ '2':   Pre-aligner
- ・ ComData: Communication data (1 ~ 55 bytes)
  - ・ User definition

[Response Format]

User can define other than below restrictions.

Response format: <STX> <Data> <Sum> <CR>

Restrictions:

- <STX>, head of message, needs to be '$', '@' or '!'
- The end of message needs to be <CR>
- <Data> must not include '$', '@', '!' or <CR>
- Total message length needs to be less than 128 bytes.

Note) <Sum> and <CR> will be added automatically by firmware.

<Sum> can be omitted by parameter setting.

[Example of Ack response]

@ <UNo> <Sts> <Ackcd> <Subcd> <Sum> <CR>

- ・ <UNo>   : Unit number   1 Byte
  - '1': Manipulator
  - '2': Pre-aligner
- ・ <Sts>   : Status   2 Bytes
- ・ <Ackcd>: Response code   4 Bytes
- ・ <Subcd>: Sub code 4 Bytes

[Example of completion response]

$ <UNo> <Sts> <Errcd> <Subcd> MACR <ComData> <Sum> <CR>

- ・ <Uno>: Unit number
  - '1': Manipulator
  - '2': Pre-aligner
- ・ <Sts>: Status bit (2Bytes)
- ・ <Errcd>: Error code (4 Bytes)
- ・ <Subcd>: Sub code (4 Bytes)
- ・ <ComData>: Response message
  - User definition   0 ~ 109 Bytes

THIS PAGE INTENTIONALLY LEFT BLANK.