YASKAWA

DX100/DX200/FS100 OPTIONS INSTRUCTIONS

REFERENCE MANUAL
FOR PROGRAMMING PENDANT CUSTOMIZATION FUNCTION
(MotoConnectCSU.dll API SPECIFICATIONS
FOR DEVELOPING Visual Studio. NET C#)

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

MOTOMAN INSTRUCTIONS

MOTOMAN-☐☐☐INSTRUCTIONS
DX100/DX200 INSTRUCTIONS
DX100/DX200 OPERATOR'S MANUAL
DX100/DX200 MAINTENANCE MANUAL

FS100 INSTRUCTIONS
FS100 OPERATOR'S MANUAL
FS100 MAINTENANCE MANUAL

The DX100/DX200/FS100 operator's manual above corresponds to specific usage.
Be sure to use the appropriate manual.

Part Number: 166583-1CD
Revision: 1

YASKAWA
MOTOMAN ROBOTICS

MANUAL NO.
HW1481485①

Table of Contents

1 Outline

In this manual, details of APIs used when the programming pendant customization application is developed with Visual Studio 2005 C# are explained.

2 LIBRARY OPNE-CLOSE API

MotoConnect

Constructor. Execute initialization of the library.

■ **Syntax: Default IP address**
(When Controller 10.0.0.2/ Pendant 10.0.0.4)
MotoConnect (bool Endian)

■ **Parameter**
[in]*Endian*

Endian type of the pendant CPU

Value	Description
true	Big endian
false	Little endian
Set “false” when the controller is either the DX100, the DX200 or the FS100.	

■ **Return Value**
No value

■ **Syntax: Other than default IP address**
MotoConnect(bool Endian, string Ycplp, string Ypplp)

■ **Parameter**
[in]*Endian*

Endian type of the pendant CPU

Value	Description
true	Big endian
false	Little endian
Set “false” when the controller is either the DX100, the DX200 or the FS100.	

[in]*Ycplp*

Controller IP address

[in]*Ypplp*

pendant IP address

■ **Return Value**
No value

3 SYSTEM MONITOR API

YppGetVarData

Acquires the value of variable (B, I, D, R).

■ **Syntax**
int YppGetVarData(ushort DataType, ushort DataIndex, out int rData)

■ **Parameter**
[in] *DataType*

Variable type

Value	Description
1	B variable
2	I variable
3	D variable
4	R variable

[in] *DataIndex*

Variable number

[out] *rData*

Variable value (result of acquisition)

■ **Return value**
0 : Normal end

Other than 0 : Error

YppReadIO

Reads I/O.

■ **Syntax**
int YppReadIO(uint Address, out ushort Value)

■ **Parameter**
[in] *Address*

Address of I/O to be read.

<DX100>

Value	Description
10 - 2567	Universal input #00010 - #02567 (2048)
10010 - 12567	Universal output #10010 - #12567 (2048)
20010 - 22567	External input #20010 - #22567 (2048)
30010 - 32567	External output #30010 - #32567 (2048)
40010 - 41607	Specific input #40010 - #41607 (1280)
50010 - 52007	Specific output #50010 - #52007 (1600)
60010 - 60647	I/F panel #60010 - #60647 (512)
70010 - 79997	Auxiliary relay #70010 - #79997 (7992)
80010 - 80647	Control input #80010 - #80647 (512)
82010 - 82207	Pseud input #82010 - #82207 (160)
25010 - 27567	Network input #25010 - #27567 (2048)
35010 - 37567	Network output #35010 - #37567 (2048)
1000000 - 1000559	Register #1000000 - #1000559 (560)

<DX200>

Value	Description
10 - 5127	Universal input #00010 - #05127 (4096)
10010 - 15127	Universal output #10010 - #15127 (4096)
20010 - 25127	External input #20010 - #25127 (4096)
30010 - 35127	External output #30010 - #35127 (4096)
40010 - 41607	Specific input #40010 - #41607 (1280)
50010 - 53007	Specific output #50010 - #53007 (2400)
60010 - 60647	I/F panel #60010 - #60647 (512)
70010 - 79997	Auxiliary relay #70010 - #79997 (7992)
80010 - 80647	Control input #80010 - #80647 (512)
82010 - 82207	Pseud input #82010 - #82207 (160)
27010 - 29567	Network input #27010 - #29567 (2048)
37010 - 39567	Network output #37010 - #39567 (2048)
1000000 - 1000559	Register #1000000 - #1000559 (560)

<FS100>

Value	Description
10 - 1278	Universal input #00010 - #01287 (1024)
10010 - 11278	Universal output #10010 - #11287 (1024)
20010 - 21278	External input #20010 - #21287 (1024)
30010 - 31278	External output #30010 - #31287 (1024)
40010 - 41607	Specific input #40010 - #41607 (1280)
50010 - 52007	Specific output #50010 - #52007 (1600)
60010 - 60647	I/F panel #60010 - #60647 (512)
70010 - 79997	Auxiliary relay #70010 - #79997 (7992)
80010 - 80647	Control input #80010 - #80647 (512)
82010 - 82207	Pseud input #82010 - #82207 (160)
25010 - 26287	Network input #25010 - #26287 (1024)
35010 - 36287	Network output #35010 - #36287 (1024)
1000000 - 1000559	Register #1000000 - #1000559 (560)

[out]Value

I/O value (result of acquisition)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetPosVarData

Acquires the position-type variable.

■ **Syntax**

int YppGetPosVarData
(ushort DataType, ushort DataIndex, out int[] rData)

■ **Parameter**

[in]*DataType*

Variable type

Value	Description
5	Robot
6	Base
7	Station

[in]*DataIndex*

Variable number

[out]*rData (int[10])*

Position data (result of acquisition) Note:Area for int[10] is required.

Array	Description	
	Bit No.	Content
rData[0]	D05 - D00	Variable type 0 Pulse 16 Cartesian (base coordinates) 17 Cartesian (robot coordinates) 18 Cartesian (tool coordinates) 19 Cartesian (user coordinates) 20 Cartesian (reserved for master tool)
	D07 - D06	Reserved by manufacturer
	D08	0:Front 1:Back
	D09	0:Upper arm 1: Lower arm
	D10	0:Flip 1 :No flip
	D11	0:R<180deg 1:R>=180deg
	D12	0:T<180deg 1:T>=180deg
	D13	0:S<180de 1:S>=180deg
	D14 - D15	Reserved by manufacturer
	D16 - D21	Tool number (0 - 23)
	D22 - D27	User coordinate number
	D28 - D31	Reserved by manufacturer
rData[1]		(Extended attribute)

Array	Description	
	Bit No.	Content
Array	Pulse	Cartesian
rData[2]	1st axis (S) pulse value	X-axis coordinate (unit: micron)
rData[3]	2nd axis (L) pulse value	Y-axis coordinate (unit: micron)
rData[4]	3rd axis (U) pulse value	Z-axis coordinate (unit: micron)
rData[5]	4th axis (R) pulse value	Wrist angle Rx (unit: 0.0001deg)
rData[6]	5th axis (B) pulse value	Wrist angle Ry (unit: 0.0001deg)
rData[7]	6th axis (T) pulse value	Wrist angle Rz (unit: 0.0001deg)
rData[8]	7th axis (E) pulse value	angle Re (unit: 0.0001deg)
rData[9]	8th axis pulse value	8th axis pulse value (micron in the case of traveling axis)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetAlarmStatus

Acquires the error and alarm status.

■ **Syntax**
int YppGetAlarmStatus(out short Status)

■ **Parameter**
[out]*Status*

Error and alarm status. (result of acquisition)

Value	Description
D00	1: In error status
D01	1: In alarm status
D02 - D15	(Unused)

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetAlarmCode

Acquires the error and alarm code.

■ **Syntax**
int YppGetAlarmCode(out ushort ErrorNo, out ushort ErrorData, out ushort AlarmNum, out ushort[] AlarmNo, out ushort[] AlarmData)

■ **Syntax**
[out] *ErrorNo*

Error number (result of acquisition)

[out*] ErrorData*

Error data (result of acquisition)

[out] *AlarmNum*

The numbers of alarm (result of acquisition) Note: 4 at maximum

[out] *AlarmNo (ushort[4])*

Alarm number (result of acquisition)

Note: The area for ushort[4] is required.

[out] *AlarmData (ushort[4])*

Alarm data (ushort[4]) (result of acquisition)

Note: The area for ushort[4] is required.

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetMode

Acquires the operation mode of the system.

■ **Syntax**
int YppGetMode(out short Mode, out short Remote)

■ **Parameter**
[out] *Mode*

Operation mode (result of acquisition)

Value	Description
1	Teach mode
2	Play mode

[out] *RemoteMode*

Command remote mode (result of acquisition)

Value	Description
0	Command remote OFF
1	Command remote ON

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetCycle

Acquires the cycle mode of the system.

■ **Syntax**
int YppGetCycle(out short Cycle)

■ **Parameter**
[out] *Cycle*

Cycle (result of acquisition)

Value	Description
0	Step
1	One Cycle
3	Auto

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetServoPower

Acquires the ON/OFF status of the servo power.

■ **Syntax**
int YppGetServoPower(out short Power)

■ **Parameter**
[out] *Power*

Cycle (result of acquisition)

Value	Description
0	Servo Power OFF
1	Servo Power ON

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetPlayStatus

Acquires the operation status of the job.

■ **Syntax**
int YppGetPlayStatus(out short Start, out short Hold)

■ **Parameter**
[out] *Start*

Operation status (result of acquisition)

Value	Description
0	Start OFF
1	Start ON

[out] *Hold*

Hold status (result of acquisition)

Value	Description
0	Hold OFF
1	Hold ON

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetMasterJob

Retrieves the master job name of the specified local task.

■ Syntax

int YppGetMasterJob(ushort TaskNo, out string JobName)

■ Parameter

[in] *TaskNo*

Task number

Value	Description
0	Master Task
1	SubTask 1
2	SubTask 2
3	SubTask 3
4	SubTask 4
5	SubTask 5
6	SubTask 6
7	SubTask 7
8	SubTask 8
9	SubTask 9
10	SubTask 10
11	SubTask 11
12	SubTask 12
13	SubTask 13
14	SubTask 14
15	SubTask 15

[out] *JobName*

Master job name (up to 32 characters for a job name)

■ Return value

0 : Normal end

Other than 0 : Error

YppGetCurrentJob

Acquires the name, line, step number of the current job.

■ **Syntax**

int YppGetCurrentJob(ushort TaskNo, out ushort LineNo, out ushort StepNo, out string JobName)

■ **Parameter**

[in] *TaskNo*

Task number

Value	Description
0	Master Task
1	SubTask 1
2	SubTask 2
3	SubTask 3
4	SubTask 4
5	SubTask 5
6	SubTask 6
7	SubTask 7
8	SubTask 8
9	SubTask 9
10	SubTask 10
11	SubTask 11
12	SubTask 12
13	SubTask 13
14	SubTask 14
15	SubTask 15

Note: In the FS100 system, Sub task 6 and later tasks are invalid.

[out] *LineNo*

Job line number

[out] *StepNo*

Step number

[out] *JobName*

Job name (up to 32 characters available)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetSpecialOpStatus

Retrieves the status of special operation.

■ Syntax

int YppGetSpecialOpStatus(out short SpecialOpStatus)

■ Parameter

[out] *SpecialOpStatus*

Special operation status (Result)

Value	Description
D0	Check operation
D1	Safety speed operation
D2	Dry-run speed operation
D3	Machine lock operation
D4	(Reserved)
D5	Low speed operation
D6	Weaving prohibited
D7-D9	(Reserved)
D10	Pressuring instruction prohibited
D11-D15	(Reserved)

■ Return value

0 : Normal end

Other than 0 : Error

■ Restrictions

This API is valid only in Play mode.

YppGetJobDate

Acquires the job date.

■ **Syntax**
int YppGetJobData(string JobName, out short[] SysTimeData)

■ **Parameter**
[in] *JobName*

Job name (up to 32 characters available)

[out] *SysTimeData (ushort[6])*

Time and date data (result of acquisition)

Note: Area for short[6] is required.

SysTimeData[0]	:	Year
SysTimeData[1]	:	Month
SysTimeData[2]	:	Date
SysTimeData[3]	:	Time
SysTimeData[4]	:	Minute
SysTimeData[5]	:	Second

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetCartPosEx

Acquires the current position in Cartesian Coordinates with a specified coordinate frame (Robot, Base, User).

■ Syntax

int YppGetCartPosEx(short RobotNo, short Frame, short ToolNo, out int[] Pos, out short Config)

■ Parameter

[in] *RobotNo*

Robot number

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)

Note: In the FS100 system, R5 (Robot 5) and later values are invalid.

[in] *Frame*

Specified coordinate frame

Value	Description
0	Base coordinate
1	Robot coordinate
2, 3, ...	User coordinate1, 2, ...

[in] *ToolNo*

Tool number 0 to 63

Note: In the FS100 system the tool number is from 1 to 15.

[out] *Pos (int[12])*

Position data (result of acquisition)

Note: Area for int[12] is required.

Value	Description
lPos[0]	X-axis coordinate (unit: micron)
lPos[1]	Y-axis coordinate (unit: micron)
lPos[2]	Z-axis coordinate (unit: micron)
lPos[3]	Wrist angle Rx (unit: 0.0001deg)
lPos[4]	Wrist angle Ry (unit: 0.0001deg)
lPos[5]	Wrist angle Rz (unit: 0.0001deg)
lPos[6]	angle Re (unit: 0.0001deg)
lPos[7]	(Reserved)
lPos[8]	1st external axis pulse value (micron in the case of traveling axis)
lPos[9]	2nd external axis pulse value (micron in the case of traveling axis)
lPos[10]	3rd external axis pulse value (micron in the case of traveling axis)
lPos[11]	(Unused)

[out] *Config*

Figure information (result of acquisition)

Value	Description
D00	0:Front 1:Back
D01	0:Upper arm 1:Lower arm
D02	0:Flip 1:No flip
D03	0:R<180deg 1:R>=180deg
D04	0:T<180deg 1:T>=180deg
D05	0:S<180deg 1:S>=180deg
D06	0:L<0deg 1:L>=0deg
D07 -D15	Reserved

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetPulsePos

Acquires the current position in pulse.

Syntax

int YppGetPulsePos(uint CtrlGrp, out int[] Pos)

Parameter

[in] *CtrlGrp*

Control group

<DX100/DX200>

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

<FS100>

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	(Reserve)
5	(Reserve)
6	(Reserve)
7	(Reserve)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	(Reserve)
13	(Reserve)
14	(Reserve)
15	(Reserve)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)

[[out] *Pos (int[8])*

Pulse coordinate position data

Note: Area for int[8] is required;

Value	Description
lPos[0]	1st axis (S) pulse value
lPos[1]	2nd axis (L) pulse value
lPos[2]	3rd axis (U) pulse value
lPos[3]	4th axis (R) pulse value
lPos[4]	5th axis (B) pulse value
lPos[5]	6th axis (T) pulse value
lPos[6]	7th axis (E) pulse value
lPos[7]	8th axis pulse value

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetFBPulsePos

Retrieves the feedback position in pulse count.

■ **Syntax**
int YppGetFBPulsePos(uint CtrlGrp, out int[] Pos)

■ **Parameter**
[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[[out] *Pos (int[8])*

Position coordinates in pulse (Result)

Note: Area for int[8] is required;

Value	Description
Pos[0]	1st axis (S) pulse value
Pos[1]	2nd axis (L) pulse value
Pos[2]	3rd axis (U) pulse value
Pos[3]	4th axis (R) pulse value
Pos[4]	5th axis (B) pulse value
Pos[5]	6th axis (T) pulse value
Pos[6]	7th axis (E) pulse value
Pos[7]	8th axis pulse value

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetServoSpeed

Retrieves the current servo speed by seconds.

■ **Syntax**
int YppGetServoSpeed(uint CtrlGrp, out int[] Speed)

■ **Parameter**
[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[[out] *Speed (int[8])*

Servo speed data (Result)

Note: Area for int[8] is required;

Value	Description
Speed[0]	1st axis (S) direction (unit: pulses per second - pps)
Speed[1]	2nd axis (L) direction (unit: pps)
Speed[2]	3rd axis (U) direction (unit: pps)
Speed[3]	4th axis (R) direction (unit: pps)
Speed[4]	5th axis (B) direction (unit: pps)
Speed[5]	6th axis (T) direction (unit: pps)
Speed[6]	7th axis (E) direction (unit: pps)
Speed[7]	8th axis direction (unit: pps)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetFBSpeed

Retrieves the feedback speed by seconds.

Syntax

int YppGetFBSpeed(uint CtrlGrp, out int[] Speed)

Parameter

[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[[out] *Speed (int[8])*

Feedback speed data (Result)

Note: Area for int[8] is required;

Value	Description
Speed[0]	1st axis (S) direction (unit: pulses per second - pps)
Speed[1]	2nd axis (L) direction (unit: pps)
Speed[2]	3rd axis (U) direction (unit: pps)
Speed[3]	4th axis (R) direction (unit: pps)
Speed[4]	5th axis (B) direction (unit: pps)
Speed[5]	6th axis (T) direction (unit: pps)
Speed[6]	7th axis (E) direction (unit: pps)
Speed[7]	8th axis direction (unit: pps)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetTorque

Retrieves the percentage to the maximum current servo torque value.

■ **Syntax**
int YppGetTorque(uint CtrlGrp, out int[] TorquePercent)

■ **Parameter**
[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[[out] *TorquePercent (int[8])*

Torque percentage (Result)

Note: Area for int[8] is required;

Value	Description
TorquePercent[0]	1st axis (S) direction (unit: 0.01% of max torque)
TorquePercent[1]	2nd axis (L) direction (unit: 0.01%)
TorquePercent[2]	3rd axis (U) direction (unit: 0.01%)
TorquePercent[3]	4th axis (R) direction (unit: 0.01%)
TorquePercent[4]	5th axis (B) direction (unit: 0.01%)
TorquePercent[5]	6th axis (T) direction (unit: 0.01%)
TorquePercent[6]	7th axis (E) direction (unit: 0.01%)
TorquePercent[7]	8th axis direction (unit: 0.01%)

■ **Return value**

0 : Normal end

Other than 0 : Error

YppGetSystemTimes

Acquires the current system time.

■ **Syntax**

int YppGetSystemTime(short TimeType, out short[] SysTimeData, out int ElapsedTime)

■ **Parameter**

[in] *TimeType*

Type of the system time

Value	Description
0	Time during the power supply of the controller is ON
1	Time during the servo power supply is ON
2	Time during the play back operation
3	Moving time
4	Operation time

[out] *SysTimeData (short[6])*

Date and time data (result of acquisition)

Note: Area for short[6] is required.

SysTimeData[0] : Year
SysTimeData[1] : Month
SysTimeData[2] : Date
SysTimeData[3] : Time
SysTimeData[4] : Minute
SysTimeData[5] : Second

[out] *ElapsedTime*

Elapsed time (result of acquisition)

■ **Return value**

0 : Normal end

Other than 0: Error

YppGetJogSpeed

Acquires the current jog speed.

■ **Syntax**
int YppGetJogSpeed(out short JogSpeed)

■ **Parameter**
[out] *JogSpeed*

Jog speed

Value	Description
0	Inching
1	Low speed
2	Middle speed
3	High speed
4	Maxinum speed

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetJogCoord

Acquires the current jog coordinates.

■ **Syntax**
int YppGetJogCoord(out short JogCoord)

■ **Parameter**
[out] *JogCoord*

Jog coordinates

Value	Description
0	Joint coordinates
1	Cartesian coordinates
2	Cylindrical coordinates
3	Tool coordinates
4	User coordinates
5	External reference point coordinates
6	Teaching line coordinates

■ **Return value**
0 : Normal end

Other than 0: Error

YppGetSVarInfo

Acquires the value of S variable.

■ **Syntax**
int YppGetSVarInfo(ushort DataIndex, out string SVarData)

■ **Parameter**
[in] *DataIndex*

Parameter

[out] *SVarData*

S variable value (S variable: up to 16 characters) (result of acquisition)

■ **Return value**
0 : Normal end

Other than 0 : Error

4 SYSTEM CONTROL API

YppSetVarData

Sets the value of the variable (B, I, D, R).

■ **Syntax**

int YppSetVarData(ushort DataType, ushort DataIndex, int sData)

■ **Parameter**

[in] *DataType*

Variable type

Value	Description
1	B variable
2	I variable
3	D variable
4	R variable

[in] *DataIndex*

Variable number

[in] *sData*

Variable value (set value)

■ **Return value**

0 : Normal end

Other than 0 : Error

■ **Relevant parameter**

Number	Function	Setting value	Default
S2C541	Permission of variable and I/O writing during play mode	0: Permitted 1: Not permitted	1 (Not permitted)
S2C542	Permission of variable and I/O writing during edit lock	0: Permitted 1: Not permitted	1 (Not permitted)

CAUTION

When “0” is set to “S2C541”, writing operation during play back mode becomes valid. However, please operate the manipulator with caution since this operation may influence the cycle time.

CAUTION

The “edit lock status” to which S2C542 can specify are following status.

– In alarm status
– External memory is being used
– Data transmitting function is being used
– Specified input EDIT_LOCK (#40064) is turned ON.

YppWriteIO

Writes I/O.

■ **Syntax**
int YppWriteIO(uint Address, uint Value)

■ **Parameter**
[in] *Address*

Address of I/O

<DX100>

Value	Description
10010 - 12567	Universal output #10010 - #12567 (2048)
60010 - 60647	I/F panel #60010 - #60647 (512)
25010 - 27567	Network input #25010 - #27567 (2048)
1000000 - 1000559	Register #1000000 - #1000559 (560)

<DX200>

Value	Description
10010 - 15127	Universal output #10010 - #15127 (4096)
60010 - 60647	I/F panel #60010 - #60647 (512)
27010 - 29567	Network input #27010 - #29567 (2048)
1000000 - 1000559	Register #1000000 - #1000559 (560)

<FS100>

Value	Description
1001 - 11287	Universal output #10010 - #11287 (1024)
60010 - 60647	I/F panel #60010 - #60647 (512)
25010 - 26287	Network input #25010 - #26287 (1024)
1000000 - 1000559	Register #1000000 - #1000559 (560)

[in] *Value*

Inputting value

■ **Return value**
0 : Normal end

Other than 0: Error

■ **Relevant parameter**

Number	Function	Setting value	Default
S2C541	Permission of variable and I/O writing during play mode	0: Permitted 1: Not permitted	1 (Not permitted)
S2C542	Permission of variable and I/O writing during edit lock	0: Permitted 1: Not permitted	1 (Not permitted)

CAUTION

When “0” is set to “S2C541”, writing operation during play back mode becomes valid. However, please operate the manipulator with caution since this operation may influence the cycle time.

CAUTION

The “edit lock status” to which S2C542 can specify are following status.

– In alarm status
– External memory is being used
– Data transmitting function is being used
– Specified input EDIT_LOCK (#40064) is turned ON.

YppSetPosVarData

Sets the position-type variable

Syntax

int YppSetPosVarData(ushort DataType, ushort DataIndex, int[] sData)

Parameter

[in] *DataType*

Variable type

Value	Description
5	Robot
6	Base
7	Station

[in] *DataIndex*

Variable number

[in] *sData (int[10])*

Position data (set value)

Note: ARea for int[10] is required.

Array	Description	
	Bit No.	**Content**
rData[0]	D05 - D00	Variable type 0 Pulse 16 Cartesian (base coordinates) 17 Cartesian (robot coordinates) 18 Cartesian (tool coordinates) 19 Cartesian (user coordinates) 20 Cartesian (reserved for master tool)
	D07 - D06	Reserved by manufacturer
	D08	0:Front 1:Back
	D09	0:Upper arm 1: Lower arm
	D10	0:Flip 1:No flip
	D11	0:R<180deg 1:R>=180deg
	D12	0:T<180deg 1:T>=180deg
	D13	0:S<180de 1:S>=180deg
	D14 - D15	Reserved by manufacturer
	D16 - D21	Tool number (0 - 23)
	D22 - D27	User coordinate number
	D28 - D31	Reserved by manufacturer
rData[1]		(Extended attribute)
Array	**Pulse**	**Cartesian**
rData[2]	1st axis (S) pulse value	X-axis coordinate (unit: micron)
rData[3]	2nd axis (L) pulse value	Y-axis coordinate (unit: micron)
rData[4]	3rd axis (U) pulse value	Z-axis coordinate (unit: micron)
rData[5]	4th axis (R) pulse value	Wrist angle Rx (unit: 0.0001deg)
rData[6]	5th axis (B) pulse value	Wrist angle Ry (unit: 0.0001deg)
rData[7]	6th axis (T) pulse value	Wrist angle Rz (unit: 0.0001deg)
rData[8]	7th axis(E) pulse value	angle Re (unit: 0.0001deg)
rData[9]	8th axis pulse value	8th axis pulse value (micron in the case of traveling axis)

■ **Return value**

0 : Normal end

Other than 0 : Error

■ **Relevant parameter**

Number	Function	Setting value	Default
S2C541	Permission of variable and I/O writing during play mode	0: Permitted 1: Not permitted	1 (Not permitted)
S2C542	Permission of variable and I/O writing during edit lock	0: Permitted 1: Not permitted	1 (Not permitted)

CAUTION

When “0” is set to “S2C541”, writing operation during play back mode becomes valid. However, please operate the manipulator with caution since this operation may influence the cycle time.

CAUTION

The “edit lock status” to which S2C542 can specify are following status.

– In alarm status
– External memory is being used
– Data transmitting function is being used
– Specified input EDIT_LOCK (#40064) is turned ON.

YppCancelError

Releases the error status.

■ **Syntax**
int YppCancelError(out ushort ErrorNo)

■ **Parameter**
[out] *ErrorNo*

Error number

Value	Description
0x0000	Normal end

■ **Return value**
0 : Normal end

Other than 0 : Error

YppResetAlarm

Resets the alarm

■ **Syntax**
int YppResetAlarm(out ushort ErrorNo)

■ **Parameter**
[out] *ErrorNo*

Error number

Value	Description
0x0000	Normal

■ **Return value**
0 : Normal end

Other than 0 : Error

YppSetCycle

Sets the cycle mode.

■ **Syntax**
int YppSetCycle(short Cycle, out ushort ErrorNo)

■ **Parameter**
[in] *Cycle*

Cycle

Value	Description
1	Step
2	One Cycle
3	Auto

[out] *ErrorNo*

Error number

Value	Description
0x0000	Normal
0x2060	In alarm status

■ **Return value**
0 : Normal end

Other than 0 : Error

YppSetServoPower

Sets ON/OFF of the servo power.

■ **Syntax**

int YppSetServoPower(short Power, out ushort ErrorNo)

■ **Parameter**

[in] *Power*

Servo power

Value	Description
0	Servo Power OFF
1	Servo Power ON

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2060	In alarm/error status (The case of Servo ON only)
0x3450	Servo power ON failed

■ **Return value**

0 : Normal end

Other than 0 : Error

YppSetMasterJob

Registers a specified job as a master job.

■ Syntax

int YppSetMasterJob(short TaskNo, string JobName, out ushort ErrorNo)

■ Parameter

[in] *TaskNo*

Task number

Value	Description
0	Master Task
1	SubTask 1
2	SubTask 2
3	SubTask 3
4	SubTask 4
5	SubTask 5
6	SubTask 6
7	SubTask 7
8	SubTask 8
9	SubTask 9
10	SubTask 10
11	SubTask 11
12	SubTask 12
13	SubTask 13
14	SubTask 14
15	SubTask 15

[in] *JobName*

Job name (up to 32 characters for a job name)

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2060	In alarm/error status
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ Return value

0 : Normal end

Other than 0 : Error

YppSetCurrentJob

Set the specified job and line number as the current job.

■ **Syntax**

int YppSetCurrentJob(ushort LineNo, string JobName, out ushort ErrorNo)

■ **Parameter**

[in] *LineNo*

Line number of the job

[in] *JobName*

Job name (up to 32 characters available)

[out] *ErrorNo*

Error number

Member: *<err_no>* error number

Value	Description
0x0000	Normal
0x2010	Manipulator is in operation
0x2060	In alarm/error status
0x2110	Inaccessible job
0x4040	Specified job not found
0x5200	Specified Line No. is out of range

■ **Return value**

0 : Normal end

Other than 0 : Error

YppStartJob

Registers a specified job as a master job and starts it.

Syntax

int YppStartJob(short TaskNo, string JobName, out ushort ErrorNo)

Parameter

[in] *TaskNo*

Task number

Value	Description
0	Master Task
1	SubTask 1
2	SubTask 2
3	SubTask 3
4	SubTask 4
5	SubTask 5
6	SubTask 6
7	SubTask 7
8	SubTask 8
9	SubTask 9
10	SubTask 10
11	SubTask 11
12	SubTask 12
13	SubTask 13
14	SubTask 14
15	SubTask 15

[in] *JobName*

Job name (up to 32 characters for a job name)

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Normal end

Other than 0: Error

YppHold

Turns ON/OFF the hold function.

■ **Syntax**
int YppHold(short Hold, out ushort ErrorNo)

■ **Parameter**
[in] *Hold*

Hold status

Value	Description
0	Hold OFF
1	Hold ON

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded

■ **Return value**
0 : Normal end

Other than 0: Error

YppWaitForJobEnd

Waits for the job completion or expiration of specified time duration.

Syntax

int YppWaitForJobEnd(short TaskNo, short WiatTime, out ushort ErrorNo)

Parameter

[in] *TaskNo*

Task number

Value	Description
0	Master Task
1	SubTask 1
2	SubTask 2
3	SubTask 3
4	SubTask 4
5	SubTask 5
6	SubTask 6
7	SubTask 7
8	SubTask 8
9	SubTask 9
10	SubTask 10
11	SubTask 11
12	SubTask 12
13	SubTask 13
14	SubTask 14
15	SubTask 15

[in] *WaitTime*

Waiting time

Note: If “-1” is specified, waits unlimitedly

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0xFFFF	Other error

Return value

0 : Normal end

Other than 0 : Error

YppDeleteJob

Deletes the specified job.

■ **Syntax**
int YppDeleteJob(string JobName, out ushort ErrorNo)

■ **Parameter**
[in] *JobName*

Job name (up to 32 characters available)

[out] *ErrorNo*

Error number

Value	Description
0x0000	Normal
0x2010	Manipulator is in operation or cannot delete JOBs during the robot operation
0x2060	In alarm/error status
0x2080	Play mode
0x2090	Accessing to the specified job
0x4020	Edit lock job
0x4040	Specified JOB not found

■ **Return value**
0 : Normal end

Other than 0 : Error

YppSetSVarInfo

Sets the value of S variable

■ **Syntax**
int YppSetVarData(ushort DataIndex, string SVarData)

■ **Parameter**
[in] *DataIndex*

Variable number

[in] *SVarData*

S variable value (S variable: 16 characters at maximum) (set value)

■ **Return value**
0 : Normal end

Other than 0 : Error

■ **Relevant parameter**

Number	Function	Setting value	Default
S2C541	Permission of variable and I/O writing during play mode	0: Permitted 1: Not permitted	1 (Not permitted)
S2C542	Permission of variable and I/O writing during edit lock	0: Permitted 1: Not permitted	1 (Not permitted)

CAUTION

When “0” is set to “S2C541”, writing operation during play back mode becomes valid. However, please operate the manipulator with caution since this operation may influence the cycle time.

CAUTION

The “edit lock status” to which S2C542 can specify are following status.

– In alarm status
– External memory is being used
– Data transmitting function is being used
– Specified input EDIT_LOCK (#40064) is turned ON.

5 MOTION CONTROL API

YppIMOV

Moves the current position of the manipulator with the incremental value of linear motion.

■ **Syntax**
int YppIMOV(uint Ctrlgrp, int Speed, short VType, short Frame, short ToolNo, int[] Pos, out ushort ErrorNo)

■ **Parameters**
[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)

Value	Description
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[in] *Speed*

Motion speed Unit : micron/s, 0.0001deg/s

[in] *VType*

Selection of motion speed

Value	Description
0	Control point
1	Position angular

[in] *Frame*

Coordinate system ID

Value	Description
0	Base coordinate
1	Robot coordinate
2, 3, …	User coordinate 1, 2, …

[in] *ToolNo*

Tool number 0 to 63

[in] *Pos (int[12])*

Specified position (Setting Value) ※Requires area for int[12]

Robot axis

Value	Description
Pos[0]	X-axis coordinate (unit: micron)
Pos[1]	Y-axis coordinate (unit: micron)
Pos[2]	Z-axis coordinate (unit: micron)
Pos[3]	Wrist angle Rx (unit: 0.0001deg)
Pos[4]	Wrist angle Ry (unit: 0.0001deg)

Value	Description
Pos[5]	Wrist angle Rz (unit: 0.0001deg)
Pos[6]	angle Re (unit: 0.0001deg)
Pos[7]	(Reserved)
Pos[8]	(Not in use)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

External axis

Value	Description
Pos[0]	(Not in use)
Pos[1]	(Not in use)
Pos[2]	(Not in use)
Pos[3]	(Not in use)
Pos[4]	(Not in use)
Pos[5]	(Not in use)
Pos[6]	1st external axis pulse value (micron in the case of traveling axis)
Pos[7]	2nd external axis pulse value (micron in the case of traveling axis)
Pos[8]	3rd external axis pulse value (micron in the case of traveling axis)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

Set "0" for data Pos[8] to Pos[11] if the system has no external axis.

Set “0” for data Pos[0] to Pos[5] and Pos[9] to Pos[11] if the system has any external axes.

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)

Value	Description
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Function succeeded.

Nonzero: Function failed.

YppMOVJ

Moves the manipulator to the specified position with the joint motion.

■ **Syntax**

int YppMOVJ(uint CtrlGrp, int Speed, short Frame, short Config, short ToolNo, int[] Pos, out ushort ErrorNo)

■ **Parameters**

[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[in] *Speed*

Motion speed 1 to 10000 (Unit : 0.01%) ※1 - 10000 representing 0.01 to 100.0%

[in] *Frame*

Coordinate system ID

Value	Description
0	Base coordinate
1	Robot coordinate
2, 3, …64	User coordinate 1, 2, …63

[in] *Config*

Configuration

Value	Description
D00	0:Front 1:Back
D01	0:Upper arm 1:Lower arm
D02	0:Flip 1:No flip
D03	0:R<180deg 1:R>=180deg
D04	0:T<180deg 1:T>=180deg
D05	0:S<180deg 1:S>=180deg
D06-D15	Reserved

[in] *ToolNo*

Tool number 0 to 63

[in] *Pos (int[12])*

Specified position (Setting Value) ※Requires area for int[12]

Robot axis

Value	Description
Pos[0]	X-axis coordinate (unit: micron)

Value	Description
Pos[1]	Y-axis coordinate (unit: micron)
Pos[2]	Z-axis coordinate (unit: micron)
Pos[3]	Wrist angle Rx (unit: 0.0001deg)
Pos[4]	Wrist angle Ry (unit: 0.0001deg)
Pos[5]	Wrist angle Rz (unit: 0.0001deg)
Pos[6]	angle Re (unit: 0.0001deg)
Pos[7]	(Reserved)
Pos[8]	(Not in use)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

External axis

Value	Description
Pos[0]	(Not in use)
Pos[1]	(Not in use)
Pos[2]	(Not in use)
Pos[3]	(Not in use)
Pos[4]	(Not in use)
Pos[5]	(Not in use)
Pos[6]	1st external axis pulse value (micron in the case of traveling axis)
Pos[7]	2nd external axis pulse value (micron in the case of traveling axis)
Pos[8]	3rd external axis pulse value (micron in the case of traveling axis)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

Set "0" for data Pos[8] to Pos[11] if the system has no external axis.

Set “0” for data Pos[0] to Pos[5] and Pos[9] to Pos[11] if the system has any external axes.

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status

Value	Description
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Function succeeded.

Nonzero: Function failed.

YppMOVL

Moves the manipulator to the specified position with the linear motion.

■ **Syntax**

int YppMOVL(uint CtrlGrp, int Speed, short VType, short Frame, short Config, short ToolNo, int[] Pos, out ushort ErrorNo)

■ **Parameters**

[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[in] *Speed*

Motion speed Unit : micron/s, 0.0001deg/s

[in] *VType*

Selection of motion speed

Value	Description
0	Control point
1	Position angular

[in] *Frame*

Coordinate system ID

Value	Description
0	Base coordinate
1	Robot coordinate
2, 3, …	User coordinate 1, 2, …

[in] *Config*

Configuration

Value	Description
D00	0:Front 1:Back
D01	0:Upper arm 1:Lower arm
D02	0:Flip 1:No flip
D03	0:R<180deg 1:R>=180deg
D04	0:T<180deg 1:T>=180deg
D05	0:S<180deg 1:S>=180deg
D06-D15	Reserved

[in] *ToolNo*

Tool number 0 to 63

[in] *Pos (int[12])*

Specified position (Setting Value) ※Requires area for int[12]

Robot axis

Value	Description
Pos[0]	X-axis coordinate (unit: micron)
Pos[1]	Y-axis coordinate (unit: micron)
Pos[2]	Z-axis coordinate (unit: micron)
Pos[3]	Wrist angle Rx (unit: 0.0001deg)
Pos[4]	Wrist angle Ry (unit: 0.0001deg)
Pos[5]	Wrist angle Rz (unit: 0.0001deg)
Pos[6]	angle Re (unit: 0.0001deg)
Pos[7]	(Reserved)
Pos[8]	(Not in use)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

External axis

Value	Description
Pos[0]	(Not in use)
Pos[1]	(Not in use)
Pos[2]	(Not in use)
Pos[3]	(Not in use)
Pos[4]	(Not in use)
Pos[5]	(Not in use)
Pos[6]	1st external axis pulse value (micron in the case of traveling axis)
Pos[7]	2nd external axis pulse value (micron in the case of traveling axis)
Pos[8]	3rd external axis pulse value (micron in the case of traveling axis)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

Set "0" for data Pos[8] to Pos[11] if the system has no external axis.

Set “0” for data Pos[0] to Pos[5] and Pos[9] to Pos[11] if the system has any external axes.

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation

Value	Description
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Function succeeded.

Nonzero: Function failed.

YppPulseMOVJ

Moves the manipulator to the specified pulse position with the joint motion.

■ **Syntax**

int YppPulseMOVJ(uint CtrlGrp, int Speed, short ToolNo, int[] Pos, out ushort ErrorNo)

■ **Parameters**

[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)
30	S15 (Station 15)

Value	Description
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[in] *Speed*

Motion speed 1 to 10000 (Unit : 0.01%) ※1 - 10000 representing 0.01 to 100.0%

[in] *ToolNo*

Tool number 0 to 63

[in] *Pos (int[12])*

Specified position (Setting Value) ※Requires area for int[12]

Robot axis

Value	Description
Pos[0]	1st axis (S) pulse value
Pos[1]	2nd axis (L) pulse value
Pos[2]	3rd axis (U) pulse value
Pos[3]	4th axis (R) pulse value
Pos[4]	5th axis (B) pulse value
Pos[5]	6th axis (T) pulse value
Pos[6]	7th axis (E) pulse value
Pos[7]	8th axis pulse value
Pos[8]	(Not in use)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

External axis

Value	Description
Pos[0]	(Not in use)
Pos[1]	(Not in use)
Pos[2]	(Not in use)
Pos[3]	(Not in use)
Pos[4]	(Not in use)

Value	Description
Pos[5]	(Not in use)
Pos[6]	(Not in use)
Pos[7]	(Not in use)
Pos[8]	1st extermal axis pulse value
Pos[9]	2nd extermal axis pulse value
Pos[10]	3rd extermal axis pulse value
Pos[11]	4th extermal axis pulse value

Set "0" for data lPos[8] to lPos[11] if the system has no external axis.

Set “0” for data Pos[0] to Pos[5] if the system has any external axes.

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Function succeeded.

Nonzero: Function failed.

YppPulseMOVL

Moves the manipulator to the specified pulse position with the linear motion.

Syntax

YppPulseMOVL(uint CtrlGrp, int Speed, short VType, short ToolNo, int[] Pos, out ushort ErrorNo)

Parameters

[in] *CtrlGrp*

Control group

Value	Description
0	R1 (Robot 1)
1	R2 (Robot 2)
2	R3 (Robot 3)
3	R4 (Robot 4)
4	R5 (Robot 5)
5	R6 (Robot 6)
6	R7 (Robot 7)
7	R8 (Robot 8)
8	B1 (Base 1)
9	B2 (Base 2)
10	B3 (Base 3)
11	B4 (Base 4)
12	B5 (Base 5)
13	B6 (Base 6)
14	B7 (Base 7)
15	B8 (Base 8)
16	S1 (Station 1)
17	S2 (Station 2)
18	S3 (Station 3)
19	S4 (Station 4)
20	S5 (Station 5)
21	S6 (Station 6)
22	S7 (Station 7)
23	S8 (Station 8)
24	S9 (Station 9)
25	S10 (Station 10)
26	S11 (Station 11)
27	S12 (Station 12)
28	S13 (Station 13)
29	S14 (Station 14)

Value	Description
30	S15 (Station 15)
31	S16 (Station 16)
32	S17 (Station 17)
33	S18 (Station 18)
34	S19 (Station 19)
35	S20 (Station 20)
36	S21 (Station 21)
37	S22 (Station 22)
38	S23 (Station 23)
39	S24 (Station 24)

[in] *Speed*

Motion speed 1Unit : micron/s, 0.0001deg/s

[in] *VType*

Selection of motion speed

Value	Description
0	Control point
1	Position angular

[in] *ToolNo*

Tool number 0 to 63

[in] *Pos (int[12])*

Specified position (Setting Value) ※Requires area for int[12]

Robot axis

Value	Description
Pos[0]	1st axis (S) pulse value
Pos[1]	2nd axis (L) pulse value
Pos[2]	3rd axis (U) pulse value
Pos[3]	4th axis (R) pulse value
Pos[4]	5th axis (B) pulse value
Pos[5]	6th axis (T) pulse value
Pos[6]	7th axis (E) pulse value
Pos[7]	8th axis pulse value
Pos[8]	(Not in use)
Pos[9]	(Not in use)
Pos[10]	(Not in use)
Pos[11]	(Not in use)

External axis

Value	Description
Pos[0]	(Not in use)
Pos[1]	(Not in use)
Pos[2]	(Not in use)
Pos[3]	(Not in use)
Pos[4]	(Not in use)
Pos[5]	(Not in use)
Pos[6]	(Not in use)
Pos[7]	(Not in use)
Pos[8]	1st external axis pulse value
Pos[9]	2nd external axis pulse value
Pos[10]	3rd external axis pulse value
Pos[11]	4th external axis pulse value

Set "0" for data lPos[8] to lPos[11] if the system has no external axis.

Set “0” for data lPos[0] to lPos[5] if the system has any external axes.

[out] *ErrorNo*

Error number

Value	Description
0x0000	Succeeded
0x2010	Robot is in operation
0x2030	In HOLD status (Pendant)
0x2040	In HOLD status (External)
0x2050	In HOLD status (Command)
0x2060	In alarm/error status
0x2070	In SERVO OFF status
0x2080	In TEACH mode status
0x3040	The origin position is not registered
0x3050	Out of range (ABSO data)
0x3400	Cannot operate MASTER JOB
0x3410	In TEACH LOCK status; or the JOB name is already registered
0x4040	Specified JOB not found

■ **Return value**

0 : Function succeeded.

Nonzero: Function failed.

6 FILE TRANSFER API

YppLoadFile

Loads the specified file.

■ **Syntax**
int YppLoadFile(int Media, string LoadPath, string FileName)

■ **Parameter**
[in] *Media*

Medias in which loading files exist.

Value	Description
1	Compact Flash (PP)
2	USB memory (PP)

[in] *loadPath*

Pointer to the folder name in which loading files exist.

- When the file exists in the route folder of the specified media, specify an empty character string (“”).
- When the number of hierarchy of the holder is two or more, use “\\” to partition them.
(Example: aaa\\bbb)

[in] *fileName*

Name of the loading file.

■ **Return value**
0 : Normal end

Other than 0 : Error

CAUTION

In consideration of the program migration from the controller, by specifying 0 (IMedia=0) to the first parameter, specification with the full path name, which is same as the full path name of the controller, to savePath is enabled.

However, on the other hand, it is recommended to specify “1” or ”2” (IMedia=1 or 2) and use the Compact Flash (PP) or the USB memory for loading/saving the file.

YppSaveFile

Saves the specified file.

Syntax

int YppSaveFile(int Media, string SavePath, string FileName)

Parameter

[in] *lMedia*

Medias for saving files

Value	Description
1	Compact Flash (PP)
2	USB memory (PP)

[in] *savePath*

Folder name in which the file is saved.

- When saving the file in the route folder of the specified media, specify an empty character string (“”).
- When the hierarchy of the holder is two or more, use “\\” to partition them.
  (Example: aaa\\bbb)

[in] *fileName*

File name to be saved.

Return value

0 : Normal end

Other than 0 : Error

CAUTION

In consideration of the program migration from the controller, by specifying 0 (IMedia=0) to the first parameter, specification with the full path name, which is same as the full path name of the controller, to savePath is enabled.

However, on the other hand, it is recommended to specify “1” or ”2” (IMedia=1 or 2) and use the Compact Flash (PP) or the USB memory for loading/saving the file.

YppRefreshFileList

Saves the specified file.

■ **Syntax**
int YppRefreshFileList(short Extension)

■ **Parameter**
[in] *extension*

File type

Value	Description
1	Independent job file (JBI)
2	Related job file (JBR)

■ **Return value**
0 : Normal end

Other than 0 : Error

YppGetFileCount

Acquires the file number in the file list.

■ **Syntax**
int YppGetFileCount()

■ **Parameter**
No parameter

■ **Return value**

0	: Normal end
Other than 0	: Error

YppGetFileName

Acquires the file name in the file list.

■ **Syntax**
int YppGetFileName(int Index, out string JobName)

■ **Paramete**
[in] *index*

Index of the acquiring file.

[out] *SavePath*

File name of the specified index (result of acquisition)

■ **Return value**
0 : Normal end

Other than 0 : Error

Appendix A Revision History

Date	CEN / ECN	Revision No.	Reason For Revision	Initials
7/30/2013	33549	0	Original Release	JFC
5/20/2014	14-0680M	1	1. Revised part number on manual from 166153-1CD to 166583-1CD. 2. Updated the offices on the back cover. 3. Added Revision History Appendix A	JFC

YASKAWA

DX100/DX200/FS100 OPTIONS INSTRUCTIONS

REFERENCE MANUAL
FOR PROGRAMMING PENDANT CUSTOMIZATION FUNCTION

HEAD OFFICE
2-1 Kurosakishiroishi, Yahatanishi-ku, Kitakyushu 806-0004, Japan
Phone +81-93-645-7703 Fax +81-93-645-8140

YASKAWA America Inc. (Motoman Robotics Division)
100 Automation Way, Miamisburg, OH 45342, U.S.A.
Phone +1-937-847-6200 Fax +1-937-847-6277

YASKAWA Europe GmbH (Robotics Divsion)
Yaskawastrasse 1, 85391 Allershausen, Germany
Phone +49-8166-90-100 Fax +49-8166-90-103

YASKAWA Nordic AB
Bredbandet 1 vån. 3 varvsholmen 392 30 Kalmar, Sweden
Phone +46-480-417-800 Fax +46-480-417-999

YASKAWA ELECTRIC (China) Co., Ltd.
12F, Carlton Building, No. 21 HuangHe Road, HuangPu District, Shanghai 200003, China
Phone +86-21-5385-2200 Fax +86-21-5385-3299

YASKAWA SHOUGANG ROBOT Co. Ltd.
No7 Yongchang North Road, Beijing E&T Development Area, China 100176
Phone +86-10-6788-2858 Fax +86-10-6788-2878

YASKAWA India Private Ltd. (Robotics Division)
#426, Udyog Vihar, Phase- IV, Gurgaon, Haryana, India
Phone +91-124-475-8500 Fax +91-124-475-8542

YASKAWA Electric Korea Co., Ltd
9F, Kyobo Securities Bldg., 26-4, Yeouido-dong, Yeongdeungpo-gu, Seoul 150-737, Korea
Phone +82-2-784-7844 Fax +82-2-784-8495

YASKAWA Electric Taiwan Corporation
12F, No.207, Sec. 3, Beishin Rd., Shindian District, New Taipei City 23143, Taiwan
Phone +886-2-8913-1333 Fax +886-2-8913-1513

YASKAWA Electric (Singapore) PTE Ltd.
151 Lorong Chuan, #04-02A, New Tech Park, Singapore 556741
Phone +65-6282-3003 Fax +65-6289-3003

YASKAWA Electric (Thailand) Co., Ltd.
252/125-126 27th Floor, Tower B Muang Thai-Phatra Complex Building,
Rachadaphisek Road, Huaykwang, Bangkok 10320, Thailand
Phone +66-2693-2200 Fax +66-2693-4200

PT. YASKAWA Electric Indonesia
Menara Anugrah Lantai 1, Kantor Taman E.3.3, JI Mega Kuningan Lot 8.6-8.7, Kawasan
Mega Kuningan, Jakarta, Indonesia
Phone +62-21-57941845 Fax +62-21-57941843

YASKAWA
MOTOMAN ROBOTICS

MANUAL NO.
HW1481485 ①