

MOTOCOM32

FUNCTION REFERENCE

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



MANDATORY

- **This manual explains the MOTOCOM32. Read this manual carefully and be sure to understand its contents before operation.**
- **General items related to safety are listed in instruction manuals supplied with the manipulator. To ensure correct and safe operation, carefully read the instructions on safety before reading this manual.**



CAUTION

- **Some drawings in this manual are shown with the protective covers or shields removed for clarity. Be sure all covers and shields are replaced before operating this product.**
- **The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.**
- **YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.**
- **If your copy of the manual is damaged or lost, contact a YASKAWA representative to order a new copy. The representatives are listed on the back cover. Be sure to tell the representative the manual number listed on the front cover.**
- **YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids your product's warranty.**
- **Software described in this manual is supplied against licensee only, with permission to use or copy under the conditions stated in the license. No part of this manual may be copied or reproduced in any form without written consent of YASKAWA.**

Notes for Safe Operation

Before using this product, read this manual and all the other related documents carefully to ensure knowledge about the product and safety, including all the cautions.

In this manual, the Notes for Safe Operation are classified as “WARNING”, “CAUTION”, “MANDATORY”, or “PROHIBITED”.



WARNING

Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.



CAUTION

Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.



MANDATORY

Always be sure to follow explicitly the items listed under this heading.



PROHIBITED

Must never be performed.

Even items described as “CAUTION” may result in a serious accident in some situations. At any rate, be sure to follow these important items.



To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as “CAUTION” and “WARNING”.

Notation for Menus and Buttons

Descriptions of the programming pendant, buttons, and displays are shown as follows:

Item	Manual Designation
Menu	The menus displayed on screen are denoted with { }. ex. {TOOL}.
Button	The buttons, check boxes, radio buttons displayed on screen are denoted with []. ex. [Close]; [Sync] check box; [Fast] radio button.

Description of the Operation Procedure

In the explanation of the operation procedure, the expression "Select • • •" means the following operations:

- To move the cursor to the object item and left-click on it with the mouse.
- To pick out the object item by the tab key and press the Enter key.
(In case of selecting a menu, use arrow keys instead of the tab key to pick out the object item, then press the Enter key.)

Registered Trademark

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and TM are omitted.

1 INTRODUCTION

1.1	MOTOCOM32	12
1.2	Features of Ethernet Communications	13
■	High speed transmission	13
■	Transmissions between a multiple number of HOSTS	13
1.3	Hardware Requirements for MOTOCOM32	15
1.3.1	RS-232C Transmission cable specifications	16
1.4	Hardware Lock Key	17
■	Multiple personal computer access	28
■	CMOS batch storage	28

7 COMMUNICATION TRANSMISSION

7.1	Outline	94
7.2	File Data Transmission Function	95
■	BscDownload	96
■	BscDownloadEx	97
■	BscUpload	98
■	BscUploadEx	99
7.3	Robot Control Function	100
■	BscFindFirst	101
■	BscFindFirstMaster	102
■	BscFindNext	103
■	BscFindNextMaster	104

■ BscGetCtrlGroup	105
■ BscGetCtrlGroupXrc	106
■ BscGetCtrlGroupDX	108
■ BscGetError	110
■ BscGetError2	111
■ BscGetFirstAlarm	112
■ BscGetFirstAlarmS	113
■ BscGetNextAlarm	114
■ BscGetNextAlarmS	115
■ BscGetStatus	116
■ BscGetUFrame	117
■ BscGetVarData	119
■ BscGetVarData2	122
■ BscHostGetVarData	125
■ BscHostGetVarDataM	129
■ BscGetVarDataEx	130
■ BsclsAlarm	134
■ BsclsCtrlGroup	135
■ BsclsCtrlGroupXrc	136
■ BsclsCtrlGroupDX	138
■ BsclsCycle	140
■ BsclsError	141
■ BsclsErrorCode	142
■ BsclsHold	143
■ BsclsJobLine	144
■ BsclsJobName	145
■ BsclsJobStep	146
■ BsclsLoc	147
■ BscGetPulsePos	149
■ BsclsPlayMode	150
■ BsclsRemoteMode	151
■ BsclsRobotPos	152
■ BscGetCartPos	154
■ BsclsServo	156
■ BsclsTaskInf	157
■ BsclsTaskInfXrc	158
■ BsclsTeachMode	159
■ BscJobWait	160
■ BscReadAlarmS	161
■ BscCancel	162
■ BscChangeTask	163
■ BscContinueJob	164
■ BscConvertJobP2R	165
■ BscConvertJobR2P	166
■ BscDeleteJob	167
■ BscHoldOff	168
■ BscHoldOn	169
■ BscHostPutVarData	170
■ BscHostPutVarDataM	174
■ BscPutVarDataEx	175

■ BscImov	179
■ BscImovEx	181
■ BscImovEx2	183
■ BscMDSP	185
■ BscMov	186
■ BscMovEx	188
■ BscMovEx2	190
■ BscMovj	192
■ BscMovjEx	194
■ BscMovl	196
■ BscMovlEx	198
■ BscOPLock	200
■ BscOPUnLock	202
■ BscPMov	203
■ BscPMovEx	204
■ BscPMovj	205
■ BscPMovjEx	206
■ BscPMovl	207
■ BscPMovlEx	208
■ BscPutUFrame	209
■ BscPutUFrameEx2	211
■ BscPutVarData	213
■ BscPutVarData2	217
■ BscStartJob	221
■ BscSelectJob	222
■ BscSelectMode	223
■ BscSelLoopCycle	224
■ BscSelOneCycle	225
■ BscSelStepCycle	226
■ BscSetLineNumber	227
■ BscSetMasterJob	228
■ BscReset	229
■ BscSetCtrlGroup	230
■ BscSetCtrlGroupXrc	231
■ BscSetCtrlGroupDX	233
■ BscServoOff	235
■ BscServoOn	236
7.4 DCI Function	237
■ BscDCILoadSave	238
■ BscDCILoadSaveOnce	239
■ BscDCIGetPos	240
■ BscDCIGetPos2	242
■ BscDCIGetVarData	244
■ BscDCIGetVarDataEx	247
■ BscDCIPutPos	250
■ BscDCIPutPos2	252
■ BscDCIPutVarData	254
■ BscDCIPutVarDataEx	257
7.5 I/O Signal Read/Write Function	260

■ BscReadIO	261
■ BscReadIO2	263
■ BscWriteIO	265
■ BscWriteIO2	267
7.6 Other Functions	269
■ BscClose	270
■ BscCommand	271
■ BscConnect	272
■ BscDisconnect	273
■ BscDiskFreeSizeGet	274
■ BscEnforcedClose	275
■ BscGets	276
■ BscInBytes	277
■ BscOpen	278
■ BscOutBytes	279
■ BscPuts	280
■ BscReConnect	281
■ BscReStartJob	282
■ BscSetBreak	283
■ BscSetCom	284
■ BscSetCondBSC	285
■ BscSetEServerMode	286
■ BscSetEther	287
■ BscStatus	288
7.7 DLL Functions Corresponding to Transmission-related	
Key words	289
7.7.1 DLL Functions Related to Transmission Commands	289
■ Read/Monitoring System	289
■ Read/Data Access System	290
■ Operation System	290
■ Editing System	291
■ Job Selection System	291
■ Startup System	291
■ Other DLL Functions	292
7.7.2 DLL Functions Related to DCI Function	292
7.7.3 DLL Functions Related to I/O Read/Write	292
7.7.4 DLL Functions Related to Personal Computer Communications	
Port	293
7.7.5 Other DLL Functions	293

8 Appendix

8.1 Frequently-asked questions	294
■ When the driver has been installed with USB type key connected to a personal computer	294
■ When the previous version key driver has been installed after installing the key driver	294

1 INTRODUCTION

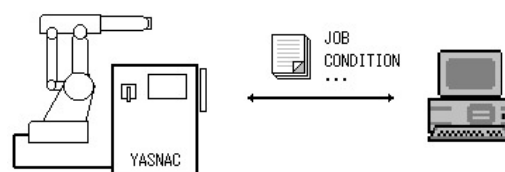
1.1 MOTOCOM32

MOTOCOM32 is software for data transmission between a personal computer and YASKAWA industrial robot controller FS100, DX100, NX100, YASNAC XRC, MRC, MRCII, ERC and ERCII.

High-speed transmission can be achieved by connecting the FS100, DX100, NX100 or YASNAC type controller and the personal computer with a serial cable (RC232C cable), or by connecting the FS100 or DX100 or NX100 or YASNAC XRC or MRC to a LAN by using a Ethernet cable (the YASNAC XRC or MRC requires mounting an Ethernet I/F board). This software operates on a personal computer and has the following function

High Speed JobExchanger Function

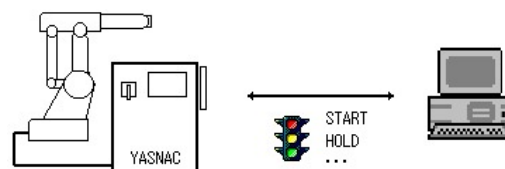
Loads or saves files such as jobs or condition data according to the command from a personal computer.



Host Control Function

Can perform the following tasks easily according to the command from a personal computer.

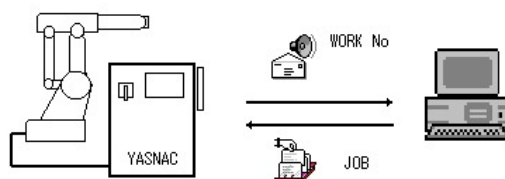
- Reads robot status (current position, alarm, error, servo status, etc.) or controls the system (start, hold, job call, etc.)
- Reads or writes I/O signals.



Automatic Work Job Replacing Function

Can automatically replace a work job using DCI (data communication by instruction) transmitting function.

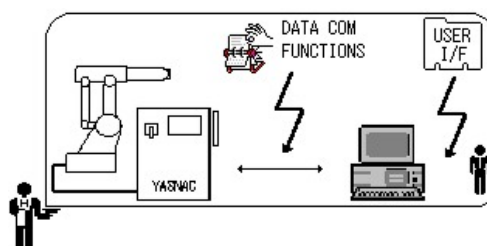
In order to compensate for the memory shortage of robots, a personal computer is used as external memory to receive work number information from the robot and send back corresponding jobs.



Transmission Application Function

Supplies the following information so that the user can develop transmission applications between a robot and personal computer.

- Supplies data transmission functions between robot and personal computer. (Motocom32.dll)
- Describes application preparation procedures using sample programs including the above functions.



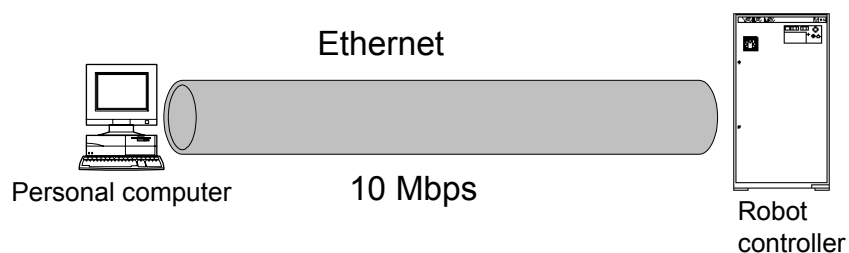
1.2 Features of Ethernet Communications

The Ethernet I/F board and the "Ethernet" function of the MOTOCOM32 transmit data at higher than normal speeds.

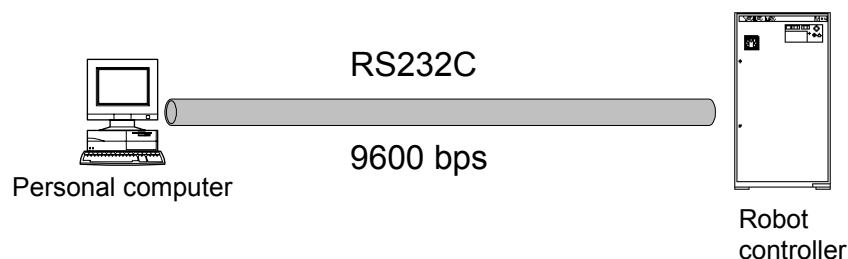
■ High speed transmission

In comparison with transmissions using RS232C, higher speed transmissions are possible with the Ethernet.

When Ethernet is used



When RS232C is used



NOTE The above transmission speed is the communication speed between network devices, not including the time used for format check of transmitted data, etc.

■ Transmissions between a multiple number of HOSTS

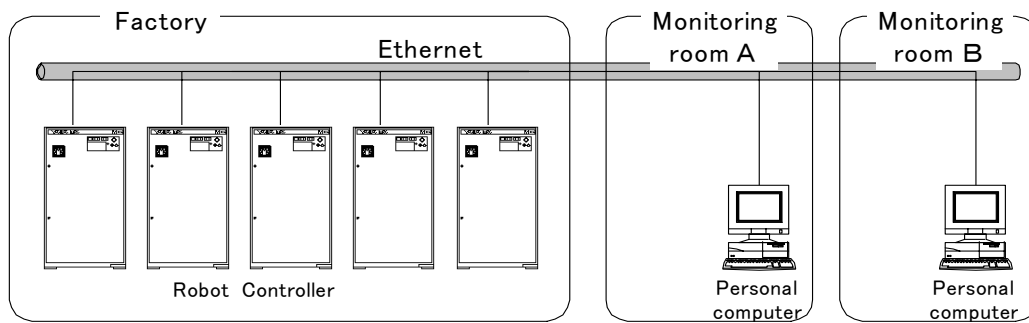
As N:N transmission is possible with an Ethernet cable, the following system configurations can be prepared.

NOTE Refer to paragraph "2.4 Restrictions" with the following Configuration Examples.

<Configuration Example>

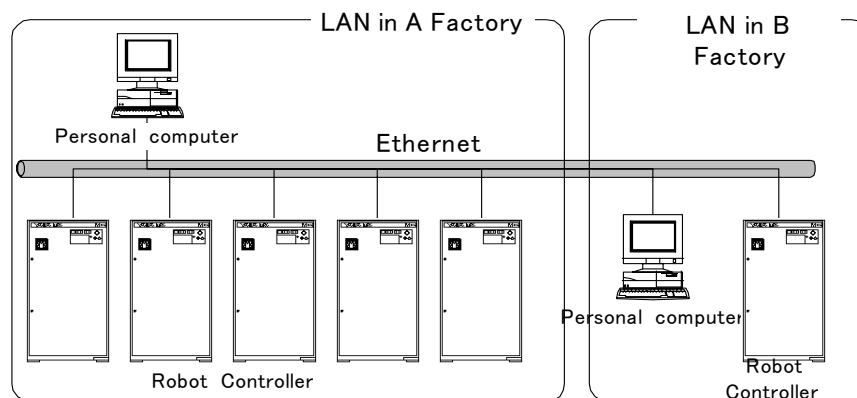
Since an Ethernet cable can be connected to a multiple number of network devices, the fac-

tory operation state and alarm occurrences can be monitored from several places.



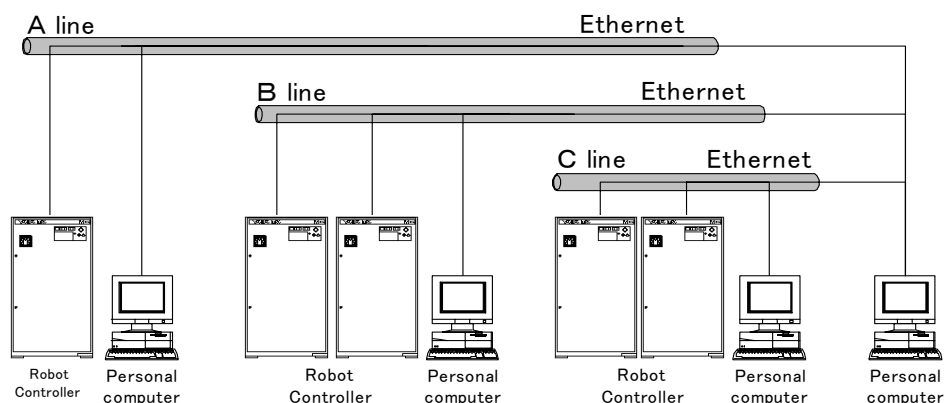
<Configuration Example 2>

By connecting the LANs of different factories with one Ethernet cable, transmission in each factory can be executed simultaneously. The transmission between the robot controller and the personal computer in factory A does not interfere with the transmission between the robot controller and the personal computer in factory B. Transmission can also be done between factories. (In both cases, the settings should be correct.)



<Configuration Example 3>

With the Ethernet cables, the job on a personal computer can be executed on the robot controller by installing a personal computer for each production line and transferring the job from the personal computers to the robot controller. Then, by connecting one personal computer to the Ethernet cables in all the production lines, monitoring of the state of all the production lines and data backup can be executed.



1.3 Hardware Requirements for MOTOCOM32

OS	Microsoft Windows 98 / NT4.0 / 2000 / XP(32-bit) / Vista(32-bit) / 7(32-bit) ^{*1}
Required Memory	128 Mbyte or more
Hardware Disk Capacity for Installation	50 Mbyte or more
Display	Supported by MS-Windows
Robot Controller	FS100, DX100, NX100, YASNAC XRC, MRC, MRC2, ERC, and ERC2.
Transmission Cable	Ethernet cable or RS232C cable
Hardware Lock Key	Used under single user environment. For details, refer to the following section " 1.4 Hardware Lock Key ".

^{*1} MS-Windows98,MS-WindowsNT4.0,MS-Windows2000,MS-WindowsXP, MS-Windows Vista and MS-Windows 7 is a registered trademark of Microsoft Corporation, USA.



- The controller Data Transmission function, Ethernet function, Ethernet board, transmission cable, or the personal computer OS are not included in this package.
- Use either an RS-232C cable or an Ethernet cable for transmission, depending on the data transmission function specifications set in the robot controller manuals. Before starting this software, check the hardware and software specifications of the robot controllers.
- Ethernet transmission is not available for the YASNAC MRC2/ ERC / ERC2 since they do not support the Ethernet function. For MRC, Ethernet transmission is available for version 4.111 or later.
- To create a transmission application, a development tool such as Microsoft Visual Basic Ver6.0 or Microsoft Visual C++ Ver6.0 is required.

For softwares and devices, refer to the robot controller Operator's Manuals, Data Transmission Operator's Manual, Ethernet I/F Board Instructions (or "Ethernet Function Instructions" for the FS100, DX100, NX100), Manuals for MS-Windows, etc.

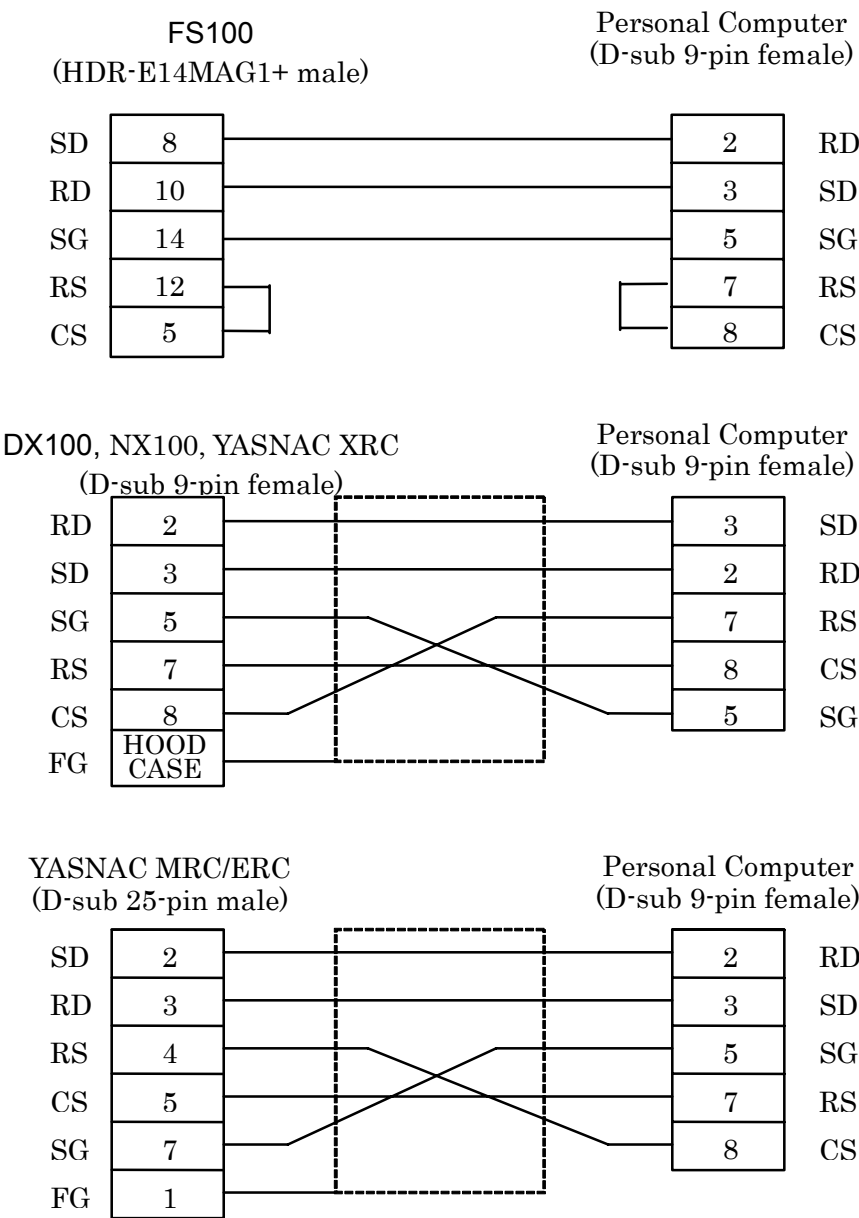
1.3.1 RS-232C Transmission cable specifications

RS-232C Transmission cable specifications are as follows.

NOTE

When using an Ethernet cable, the RS232C cable is not required.

[PC/AT]



1.4 Hardware Lock Key

For proper operation, connect provided hardware lock key (USB type) to personal computer before using this software.

Check and execute <Check the computing environment> <Installation of driver> before connecting the key to USB port.

<Check the computing environment>

Multi-connection of USB type key is not available for one USB port because of hardware structure. Therefore, only one key should be connected to one USB port. When installing multiple offline software into one personal computer and multi-connectiong USB keys, use the personal computer which is provided same numbers of USB ports as the number of software to be installed.

<Installation of driver>



Please install the driver after detaching the all sentinel hardware key from the personal computer.

Execute "\\SentinelDriver\\Sentinel System Driver Installer 7.5.7.exe" of installation CD-ROM. Refer to "\\SentinelDriver\\Manual\\SafeNet_Sentinel_EndUser_Guide.pdf" for the details of installation.



- Be sure to install the driver.
- When installing the driver, be sure to login in administrator mode in order to add files to system folder and input information in registry.
- If a key is connected to personal computer before installing the driver, the message concerning the driver is displayed. In this case, detach the key from personal computer and then install the driver.

If a key is connected to personal computer before installing the driver under Windows 95/98/NT4.0/2000/XP environment, Windows wizard ([Add New Hardware] Wizard) starts up. In this case, push [cancel], and detach the key from personal computer and then install the driver.

- When installing the driver under Windows NT4.0, 2000 environment, please install the driver located in the folder "\\SentinelDriver\\SSD5411\\SSD5411-32bit.EXE" of installation CD-ROM.

For the driver installation procedure, please consult the installation manual "\\Sentinel-Driver\\SSD5411\\Manual\\us\\Readme.pdf".

Refer to " **8.1 Frequently-asked questions** " for other countermeasures concerning hardware lock key.

7 COMMUNICATION TRANSMISSION

7.1 Outline

MOTOCOM32.DLL is a transmission library that controls the data transmission function of the FS100, DX100, NX100, the YASNAC XRC, MRC, ERC, and ERC2 on a personal computer. This library is composed in the form of Microsoft Windows DLL (Dynamic Link Library).



MOTOCOM32.DLL is located below the MOTOCOM32 installation directory. When a transmission application is created, copy this file to the same directory as the application. MOTOCOM.H and MOTOCOM32.LIB files are provided in the MOTOCOM32 installation directory. Use these files when a transmission application is created in C-language.

Transmission library has the following functions.

- File data transmission function
- Robot control function
- DCI function
- I/O signal read/write function
- Other functions

7.2 File Data Transmission Function

Loads and saves the files containing job, condition data, system information, etc.
The following functions are available.

BscDownload
BscDownloadEx
BscUpload
BscUploadEx

■ BscDownload

FUNCTION	Sends a specified file to the robot controller.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDownload(short nCid,char *fname);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *fname File name to be sent.
	OUT (Return) None
	Return Value 0 : Normal completion Others : Transmission error
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	“BscDownloadEx” “BscUpload” “BscUploadEx”

■ BscDownloadEx

FUNCTION	Sends a specified file to the robot controller. A directory where the sending file exists can be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDownloadEx(short nCid,char *fname, char *path, BOOL nFlg);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*fname File name to be sent</p> <p>*path Diretory path of sending source data</p> <p>nFlg TRUE : Changes the diretory temporarily and restores it at the end.</p> <p> FALSE : Changes the diretory and completes the processing.</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Transmission error</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	“BscDownload” “BscUpload” “BscUploadEx”

■ BscUpload

FUNCTION	Receives a specified file from the robot controller.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscUpLoad(short nCid,char *fname);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *fname File name to be received
	OUT (Return) None
	Return Value 0 : Normal completion Others : Receiving error
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	“BscUploadEx” “BscDownload” “BscDownloadEx”

■ BscUploadEx

FUNCTION	Receives a specified file from the robot controller. The directory where the file is send to can be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscUpLoadEx(short nCid,char *fname, char *path, BOOL nFlg);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*fname File name to be received</p> <p>*path Diretory path of sending source data</p> <p>nFlg TRUE : Changes the directory temporarily and restores it at the end.</p> <p> FALSE : Changes the directory and completes the processing.</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Receiving error</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	“BscUpload” “BscDownload” “BscDownloadEx”

7.3 Robot Control Function

Reads the robot status (current position, alarm, error, servo status, etc.) and controls the system (start, hold, job call, etc.)

The following functions are available.

Status Read

BscFindFirst
 BscFindFirstMaster
 BscFindNext
 BscFindNextMaster
 BscGetCtrlGroup
 BscGetCtrlGroupXrc
 BscGetCtrlGroupDX
 BscDownload
 BscDownloadEx
 BscGetError
 BscGetError2
 BscGetFirstAlarm
 BscGetFirstAlarmS
 BscGetNextAlarm
 BscGetNextAlarmS
 BscGetStatus
 BscGetUFrame
 BscGetVarData
 BscGetVarData2
 BscHostGetVarData
 BscHostGetVarDataM
 BscGetVarDataEx
 BscIsAlarm
 BscIsCtrlGroup
 BscIsCtrlGroupXrc
 BscIsCtrlGroupDX
 BscIsCycle
 BscIsError
 BscIsErrorCode
 BscIsHold
 BscIsJobLine
 BscIsJobName
 BscIsJobStep
 BscIsLoc
 BscGetPulsePos
 BscIsPlayMode
 BscIsRemoteMode
 BscIsRobotPos
 BscGetCartPos

BscIsServo
 BscIsTaskInf
 BscIsTaskInfXrc
 BscIsTeachMode
 BscJobWait
 BscReadAlarmS

System Control

BscCancel
 BscChangeTask
 BscContinueJob
 BscConvertJobP2R
 BscConvertJobR2P
 BscDeleteJob
 BscHoldOff
 BscHoldOn
 BscHostPutVarData
 BscHostPutVarDataM
 BscPutVarDataEx
 BscImov
 BscImovEx
 BscMDSP
 BscMov
 BscMovEx
 BscMovj
 BscMovjEx
 BscMovl
 BscMovlEx
 BscOPLock
 BscOPUnLock
 BscPMov
 BscPMovEx
 BscPMovj
 BscPMovjEx
 BscPMovl
 BscPMovlEx
 BscPutUFrame
 BscPutVarData
 BscPutVarData2
 BscStartJob
 BscSelectJob
 BscSelectMode
 BscSelLoopCycle
 BscSelOneCycle
 BscSelStepCycle
 BscSetLineNumber
 BscSetMasterJob

BscReset
 BscSetCtrlGroup
 BscSetCtrlGroupXrc
 BscSetCtrlGroupDX
 BscServoOff
 BscServoOn
 BscUpload
 BscUploadEx

■ BscFindFirst

FUNCTION	Reads the first job name from the all job list registered at the present time.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscFindFirst(short nCid,char *fname,short size);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *fname First job name storage pointer size Job name storage area size</p> <p>OUT (Return) *fname First job name storage pointer</p> <p>Return Value -1 : No job -2 : Internal error (memory allocation error) -3 : Internal error (memory lock error) -4 : Other errors 0 : Job found</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscFindNext"

■ BscFindFirstMaster

FUNCTION	Reads the first job name from the job list that belongs to the target job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscFindFirstMaster(short nCid,char *fname,short size);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *fname First job name storage pointer size Job name storage area size</p> <p>OUT (Return) *fname First job name storage pointer</p> <p>Return Value -1 : No job -2 : Internal error (memory allocation error) -3 : Internal error (memory lock error) -4 : Other errors 0 : Job found</p>
REMARKS	<p>Call Condition The BscSelectJob function must be called up and the target job name must be selected before executing this function.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscFindNextMaster" "BscSelectJob"

■ BscFindNext

FUNCTION	Reads the next job name registered at the present time.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscFindNext(short nCid,char *fname,short size);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *fname N-th job name storage pointer size Job name storage area size</p> <p>OUT (Return) *fname N-th job name storage pointer</p> <p>Return Value -1 : No next job 0 : Next job found</p>
REMARKS	<p>Call Condition The BscFindFirst function must be called up before executing this function.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	" BscFindFirst "

■ BscFindNextMaster

FUNCTION	Reads the next job name in the job list that belongs to the target job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscFindNextMaster(short nCid,char *fname,short size);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *fname N-th job name storage pointer size Job name storage area size</p> <p>OUT (Return) *fname N-th job name storage pointer</p> <p>Return Value -1 : No next job 0 : Next job found</p>
REMARKS	<p>Call Condition The BscFindFirstMaster function must be called up before executing this function.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	" BscFindFirstMaster "

■ BscGetCtrlGroup

FUNCTION	Reads control group and task information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetCtrlGroup(short nCid,short *groupinf,short *taskinf);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *groupinf Control group information storage pointer *taskinf Task information storage pointer</div><div>OUT (Return) *groupinf Control group information storage pointer *taskinf Task information storage pointer</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>
REMARKS	<div><div>Restrictions This function is effective only for transmission with the MRC. Refer to the BscGetCtrlGroupDX for transmission with the DX100. Refer to the BscGetCtrlGroupXrc for transmission with the FS100/ NX100/XRC.</div><div>Control Group Information The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : S1 (Station1) D3 : S2 (Station2) D4 : S3 (Station3) D5 : S4 (Station4) D6 : S5 (Station5) D7 : S6 (Station6)</div></div><div>Task Information The task information is represented as follows. 0 : Master task 1 : Sub 1 task 2 : Sub 2 task "0" is returned if independent control is not allowed in the system.</div></div>
CONTROLLER	MRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BsclsCtrlGroupDX" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BsclsCtrlGroupXrc" "BsclsTaskInfXrc" "BscSetCtrlGroup" "Bscls- CtrlGroup" "BsclsTaskInf" "BscChangeTask"


■ BscGetCtrlGroupXrc

FUNCTION	Reads control group and task information.																								
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetCtrlGroupXrc(short nCid,short *groupinf,short *stationinf,short *taskinf);</code>																								
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *groupinf Control group information storage pointer (robot axis) *stationinf Control group information storage pointer (station axis) *taskinf Task information storage pointer</div><div>OUT (Return) *groupinf Control group information storage pointer (robot axis) *stationinf Control group information storage pointer (station axis) *taskinf Task information storage pointer</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>																								
REMARKS	<div><div>Restrictions This function is effective for transmission with the FS100/NX100/XRC. Refer to the BscGetCtrlGroupDX for transmission with the DX100. Refer to BscGetCtrlGroup for transmission with the MRC.</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div>D7 D6 D5 D4 D3 D2 D1 D0</div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : R3 (Robot3) D3 : R4 (Robot4)</div><div>Control Group Information (Station Axis) The control group information is represented by bit data in decimals. <div>D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0</div><table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12)</div></div>																								

REMARKS	Task Information The task information is represented as follows. 0 : Master task 1 : Sub 1 task 2 : Sub 2 task 3 : Sub 3 task 4 : Sub 4 task 5 : Sub 5 task 6 : Sub 6 task 7 : Sub 7 task "0" is returned if independent control is not allowed in the system.
CONTROLLER	FS100, NX100, XRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BscIsCtrlGroupDX" "BscSetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInfXrc" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsCtrlGroup" "BscIsTaskInf" "BscChangeTask"

■ BscGetCtrlGroupDX

FUNCTION	Reads control group and task information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetCtrlGroupDX(short nCid,long *groupinf,long *stationinf,short *taskinf);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *groupinf Control group information storage pointer (robot axis) *stationinf Control group information storage pointer (station axis) *taskinf Task information storage pointer</div><div>OUT (Return) *groupinf Control group information storage pointer (robot axis) *stationinf Control group information storage pointer (station axis) *taskinf Task information storage pointer</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>
REMARKS	<div><div>Restrictions This function is effective for transmission with the DX100. Refer to BscGetCtrlGroupXrc for transmission with the FS100/ NX100/XRC. Refer to BscGetCtrlGroup for transmission with the MRC.</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : R3 (Robot3) D3 : R4 (Robot4) : : D7 : R8 (Robot8)</div></div></div></div>

REMARKS	<p>Control Group Information (Station Axis)</p> <p>The control group information is represented by bit data in decimals.</p> <p style="text-align: center;">D23 ... D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0</p> <div style="text-align: center;">  </div> <p>D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12) : : D23 : S24 (Station24)</p>
	<p>Task Information</p> <p>The task information is represented as follows.</p> <p>0 : Master task 1 : Sub 1 task 2 : Sub 2 task 3 : Sub 3 task 4 : Sub 4 task 5 : Sub 5 task 6 : Sub 6 task 7 : Sub 7 task</p> <p>"0" is returned if independent control is not allowed in the system.</p>
CONTROLLER	DX100 (Serial Port, Ethernet)
REFERENCE	<p>"BscSetCtrlGroupDX" "BscIsCtrlGroupDX" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInfXrc" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsCtrlGroup" "BscIsTaskInf" "BscChangeTask"</p>

■ BscGetError

FUNCTION	Reads an error code or alarm code.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetError(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : No error Others : Error codes
REMARKS	Restrictions This function is effective for transmission with the ERC. Refer to BscGetError2 for transmission with the FS100/DX100/ NX100/XRC/MRC.
CONTROLLER	ERC (Serial Port)
REFERENCE	"BscGetFirstAlarm" "BscGetNextAlarm" "BsclsAlarm" "BsclsError" "BsclsErrorCode"

■ BscGetError2

FUNCTION	Reads an error code or alarm code.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetError2(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : No error Others : Error codes
REMARKS	Restrictions This function is effective for transmission with the FS100/DX100/ NX100/XRC/MRC. Refer to BscGetError for transmission with the ERC.
CONTROLLER	DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetFirstAlarm" "BscGetNextAlarm" "BsclsAlarm" "BsclsError" "BsclsErrorCode"

■ BscGetFirstAlarm

FUNCTION	Reads an alarm code and returns the alarm code and alarm data.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetFirstAlarm(short nCid,short *data);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *data Alarm data storage pointer
	OUT (Return) *data Alarm data storage pointer
	Return Value 0 : No alarm Others : Alarm code numbers
REMARKS	Call Condition The BscGetError2 function must be called up before executing this function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetError2" "BscGetNextAlarm" "BsclsAlarm"

■ BscGetFirstAlarmS

FUNCTION	Reads an alarm code and returns the alarm code, alarm data and alarm message.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetFirstAlarmS(short nCid,short *data,char *msg);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *data Alarm data storage pointer *msg Alarm message storage pointer</p> <p>OUT (Return) *data Alarm data storage pointer *msg Alarm message storage pointer</p> <p>Return Value 0 : No alarm Others : Alarm code numbers</p>
REMARKS	<p>Call Condition The BscReadAlarmS function must be called up before executing this function.</p> <p>Restrictions This function is effective for transmission with the FS100/DX100/NX100.</p>
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)
REFERENCE	"BscReadAlarmS" "BscGetNextAlarm" "BscIsAlarm"

■ BscGetNextAlarm

FUNCTION	Reads the next alarm code and alarm data.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetNextAlarm(short nCid,short *data);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *data Alarm data storage pointer</p> <p>OUT (Return) *data Alarm data storage pointer</p> <p>Return Value 0 : No alarm Others : Alarm code numbers</p>
REMARKS	<p>Call Condition The BscGetFirstAlarm function must be called up before executing this function.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetError2" "BscGetFirstAlarm" "BsclsAlarm"

■ BscGetNextAlarmS

FUNCTION	Reads the next alarm code, alarm data and alarm message.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetNextAlarmS(short nCid,short *data,char *msg);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *data Alarm data storage pointer *msg Alarm message storage pointer</p> <p>OUT (Return) *data Alarm data storage pointer *msg Alarm message storage pointer</p> <p>Return Value 0 : No alarm Others : Alarm code numbers</p>
REMARKS	<p>Call Condition The BscGetFirstAlarmS function must be called up before executing this function.</p> <p>Restrictions This function is effective for transmission with the FS100/DX100/NX100.</p>
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)
REFERENCE	“BscReadAlarmS” “BscGetFirstAlarmS” “BsclsAlarm”

■ BscGetStatus

FUNCTION	Reads the status information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetStatus(short nCid,short *d1,short *d2);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *d1 Data 1 storage pointer *d2 Data 2 storage pointer</div><div>OUT (Return) *d1 Data 1 storage pointer *d2 Data 2 storage pointer</div><div>Return Value -1 : Acquisition Failure Others : Normal completion</div></div>
REMARKS	<div><div>Data 1 Data 1 are represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div>D0 : Step D1 : 1-cycle D2 : Auto operation D3 : Operating D4 : Operation at safe speed D5 : Teach * D6 : Play * D7 : Command remote * * : Effective only for DX100, NX100, XRC and MRC.</div></div><div><div>Data 2 Data 2 are represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div>D0 : Hold (DX100/NX100/XRC/MRC: Playback box hold, ERC:Panel hold) D1 : Hold (DX100/NX100/XRC/MRC: Programming pendant hold, ERC: T-BOX hold) D2 : Hold (External hold) D3 : Hold (Command hold) D4 : Alarm occurred D5 : Error occurred D6 : Servo ON</div></div></div></div>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	“BscStatus” “BsclsAlarm” “BsclsCycle” “BsclsHold” “BsclsPlayMode” “BsclsRemoteMode” “BsclsServo” “BsclsTeachMode”

■ BscGetUFrame

FUNCTION	Reads specified user frame data.																		
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetUFrame (short nCid,char *ufname,double *p);</code>																		
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*ufname Storage pointer of user coordinate name</p> <p>*p User coordinate data storage pointer</p> <p>OUT (Return)</p> <p>*p User coordinate data storage pointer</p> <p>Return Value</p> <p>-1 : User coordinate name error</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																		
REMARKS	<p>Restrictions</p> <p>This function does not support the robot with 7 axes or more.</p> <p>User Coordinate Name</p> <p>The following coordinate names correspond to the user coordinate numbers.</p> <table border="1"> <thead> <tr> <th>User Coordinate Name</th><th>Specified Name</th></tr> </thead> <tbody> <tr> <td>User coordinate 1</td><td>UF1</td></tr> <tr> <td>User coordinate 2</td><td>UF2</td></tr> <tr> <td>User coordinate 3</td><td>UF3</td></tr> <tr> <td>⋮</td><td>⋮</td></tr> <tr> <td>User coordinate 60</td><td>UF60</td></tr> <tr> <td>User coordinate 61</td><td>UF61</td></tr> <tr> <td>User coordinate 62</td><td>UF62</td></tr> <tr> <td>User coordinate 63</td><td>UF63</td></tr> </tbody> </table> <p>* User coordinate numbers 9 to 16 are effective for FS100.</p> <p>* User coordinate numbers 9 to 63 are effective for DX100.</p> <p>* User coordinate numbers 9 to 24 are effective for NX100/XRC/MRC.</p>	User Coordinate Name	Specified Name	User coordinate 1	UF1	User coordinate 2	UF2	User coordinate 3	UF3	⋮	⋮	User coordinate 60	UF60	User coordinate 61	UF61	User coordinate 62	UF62	User coordinate 63	UF63
User Coordinate Name	Specified Name																		
User coordinate 1	UF1																		
User coordinate 2	UF2																		
User coordinate 3	UF3																		
⋮	⋮																		
User coordinate 60	UF60																		
User coordinate 61	UF61																		
User coordinate 62	UF62																		
User coordinate 63	UF63																		

REMARKS	<div><div>Variable type</div><div>Coordinate values of the user coordinate system specified with the user coordinate number are assigned to the user coordinate data as follows.</div><table><thead><tr><th>Variables</th><th>Coordinate System</th><th>Meaning</th></tr></thead><tbody><tr><td>P[0]</td><td rowspan="7">ORG</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[6]</td><td>Form</td></tr><tr><td>P[7]</td><td rowspan="7">XX</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[8]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[9]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[10]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[11]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[12]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[13]</td><td>Form</td></tr><tr><td>P[14]</td><td rowspan="7">XY</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[15]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[16]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[17]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[18]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[19]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[20]</td><td>Form</td></tr><tr><td>P[21]</td><td colspan="2">Tool number (0 to 23)</td></tr><tr><td>P[22]</td><td colspan="2">7th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[23]</td><td colspan="2">8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[24]</td><td colspan="2">9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[25]</td><td colspan="2">10th axis pulse number</td></tr><tr><td>P[26]</td><td colspan="2">11th axis pulse number</td></tr><tr><td>P[27]</td><td colspan="2">12th axis pulse number</td></tr></tbody></table><div><div>*1 DX100 is effective down to 4 decimal places.</div><div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>0:Flip,</div><div>0:Elbow Above,</div><div>0:FrontSide,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div></div><div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1:Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div><div></div></div></div></div></div><div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D6 and D7 are disregarded.</div></div></div></div>	Variables	Coordinate System	Meaning	P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[1]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[2]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[3]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[4]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[5]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[6]	Form	P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[8]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[9]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[10]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[11]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[12]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[13]	Form	P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[15]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[16]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[17]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[18]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[19]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[20]	Form	P[21]	Tool number (0 to 23)		P[22]	7th axis pulse number (mm for traveling axis)		P[23]	8th axis pulse number (mm for traveling axis)		P[24]	9th axis pulse number (mm for traveling axis)		P[25]	10th axis pulse number		P[26]	11th axis pulse number		P[27]	12th axis pulse number	
Variables	Coordinate System	Meaning																																																																				
P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[1]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[2]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[3]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[4]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[5]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[6]		Form																																																																				
P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[8]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[9]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[10]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[11]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[12]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[13]		Form																																																																				
P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[15]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[16]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[17]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[18]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[19]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[20]		Form																																																																				
P[21]	Tool number (0 to 23)																																																																					
P[22]	7th axis pulse number (mm for traveling axis)																																																																					
P[23]	8th axis pulse number (mm for traveling axis)																																																																					
P[24]	9th axis pulse number (mm for traveling axis)																																																																					
P[25]	10th axis pulse number																																																																					
P[26]	11th axis pulse number																																																																					
P[27]	12th axis pulse number																																																																					
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																																																					
REFERENCE	“BscPutUFrame”																																																																					

■ BscGetVarData

FUNCTION	Receives variables.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetVarData(short nCid,short type,short varno,double *p);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number type Variable type varno Variable number *p Head pointer to the numeric variable storage area</p> <p>OUT (Return) *p Head pointer to the numeric variable storage area</p> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <p>Variable Types The variable types are represented as follows. 0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type 4 : Robot axis position type 5 : Base axis position type 6 : Station axis position type (pulse type only)</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	8	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	8	B-axis Pulses	T-axis Pulses	Tool Number		
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number		
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number		

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See the following for details on the coordinate system types and form.

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

REMARKS

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC/MRC**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



- 0:Flip, 1:No-Flip
- 0:Elbow Above, 1:Elbow Under
- 0:FrontSide, 1:Back Side
- 0:R<180, 1:R>=180
- 0:T<180, 1:T>=180
- 0:S<180, 1:S>=180
- Reserved

* With the MRC or MRC2, the data of D5 and D7 are disregarded.

CONTROLLER

FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)

REFERENCE

"BscPutVarData" "BscPutVarData2"

■ BscGetVarData2

FUNCTION	Receives variables. (robot with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetVarData2(short nCid,short type,short varno,double *p);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>type Variable type</p> <p>varno Variable number</p> <p>*p Head pointer to the numeric variable storage area</p> <hr/> <p>OUT (Return)</p> <p>*p Head pointer to the numeric variable storage area</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100/XRC.</p> <hr/> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type</p> <p>1 : Integer type</p> <p>2 : Double-precision type</p> <p>3 : Real type</p> <p>4 : Robot axis position type</p> <p>5 : Base axis position type</p> <p>6 : Station axis position type (pulse type only)</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	10	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	10	B-axis Pulses	T-axis Pulses	7 th axis Pulses	8 th axis Pulses	Tool Number
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See the following for details on the coordinate system types and form.

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

REMARKS

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



- 0: Flip, 1: No-Flip
- 0: Elbow Abov, 1: Elbow Under
- 0: Front Side, 1: Back Side
- 0: R<180, 1: R>=180
- 0: T<180, 1: T>=180
- 0: S<180, 1: S>=180
- Reserved

CONTROLLER

FS100, DX100, NX100, XRC (Serial Port, Ethernet)

REFERENCE

"BscPutVarData" "BscPutVarData2"

■ BscHostGetVarData

FUNCTION	Receives variables.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHostGetVarData(short nCid,short type,short varno,double *p,char *str);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>type Variable Types</p> <p>varno Variable number</p> <p>*p Head pointer to the numeric variable storage area</p> <p>*str Head pointer to the character variable storage area</p> <hr/> <p>OUT (Return)</p> <p>*p Head pointer to the numeric variable storage area</p> <p>*str Head pointer to the character variable storage area</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <p>String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</p> <hr/> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type</p> <p>1 : Integer type</p> <p>2 : Double-precision type</p> <p>3 : Real type</p> <p>4 : Robot axis position type</p> <p>5 : Base axis position type</p> <p>6 : Station axis position type (pulse type only)</p> <p>7 : String type</p>

REMARKS

Content of the numeric variable storage area
Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	8	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	8	B-axis Pulses	T-axis Pulses	Tool Number	-	-
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value.
The station axis position type variable contains the pulse type only.

See below for details on the coordinate system types and form.

Content of the character variable storage area

Variable Type Number	Data Type (Pulse / XYZ)	Number of Values	Content
			str
7	-	16	String



When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17

Declaration in C++: char S_Variable[17]

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC/MRC**

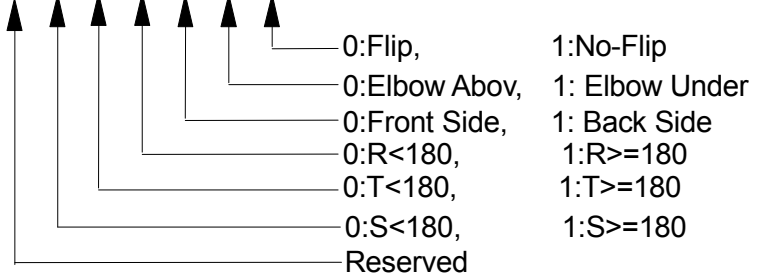
The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



* With the MRC or MRC2, the data of D5 and D7 are disregarded.

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscHostPutVarData"

■ BscHostGetVarDataM

FUNCTION	Receives multiple variables at the same time.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHostGetVarDataM(short nCid,short type,short varno,short num,double *p);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number type Variable Types varno Variable number num Number of variables *p Head pointer to the numeric variable storage area</p> <p>OUT (Return) *p Head pointer to the numeric variable storage area</p> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/DX100/NX100.</p> <p>Variable Types The variable types are represented as follows. 0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type</p> <p>Variable Designation Method The variable information transmitted is composed of the number of values (num) requested of the specified variable type, beginning with the value of the specified variable number (varno) followed by the values of subsequent variables.</p>
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)
REFERENCE	"BscHostPutVarDataM"

■ BscGetVarDataEx

FUNCTION	Receives variables. (robots with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetVarDataEx(short nCid,short type,short varno,double *p,char *str,short *axisNum);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>type Variable Types</p> <p>varno Variable number</p> <p>*p Head pointer to the numeric variable storage area</p> <p>*str Head pointer to the character variable storage area</p> <p>*axisNum Number of axis (pointer)</p> <hr/> <p>OUT (Return)</p> <p>*p Head pointer to the numeric variable storage area</p> <p>*str Head pointer to the character variable storage area</p> <p>*axisNum Number of axis (pointer)</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <p>String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</p> <hr/> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type</p> <p>1 : Integer type</p> <p>2 : Double-precision type</p> <p>3 : Real type</p> <p>4 : Robot axis position type</p> <p>5 : Base axis position type</p> <p>6 : Station axis position type (pulse type only)</p> <p>7 : String type</p>

REMARKS

Content of the numeric variable storage area
Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte	-	-	-	-
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	9	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	11	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/ XYZ)	Number of values	Content					
			P[5]	P[6]	P[7]	P[8]	P[9]	P[10]
0	-	1	-	-	-	-	-	-
1	-	1	-	-	-	-	-	-
2	-	1	-	-	-	-	-	-
3	-	1	-	-	-	-	-	-
4	Pulse	9	B-axis Pulses	T-axis Pulses	E-axis Pulses	Tool Number	-	-
4	XYZ	11	Rx Angle(deg)	Ry Angle(deg)	Rz Angle (deg)	Re (deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-	-
5	XYZ	10	Rx Angle (deg)	Ry Angle(deg)	Rz Angle (deg)	Form	Tool Number	-
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value.
The station axis position type variable contains the pulse type only.

See below for details on the coordinate system types and form.

Content of the character variable storage area

Variable Type Number	Data Type (Pulse / XYZ)	Number of Values	Content
			str
7	-	16	String



When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17

Declaration in C++: char S_Variable[17]

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC/MRC**

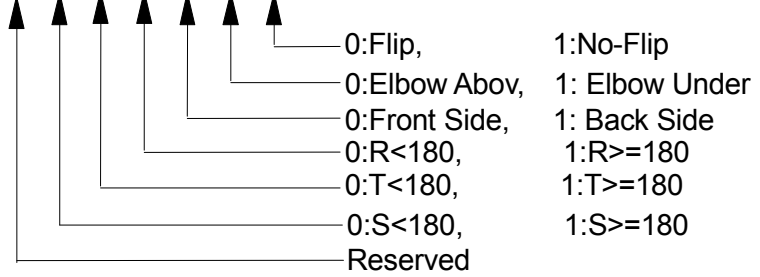
The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



* With the MRC or MRC2, the data of D5 and D7 are disregarded.

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscPutVarDataEx"

■ BscIsAlarm

FUNCTION	Reads alarm status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsAlarm(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : No alarm 1 : Alarm
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC(Serial Port)
REFERENCE	"BscGetError2" "BscGetFirstAlarm" "BscGetNextAlarm" "BscGetStatus" "BscIsCycle" "BscIsError" "BscIsHold" "BscIsPlayMode" "BscIsRemoteMode" "BscIsServo" "BscIsTeachMode"

■ BscIsCtrlGroup

FUNCTION	Reads control group information.
FORMAT	_declspec(dllexport) short APIENTRY BscIsCtrlGroup(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure Others : Control group information
REMARKS	Restrictions This function is effective only for transmission with MRC. Refer to BscIsCtrlGroupDX for transmission with DX100. Refer to BscIsCtrlGroupXrc for transmission with FS100/NX100/XRC.
	Control Group Information The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : S1 (Station1) D3 : S2 (Station2) D4 : S3 (Station3) D5 : S4 (Station4) D6 : S5 (Station5) D7 : S6 (Station6)</div></div>
CONTROLLER	MRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BscIsCtrlGroupDX" "BscIsCtrlGroupXrc" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsTaskInfXrc" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsTaskInf" "BscChangeTask"

■ BsclsCtrlGroupXrc

FUNCTION	Reads control group information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BsclsCtrlGroupXrc(short nCid,short *robtask,short *stattask);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *robtask Control group information storage pointer (robot axis) *stattask Control group information storage pointer (station axis)</div><div>OUT (Return) *robtask Control group information storage pointer (robot axis) *stattask Control group information storage pointer (station axis)</div><div>Return Value -1 : Acquisition Failure 0 : Normal completion</div></div>
REMARKS	<div><div>Restrictions This function is effective only for transmission with FS100/NX100/XRC. Refer to BsclsCtrlGroupDX for transmission with DX100. Refer to BsclsCtrlGroup for transmission with MRC.</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : R3 (Robot3) D3 : R4 (Robot4)</div></div></div></div>
	<div><div>Control Group Information (Station Axis) The control group information is represented by bit data in decimals. <div><div>D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12)</div></div></div></div>
CONTROLLER	FS100, NX100, XRC (Serial Port, Ethernet)

REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BscIsCtrlGroupDX" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsTaskInfXrc" "BscIsCtrlGroup" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsTaskInf" "BscChangeTask"
-----------	---

■ BsclsCtrlGroupDX

FUNCTION	Reads control group information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BsclsCtrlGroupDX(short nCid,long *robtask,long *stattask);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *robtask Control group information storage pointer (robot axis) *stattask Control group information storage pointer (station axis)</div><div>OUT (Return) *robtask Control group information storage pointer (robot axis) *stattask Control group information storage pointer (station axis)</div><div>Return Value -1 : Acquisition Failure 0 : Normal completion</div></div>
REMARKS	<div><div>Restrictions This function is effective only for transmission with DX100. Refer to BsclsCtrlGroupXrc for transmission with FS100/NX100/XRC. Refer to BsclsCtrlGroup for transmission with MRC.</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot1) D1 : R2 (Robot2) D2 : R3 (Robot3) D3 : R4 (Robot4) : : D7 : R8 (Robot8)</div></div></div></div>

REMARKS	<p>Control Group Information (Station Axis)</p> <p>The control group information is represented by bit data in decimals.</p> <p>D23 ... D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0</p> <table><tr><td></td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12) : : D23 : S24 (Station24)</p>		...																
	...																		
CONTROLLER	DX100 (Serial Port, Ethernet)																		
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BscIsCtrlGroupXrc" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsTaskInfXrc" "BscIsCtrlGroup" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsTaskInf" "BscChangeTask"																		

■ BscIsCycle

FUNCTION	Reads playback mode information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsCycle(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Step mode 1 : 1-cycle mode 2 : Auto mode
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsError" "BscIsHold" "BscIsPlayMode" "BscIsRemoteMode" "BscIsServo" "BscIsTeachMode"

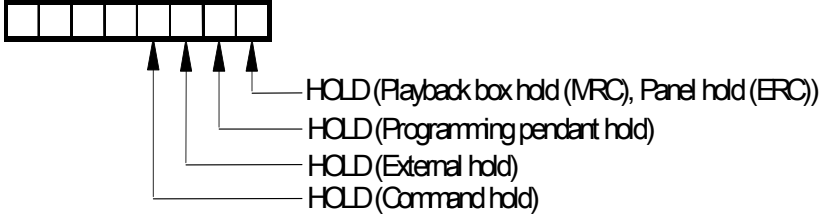
■ BscIsError

FUNCTION	Reads error status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsError(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition failure 0 : No error 1 : Error found
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetError2" "BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsErrorCode" "BscIsHold" "BscIsPlayMode" "BscIsRemoteMode" "BscIsServo" "BscIsTeachMode"

■ BscIsErrorCode

FUNCTION	Reads the error code.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsErrorCode(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : No error Others : Error codes
	Call Condition Before executing this function, the existence of an error must be confirmed by calling the <code>BscIsError</code> function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetError2" "BscIsError"

■ BscIsHold

FUNCTION	Reads hold status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsHold(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Not held Others : See below
REMARKS	Hold Status The hold status data are represented by bit data in decimals. <div style="text-align: center;"> D7 D6 D5 D4 D3 D2 D1 D0  </div>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsError" "BscIsPlayMode" "BscIsRemoteMode" "BscIsServo" "BscIsTeachMode"

■ BsclsJobLine

FUNCTION	Reads the current job line number.
FORMAT	_declspec(dllexport) short APIENTRY BsclsJobLine(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure Others : Line numbers
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BsclsJobName" "BsclsJobStep"

■ BscIsJobName

FUNCTION	Reads the current job name.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsJobName(short nCid,char *jobname,short size);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *jobname Job name storage pointer size Job name storage area size OUT (Return) *jobname Job name storage pointer Return Value -1 : Acquisition Failure 0 : Normal completion
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscIsJobLine" "BscIsJobStep"

■ BsclsJobStep

FUNCTION	Reads the current job step number.
FORMAT	_declspec(dllexport) short APIENTRY BsclsJobStep(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure Others : Step numbers
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BsclsJobName" "BsclsJobLine"

■ BsclsLoc

FUNCTION	Reads the current robot position in pulse or XYZ frame system.
FORMAT	<code>_declspec(dllexport) short APIENTRY BsclsLoc(short nCid,short ispulse,short *rconf,double *p);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>ispulse 0 : Cartesian coordinate system (Effective only for MRC and ERC.) 1 : Joint coordinate system</p> <p>*rconf Form storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>OUT (Return)</p> <p>*rconf Form storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>Return Value</p> <p>-1 : Acquisition Failure</p> <p>0 : Normal completion</p>
REMARKS	<p>From</p> <p>The form data are represented by bit data in decimals.</p> <div style="text-align: center;"> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> </div> <p>0:Flip, 1:No-Flip</p> <p>0:Elbow Above, 1:Elbow Under</p> <p>0:FrontSide, 1:Back Side</p> <p>0:R<180, 1:R>=180</p> <p>0:T<180, 1:T>=180</p> <p>0:S<180, 1:S>=180</p> <p>Reserved</p> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</p> <p>* With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Current Position</p> <p>The current position data are as follows when the joint coordinate system or Cartesian coordinate system are specified.</p> <table><tr><th></th><th>Joint coordinate system</th><th>Cartesian coordinate system</th></tr><tr><td>P[0]</td><td>S-axis pulse number</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>L-axis pulse number</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>U-axis pulse number</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>R-axis pulse number</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>B-axis pulse number</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>T-axis pulse number</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number</td><td>7th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[7]</td><td>8th axis pulse number</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr></table>		Joint coordinate system	Cartesian coordinate system	P[0]	S-axis pulse number	X-axis coordinate (unit: mm)	P[1]	L-axis pulse number	Y-axis coordinate (unit: mm)	P[2]	U-axis pulse number	Z-axis coordinate (unit: mm)	P[3]	R-axis pulse number	Wrist angle Rx (unit: °)	P[4]	B-axis pulse number	Wrist angle Ry (unit: °)	P[5]	T-axis pulse number	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number
	Joint coordinate system	Cartesian coordinate system																																						
P[0]	S-axis pulse number	X-axis coordinate (unit: mm)																																						
P[1]	L-axis pulse number	Y-axis coordinate (unit: mm)																																						
P[2]	U-axis pulse number	Z-axis coordinate (unit: mm)																																						
P[3]	R-axis pulse number	Wrist angle Rx (unit: °)																																						
P[4]	B-axis pulse number	Wrist angle Ry (unit: °)																																						
P[5]	T-axis pulse number	Wrist angle Rz (unit: °)																																						
P[6]	7th axis pulse number	7th axis pulse number (mm for traveling axis)																																						
P[7]	8th axis pulse number	8th axis pulse number (mm for traveling axis)																																						
P[8]	9th axis pulse number	9th axis pulse number (mm for traveling axis)																																						
P[9]	10th axis pulse number	10th axis pulse number																																						
P[10]	11th axis pulse number	11th axis pulse number																																						
P[11]	12th axis pulse number	12th axis pulse number																																						
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																							
REFERENCE	"BsclsRobotPos"																																							

■ BscGetPulsePos

FUNCTION	Reads the current robot position in pulse frame system.																																										
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetPulsePos(short nCid,double *p);</code>																																										
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *p Head pointer to the current position storage area																																										
	OUT (Return) *p Head pointer to the current position storage area																																										
	Return Value -1 : Acquisition Failure 0 : Normal completion																																										
REMARKS	<p>Current Position The current position data are as follows when the joint coordinate system or Cartesian coordinate system are specified.</p> <table><tr><th></th><th>Joint coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>S-axis pulse number</td><td>S-axis pulse number</td></tr><tr><td>P[1]</td><td>L-axis pulse number</td><td>L-axis pulse number</td></tr><tr><td>P[2]</td><td>U-axis pulse number</td><td>U-axis pulse number</td></tr><tr><td>P[3]</td><td>R-axis pulse number</td><td>R-axis pulse number</td></tr><tr><td>P[4]</td><td>B-axis pulse number</td><td>B-axis pulse number</td></tr><tr><td>P[5]</td><td>T-axis pulse number</td><td>T-axis pulse number</td></tr><tr><td>P[6]</td><td>7th axis pulse number</td><td>E-axis pulse number</td></tr><tr><td>P[7]</td><td>8th axis pulse number</td><td>8th axis pulse number</td></tr><tr><td>P[8]</td><td>9th axis pulse number</td><td>9th axis pulse number</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table>		Joint coordinate system	Robots with 7 axes	P[0]	S-axis pulse number	S-axis pulse number	P[1]	L-axis pulse number	L-axis pulse number	P[2]	U-axis pulse number	U-axis pulse number	P[3]	R-axis pulse number	R-axis pulse number	P[4]	B-axis pulse number	B-axis pulse number	P[5]	T-axis pulse number	T-axis pulse number	P[6]	7th axis pulse number	E-axis pulse number	P[7]	8th axis pulse number	8th axis pulse number	P[8]	9th axis pulse number	9th axis pulse number	P[9]	10th axis pulse number	10th axis pulse number	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Joint coordinate system	Robots with 7 axes																																									
P[0]	S-axis pulse number	S-axis pulse number																																									
P[1]	L-axis pulse number	L-axis pulse number																																									
P[2]	U-axis pulse number	U-axis pulse number																																									
P[3]	R-axis pulse number	R-axis pulse number																																									
P[4]	B-axis pulse number	B-axis pulse number																																									
P[5]	T-axis pulse number	T-axis pulse number																																									
P[6]	7th axis pulse number	E-axis pulse number																																									
P[7]	8th axis pulse number	8th axis pulse number																																									
P[8]	9th axis pulse number	9th axis pulse number																																									
P[9]	10th axis pulse number	10th axis pulse number																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100, NX100, XRC (Serial Port, Ethernet)																																										
REFERENCE	"BscGetCartPos"																																										

■ BscIsPlayMode

FUNCTION	Reads the operation mode.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsPlayMode(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Not operating 1 : Operating 2 : Operating at safe speed
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsError" "BscIsRemoteMode" "BscIsServo" "BscIsTeachMode"

■ BscIsRemoteMode

FUNCTION	Reads the command remote mode status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsRemoteMode(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Not command remote mode 1 : Command remote mode
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsError" "BscIsHold" "BscIsPlayMode" "BscIsServo" "BscIsTeachMode"

■ BsclsRobotPos

FUNCTION	Reads the current robot position in a specified frame system. The existence of the external axis can also be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BsclsRobotPos(short nCid,char *frameName,int isex,short *rconf,short *toolno,double *p);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*frameName Coordinate name;</p> <p> BASE : Base coordinate system;</p> <p> ROBOT : Robot coordinate system;</p> <p> UF1 : User coordinate system1...</p> <p>isex 0 : No external axis, 1 : With external axis</p> <p>*rconf Form storage pointer</p> <p>*toolno Tool number storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>OUT (Return)</p> <p>*rconf Form storage pointer</p> <p>*toolno Tool number storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>Return Value</p> <p>-1 : Acquisition Failure</p> <p>0 : Normal completion</p>
REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip</p> <p>0:Elbow Above, 1:Elbow Under</p> <p>0:FrontSide, 1:Back Side</p> <p>0:R<180, 1:R>=180</p> <p>0:T<180, 1:T>=180</p> <p>0:S<180, 1:S>=180</p> <p>Reserved</p> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</p> <p>* With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Current Position</p> <p>The current position data are as follows when the joint coordinate system or Cartesian coordinate system is specified.</p> <table border="1"> <thead> <tr> <th></th><th>Current position in the specified coordinate system</th></tr> </thead> <tbody> <tr> <td>P[0]</td><td>X-axis coordinate (unit: mm)</td></tr> <tr> <td>P[1]</td><td>Y-axis coordinate (unit: mm)</td></tr> <tr> <td>P[2]</td><td>Z-axis coordinate (unit: mm)</td></tr> <tr> <td>P[3]</td><td>Wrist angle Rx (unit: °)</td></tr> <tr> <td>P[4]</td><td>Wrist angle Ry (unit: °)</td></tr> <tr> <td>P[5]</td><td>Wrist angle Rz (unit: °)</td></tr> <tr> <td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[9]</td><td>10th axis pulse number</td></tr> <tr> <td>P[10]</td><td>11th axis pulse number</td></tr> <tr> <td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table>		Current position in the specified coordinate system	P[0]	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Current position in the specified coordinate system																										
P[0]	X-axis coordinate (unit: mm)																										
P[1]	Y-axis coordinate (unit: mm)																										
P[2]	Z-axis coordinate (unit: mm)																										
P[3]	Wrist angle Rx (unit: °)																										
P[4]	Wrist angle Ry (unit: °)																										
P[5]	Wrist angle Rz (unit: °)																										
P[6]	7th axis pulse number (mm for traveling axis)																										
P[7]	8th axis pulse number (mm for traveling axis)																										
P[8]	9th axis pulse number (mm for traveling axis)																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																										
REFERENCE	"BsclsLoc"																										

■ BscGetCartPos

FUNCTION	Reads the current robot position in a specified frame system. The existence of the external axis can also be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGetCartPos(short nCid,char *frameName,int isex,long *rconf,short *toolno,double *p);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*frameName Coordinate name;</p> <p> BASE : Base coordinate system;</p> <p> ROBOT : Robot coordinate system;</p> <p> UF1 : User coordinate system1...</p> <p>isex 0 : No external axis, 1 : With external axis</p> <p>*rconf Form storage pointer</p> <p>*toolno Tool number storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>OUT (Return)</p> <p>*rconf Form storage pointer</p> <p>*toolno Tool number storage pointer</p> <p>*p Head pointer to the current position storage area</p> <p>Return Value</p> <p>-1 : Acquisition Failure</p> <p>0 : Normal completion</p>
REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip</p> <p>0:Elbow Above, 1:Elbow Under</p> <p>0:FrontSide, 1:Back Side</p> <p>0:R<180, 1:R>=180</p> <p>0:T<180, 1:T>=180</p> <p>0:S<180, 1:S>=180</p> <p>Reserved</p> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</p> <p>* With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Current Position</p> <p>The current position data are as follows when the joint coordinate system or Cartesian coordinate system is specified.</p> <table><tr><th></th><th>Current position in the specified coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table>		Current position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Current position in the specified coordinate system	Robots with 7 axes																																									
P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																									
P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																									
P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																									
P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																									
P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																									
P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																									
P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																									
P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																									
P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																									
P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																										
REFERENCE	"BscGetPulsePos"																																										

■ BscIsServo

FUNCTION	Reads the servo status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsServo(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Servo OFF 1 : Servo ON
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsError" "BscIsHold" "BscIsPlayMode" "BscIsRemoteMode" "BscIsTeachMode" "BscServoOn" "BscServoOff"

■ BscIsTaskInf

FUNCTION	Reads task information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsTaskInf(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid : Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure Others : Task information
REMARKS	Restrictions This function is effective only for transmission with the MRC. Refer to BscIsTaskInfXrc for transmission with the FS100/DX100/NX100/XRC.
	Task Information The task information is represented as follows. 0 : Master task 1 : Sub 1 task 2 : Sub 2 task "0" is returned when independent control is not allowed in the system.
CONTROLLER	MRC (Serial Port, Ethernet)
REFERENCE	"BscIsTaskInfXrc" "BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsCtrlGroup" "BscChangeTask"

■ BscIsTaskInfXrc

FUNCTION	Reads task information.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsTaskInfXrc(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure Others : Error codes
REMARKS	Restrictions This function is effective for transmission with the FS100/DX100/NX100/XRC. Refer to BscIsTaskInf for transmission with the MRC.
	Task Information The task information is represented as follows. 0 : Master task 1 : Sub 1 task 2 : Sub 2 task 3 : Sub 3 task 4 : Sub 4 task 5 : Sub 5 task 6 : Sub 6 task 7 : Sub 7 task "0" is returned if independent control is not allowed in the system.
CONTROLLER	FS100, DX100, NX100, XRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInf" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsCtrlGroup" "BscChangeTask"

■ BscIsTeachMode

FUNCTION	Reads whether in the teach mode or play mode.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscIsTeachMode(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Acquisition Failure 0 : Teach mode 1 : Play mode
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetStatus" "BscIsAlarm" "BscIsCycle" "BscIsError" "BscIsHold" "BscIsPlayMode" "BscIsRemoteMode" "BscIsServo"

■ BscJobWait

FUNCTION	Waits for job completion until the robot stops or specified time expires.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscJobWait(short nCid,short time);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number time Wait time (-1 : Unlimited; 0 to 32767seconds)</p> <p>OUT (Return) None</p> <p>Return Value -2 : Abnormal completion -1 : Operation incomplete 0 : Operation completed Others : Error codes</p>
REMARKS	<p>Cause of Incomplete Operation These causes are considered for imcomplete operation.</p> <ul style="list-style-type: none"> * The robot was stopped via teach pendant or by external hold. * The robot was stopped by alarm. * The robot was stopped by emergency stop. * Time expired
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscReadAlarmS

FUNCTION	Reads the error code, error data and error message.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReadAlarmS(short nCid,short *data,char *msg);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *data Error data storage pointer *msg Error message storage pointer</p> <p>OUT (Return) *data Error data storage pointer *msg Error message storage pointer</p> <p>Return Value -1 : Error code acquisition failure 0 : No Error Others : Error codes numbers</p>
REMARKS	<p>Restrictions This function is effective for transmission with the FS100/DX100/NX100.</p>
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)
REFERENCE	"BscGetFirstAlarmS" "BscGetNextAlarmS" "BscGetError2"

■ BscCancel

FUNCTION	Cancels an error.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscCancel(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscReset"

■ BscChangeTask

FUNCTION	Changes a task.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscChangeTask(short nCid,short task);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number task Specified task number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Restrictions This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC. When the power supply of robot controller is started up, a master task is selected as the control task. This function cannot be used in a system where independent control is not enabled.
	Specified Task Number The task number is represented as follows. 0 : Task 1 : Sub 1 task 2 : Sub 2 task 3 : Sub 3 task 4 : Sub 4 task 5 : Sub 5 task 6 : Sub 6 task 7 : Sub 7 task * Specified task number 3 to 5 are only for the FS100. * Specified task number 3 to 7 are only for the DX100 and NX100 and XRC.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupXrc" "BscSetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInfXrc" "BscGetCtrlGroup" "BscSetCtrlGroup" "BscIsCtrlGroup" "BscIsTaskInf"

■ BscContinueJob

FUNCTION	Starts job. (Execution starts from the current line of the current job.)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscContinueJob(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscStartJob"

- BscConvertJobP2R

FUNCTION	Converts a pulse job to a relative job in a specified frame system.
FORMAT	_declspec(dllexport) short APIENTRY BscConvertJobP2R(short nCid,char *name,char *frameName);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number *name Job name storage pointer *frameName Coordinate name: BASE : Base coordinate, ROBOT : Robot coordinate, UF1 : User coordinate1 ...</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscConvertJobR2P"

■ BscConvertJobR2P

FUNCTION	Converts a relative job in a specified frame system to a pulse job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscConvertJobR2P(short nCid,char *name,short cv_type,char *var_no);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number *name Job name storage pointer cv_type Conversion method var_no Reference position variable number</p> <p>OUT (Return) None</p> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/ DX100/NX100/XRC/MRC.</p> <p>Conversion Method The conversion method is represented as follows. 0 : Previous step regarded (B-axis sign constant) 1 : Type regarded 2 : Previous step regarded (Minimum R-axis movement)</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscConvertJobP2R"

■ BscDeleteJob

FUNCTION	Deletes a job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDeleteJob(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion 1 : Cannot delete Otherss : Error codes
REMARKS	Call Condition The BscSelectJob function must be called up and the job name to be deleted must be specified before executing this function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscSelectJob" "BscSetMasterJob"

■ BscHoldOff

FUNCTION	Sets hold-OFF.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHoldOff(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	BscHoldOn

■ BscHoldOn

FUNCTION	Sets hold-ON.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHoldOn(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscHoldOff"

■ BscHostPutVarData

FUNCTION	Sets variables.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHostPutVarData(short nCid,short type,short varno,double *p, char *str);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number type Variable Types varno Variable number *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/ DX100/NX100/XRC/MRC. String variables can only be used with the FS100 or DX100 or NX100 ver3.0 or later.</p> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type 4 : Robot axis position type 5 : Base axis position type 6 : Station axis position type (pulse type only) 7 : String type</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	8	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	8	B-axis Pulses	T-axis Pulses	Tool Number	-	-
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See below for details on the coordinate system types and form.

Content of the character variable storage area

Variable Type Number	Data Type (Pulse / XYZ)	Number of Values	Content
			str
7	-	16	String



When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17
Declaration in C++: char S_Variable[17]

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC/MRC**

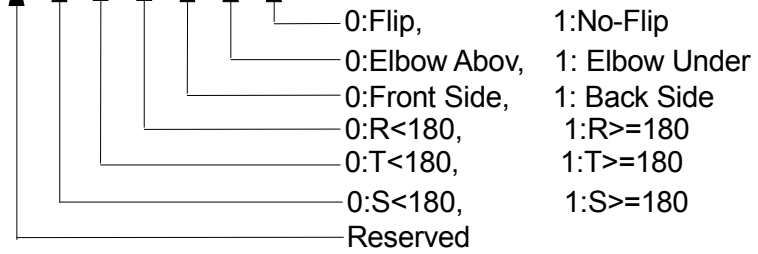
The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



* With the MRC or MRC2, the data of D5 and D7 are disregarded.

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscHostGetVarData"

■ BscHostPutVarDataM

FUNCTION	Sets multiple variables at the same time.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscHostPutVarDataM(short nCid,short type,short varno, short num, double *p);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number type Variable Types varno Variable number num Number of variables *p Head pointer to the numeric variable storage area</p> <p>OUT (Return) None</p> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/ DX100/NX100.</p> <p>Variable Types The variable types are represented as follows. 0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type</p> <p>Variable Designation Method The variable information transmitted is composed of the number of values (num) requested of the specified variable type, beginning with the value of the specified variable number (varno) followed by the values of subsequent variables.</p>
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)
REFERENCE	"BscHostGetVarDataM"

■ BscPutVarDataEx

FUNCTION	Sets variables.(robots with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BsctPutVarDataEx(short nCid,short type,short varno,double *p, char *str,short axisNum);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>type Variable Types</p> <p>varno Variable number</p> <p>*p Head pointer to the numeric variable storage area</p> <p>*str Head pointer to the character variable storage area</p> <p>axisNum Number of axis</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <p>String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</p> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type</p> <p>1 : Integer type</p> <p>2 : Double-precision type</p> <p>3 : Real type</p> <p>4 : Robot axis position type</p> <p>5 : Base axis position type</p> <p>6 : Station axis position type (pulse type only)</p> <p>7 : String type</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	9	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	11	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/ XYZ)	Number of values	Content						
			P[5]	P[6]	P[7]	P[8]	P[9]	P[10]	
0	-	1							
1	-	1	-	-	-	-	-	-	
2	-	1	-	-	-	-	-	-	
3	-	1	-	-	-	-	-	-	
4	Pulse	9	B-axis Pulses	T-axis Pulses	E-axis Pulses	Tool Number	-	-	
4	XYZ	11	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Re (deg)	Form	Tool Number	
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-	-	
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number	-	
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-	-	

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See below for details on the coordinate system types and form.

Content of the character variable storage area

Variable Type Number	Data Type (Pulse / XYZ)	Number of Values	Content
			str
7	-	16	String



When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17

Declaration in C++: char S_Variable[17]

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC/MRC**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

REMARKS	<p>Form The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved</p> <p>* With the MRC or MRC2, the data of D5 and D7 are disregarded.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
REFERENCE	"BscGetVarDataEx"

■ BscImov

FUNCTION	Moves robot with linear motion from the current position for the increment value in a specified frame system.																										
FORMAT	_declspec(dllexport) short APIENTRY Bsclmov(short nCid,char *vtype,double spd,char *framename,short toolno,double *p);																										
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular</p> <p>spd Move speed (0.1 to ****.*mm/s, 0.1 to ***.* ° /s)</p> <p>*framename Coordinate name;</p> <p style="padding-left: 180px;">BASE : Base coordinate;</p> <p style="padding-left: 180px;">ROBOT : Robot coordinate;</p> <p style="padding-left: 180px;">UF1 : User coordinate1...</p> <p style="padding-left: 180px;">TOOL : Tool coordinate (Only for NX100/XRC/MRC)</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																										
REMARKS	<p>Target Position</p> <p>The target position is represented as follows.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="background-color: #cccccc;">Target position in the specified coordinate system</th> </tr> </thead> <tbody> <tr> <td>P[0]</td> <td>X-axis coordinate (unit: mm)</td> </tr> <tr> <td>P[1]</td> <td>Y-axis coordinate (unit: mm)</td> </tr> <tr> <td>P[2]</td> <td>Z-axis coordinate (unit: mm)</td> </tr> <tr> <td>P[3]</td> <td>Wrist angle Rx (unit: °)</td> </tr> <tr> <td>P[4]</td> <td>Wrist angle Ry (unit: °)</td> </tr> <tr> <td>P[5]</td> <td>Wrist angle Rz (unit: °)</td> </tr> <tr> <td>P[6]</td> <td>7th axis pulse number (mm for traveling axis)</td> </tr> <tr> <td>P[7]</td> <td>8th axis pulse number (mm for traveling axis)</td> </tr> <tr> <td>P[8]</td> <td>9th axis pulse number (mm for traveling axis)</td> </tr> <tr> <td>P[9]</td> <td>10th axis pulse number</td> </tr> <tr> <td>P[10]</td> <td>11th axis pulse number</td> </tr> <tr> <td>P[11]</td> <td>12th axis pulse number</td> </tr> </tbody> </table> <p>* Set "0" for data P[7] to P[11] if the system has no external axis.</p>	Target position in the specified coordinate system		P[0]	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
Target position in the specified coordinate system																											
P[0]	X-axis coordinate (unit: mm)																										
P[1]	Y-axis coordinate (unit: mm)																										
P[2]	Z-axis coordinate (unit: mm)																										
P[3]	Wrist angle Rx (unit: °)																										
P[4]	Wrist angle Ry (unit: °)																										
P[5]	Wrist angle Rz (unit: °)																										
P[6]	7th axis pulse number (mm for traveling axis)																										
P[7]	8th axis pulse number (mm for traveling axis)																										
P[8]	9th axis pulse number (mm for traveling axis)																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										

REMARKS	<p>Form The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved</p> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded. * With the MRC or MRC2, the data D5 and D7 are disregarded.</p>
CONTROLLER	<p>NX100, XRC, MRC (Serial Port, Ethernet), ERC (Serial Port) * Refer to the BscImovEx2 for the FS100/DX100.</p>
REFERENCE	<p>"BscMov" "BscMovEx" "BscMovI" "BscImovEx2"</p>

■ BscImovEx

FUNCTION	Moves robot with linear motion form the current position for the increment value in a specified frame system.(robots with 7 axes or more)																																										
FORMAT	_declspec(dllexport) short APIENTRY BsclmovEx(short nCid,char *vtype,double spd,char *framename,short toolno,double *p,short axisNum);																																										
ARGUMENTS	<div><div><div>IN (Transfer)</div><div>nCid Communication handler ID number</div><div>*vtype Move speed selection; V : Control point; VR : Position angular</div><div>spd Move speed (0.1 to ****.*mm/s, 0.1 to ***.* ° /s)</div><div>*framename Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1... TOOL : Tool coordinate</div><div>toolno Tool number</div><div>*p Head pointer to the target position storage area</div><div>axisNum Number of axis</div></div><div><div>OUT (Return)</div><div>None</div></div><div><div>Return Value</div><div>0 : Normal completion</div><div>Others : Error codes</div></div></div>																																										
REMARKS	<div><div>Restrictions</div><div>This function is to keep compatibility. Please use BsclmovEx2.</div></div> <div><div>Target Position</div><div>The target position is represented as follows.</div><table><tr><th></th><th>Target position in the specified coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table><div><div>* Set "0" for data P[6] to P[11] if the system has no external axis.</div><div>Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div></div>		Target position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in the specified coordinate system	Robots with 7 axes																																									
P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																									
P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																									
P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																									
P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																									
P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																									
P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																									
P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																									
P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																									
P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																									
P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									

REMARKS	<p>Form The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved</p>
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)
REFERENCE	"BscMovEx2" "BscMovjEx" "BscMovlEx"

■ BscImovEx2

FUNCTION	Moves robot with linear motion from the current position for the increment value in a specified frame system.(robots with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscImovEx(short nCid,short ctype,char *vtype,double spd,char *frameName,short toolNo,double *p,short axisNum);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>ctype Controller selection; 0 : ERC, 1 : MRC, 2 : XRC, 3 : NX100 , 4 : DX100, 5 : FS100</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular</p> <p>spd Move speed (0.1 to ****.*mm/s, 0.1 to ***.* ° /s)</p> <p>*frameName Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1... TOOL : Tool coordinate (Only for FS100/DX100/NX100/XRC/MRC)</p> <p>toolNo Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>axisNum Number of robot axes</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>DX100 and FS100 only can be used robot with 7 axes or more.</p>

REMARKS	<div><div>Target Position</div><div>The target position is represented as follows.</div><table><thead><tr><th></th><th>Target position in the specified coordinate system</th><th>Robots with 7 axes</th></tr></thead><tbody><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></tbody></table><div><div>* Set "0" for data P[6] to P[11] if the system has no external axis.</div><div>Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div></div>		Target position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in the specified coordinate system	Robots with 7 axes																																									
P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																									
P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																									
P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																									
P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																									
P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																									
P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																									
P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																									
P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																									
P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																									
P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>0:Flip,</div><div>0:Elbow Above,</div><div>0:Front Side,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div></div></div></div></div>																																										
	<div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div>																																										
CONTROLLER	FS100, DX100, NX100, XRC, MRC(Serial Port, Ethernet), ERC(Serial Port)																																										
REFERENCE	"BscMovEx2" "BscMovjEx" "BscMovlEx"																																										

■ BscMDSP

FUNCTION	Sends message data.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscMDSP(short nCid,char *ptr);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *ptr Message storage pointer
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Number of message characters The character string for a message is restricted as follows. With FS100/DX100/NX100/XRC/MRC; Character string with 30 characters maximum. With ERC; Character string with 28 characters maximum.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscMov

FUNCTION	Moves robot with specified motion from the current position to a target position in a specified frame system.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscMov(short nCid,char *movtype,char *vtype,double spd,char *framename,short rconf,short toolno,double *p);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*movtype Motion type; MOVJ : Joint; MOVL : Linear; IMOV : Linear (incrementa value)</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular VJ : Joint speed</p> <p>spd Move speed (0.1 to ***.* mm/s, 0.1 to ***.* ° /s, 0.001 to 100.00%)</p> <p>*framename Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1... TOOL : Tool coordinate (Only for NX100/XRC/MRC with motion type "IMOV.")</p> <p>rconf Form</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p>0:Flip, 1:No-Flip</p> <p>0:Elbow Abov, 1: Elbow Under</p> <p>0:Front Side, 1: Back Side</p> <p>0:R<180, 1:R>=180</p> <p>0:T<180, 1:T>=180</p> <p>0:S<180, 1:S>=180</p> <p>Reserved</p> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</p> <p>* With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Target Position</p> <p>The target position data are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in the specified coordinate system</th></tr> </thead> <tbody> <tr> <td>P[0]</td><td>X-axis coordinate (unit: mm)</td></tr> <tr> <td>P[1]</td><td>Y-axis coordinate (unit: mm)</td></tr> <tr> <td>P[2]</td><td>Z-axis coordinate (unit: mm)</td></tr> <tr> <td>P[3]</td><td>Wrist angle Rx (unit: °)</td></tr> <tr> <td>P[4]</td><td>Wrist angle Ry (unit: °)</td></tr> <tr> <td>P[5]</td><td>Wrist angle Rz (unit: °)</td></tr> <tr> <td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[9]</td><td>10th axis pulse number</td></tr> <tr> <td>P[10]</td><td>11th axis pulse number</td></tr> <tr> <td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in the specified coordinate system	P[0]	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in the specified coordinate system																										
P[0]	X-axis coordinate (unit: mm)																										
P[1]	Y-axis coordinate (unit: mm)																										
P[2]	Z-axis coordinate (unit: mm)																										
P[3]	Wrist angle Rx (unit: °)																										
P[4]	Wrist angle Ry (unit: °)																										
P[5]	Wrist angle Rz (unit: °)																										
P[6]	7th axis pulse number (mm for traveling axis)																										
P[7]	8th axis pulse number (mm for traveling axis)																										
P[8]	9th axis pulse number (mm for traveling axis)																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	<p>NX100, XRC, MRC (Serial Port, Ethernet), ERC (Serial Port)</p> <p>* Refer to the BscMovEx2 for the FS100/DX100.</p>																										
REFERENCE	<p>"BscMovEx" "BscMovI" "BscImov" "BscMovEx2"</p>																										

■ BscMovEx

FUNCTION	Moves robot with specified motion from the current position to a target position in a specified frame system.(robots with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscMovEx(short nCid,char *movtype,char *vtype,double spd,char *framename,long rconf,short toolno,double *p,short axisNum);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*movtype Motion type; MOVJ : Joint; MOVL : Linear; IMOV : Linear (incrementa value)</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular VJ : Joint speed</p> <p>spd Move speed (0.1 to ***.* mm/s, 0.1 to ***.* ° /s, 0.01 to 100.00 %)</p> <p>*framename Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1... TOOL : Tool coordinate (Only motion type "IMOV.")</p> <p>rconf Form</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>axisNum Number of axis</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is to keep compatibility. Please use BscMovEx2.</p> <p>Form</p> <p>The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p> 0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved </p>

REMARKS	<div><div><div><div><div><div></div><div>Target Position</div></div></div><div>The target position data are represented as follows.</div></div></div><table><tr><th></th><th>Target position in the specified coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table><div>*Set "0" for data P[6] to P[11] if the system has no external axis. Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div>		Target position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in the specified coordinate system	Robots with 7 axes																																									
P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																									
P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																									
P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																									
P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																									
P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																									
P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																									
P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																									
P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																									
P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																									
P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)																																										
REFERENCE	"BscMovjEx" "BscMovlEx" "BscImovEx"																																										

■ BscMovEx2

FUNCTION	Moves robot with specified motion from the current position to a target position in a specified frame system.(robots with 7 axes or more)
FORMAT	_declspec(dllexport) short APIENTRY BscMovEx(short nCid,short ctype,char *movtype,char *vtype,double spd,char *frameName,long rconf,short toolNo,double *p,short axisNum);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>ctype Controller selection; 0 : ERC, 1 : MRC, 2 : XRC, 3 : NX100 , 4 : DX100, 5 : FS100</p> <p>*movtype Motion type; MOVJ : Joint; MOVL : Linear; IMOV : Linear (incremental value)</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular VJ : Joint speed</p> <p>spd Move speed (0.1 to ***.* mm/s, 0.1 to ***.* ° /s, 0.01 to 100.00 %)</p> <p>*frameName Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1... TOOL : Tool coordinate (Only for FS100/DX100/NX100/XRC /MRC with motion type "IMOV.")</p> <p>rconf Form</p> <p>toolNo Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>axisNum Number of robot axes</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>DX100 and FS100 only can be used robot with 7 axes or more.</p>

REMARKS	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div><div>0:Flip,</div><div>0:Elbow Abov,</div><div>0:Front Side,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div></div><div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div></div></div></div> <div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div>																																										
	<div><div>Target Position</div><div>The target position data are represented as follows.</div><div><table><tr><th></th><th>Target position in the specified coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table></div><div><div>*Set "0" for data P[6] to P[11] if the system has no external axis.</div><div>Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div></div>		Target position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in the specified coordinate system	Robots with 7 axes																																									
P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																									
P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																									
P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																									
P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																									
P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																									
P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																									
P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																									
P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																									
P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																									
P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100, NX100, XRC, MRC(Serial Port, Ethernet), ERC(Serial Port)																																										
REFERENCE	"BscMovjEx" "BscMovlEx" "BscImovEx2"																																										

- BscMovj

FUNCTION	Moves robot with joint motion to a target position in a specified frame system.
FORMAT	_declspec(dlllexport) short APIENTRY BscMovj(short nCid,double spd,char *framename,short rconf,short toolno,double *p);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number spd Move speed (0.01 to 100.0%) *framename Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1...</p> <p>rconf Form toolno Tool number *p Head pointer to the target position storage area</p> <hr/> <p>OUT (Return) None</p> <hr/> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>From</p> <p>The form data are represented by bit data in decimals.</p> <div style="text-align: center;"> D7 D6 D5 D4 D3 D2 D1 D0 0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved </div> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded. * With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Target Position</p> <p>The target positions are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in the specified coordinate system</th></tr> </thead> <tbody> <tr> <td>P[0]</td><td>X-axis coordinate (unit: mm)</td></tr> <tr> <td>P[1]</td><td>Y-axis coordinate (unit: mm)</td></tr> <tr> <td>P[2]</td><td>Z-axis coordinate (unit: mm)</td></tr> <tr> <td>P[3]</td><td>Wrist angle Rx (unit: °)</td></tr> <tr> <td>P[4]</td><td>Wrist angle Ry (unit: °)</td></tr> <tr> <td>P[5]</td><td>Wrist angle Rz (unit: °)</td></tr> <tr> <td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[9]</td><td>10th axis pulse number</td></tr> <tr> <td>P[10]</td><td>11th axis pulse number</td></tr> <tr> <td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in the specified coordinate system	P[0]	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in the specified coordinate system																										
P[0]	X-axis coordinate (unit: mm)																										
P[1]	Y-axis coordinate (unit: mm)																										
P[2]	Z-axis coordinate (unit: mm)																										
P[3]	Wrist angle Rx (unit: °)																										
P[4]	Wrist angle Ry (unit: °)																										
P[5]	Wrist angle Rz (unit: °)																										
P[6]	7th axis pulse number (mm for traveling axis)																										
P[7]	8th axis pulse number (mm for traveling axis)																										
P[8]	9th axis pulse number (mm for traveling axis)																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	<p>FS100,DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)</p> <p>* Refer to the BscMovjEx for the FS100/DX100 with 7 axes or more.</p>																										
REFERENCE	"BscMov" "BscMovl" "BscImov"																										

■ BscMovjEx

FUNCTION	Moves robot with joint motion to a target position in a specified frame system. (robots with 7 axes or more)
FORMAT	_declspec(dllexport) short APIENTRY BscMovjEx(short nCid,double spd,char *frameName,long rconf,short toolNo,double *p,short axisNum);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>spd Move speed (0.01 to 100.0%)</p> <p>*frameName Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1...</p> <p>rconf Form</p> <p>toolNo Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>axisNum Number of axis</p> <hr/> <p>OUT (Return)</p> <p>None</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
REMARKS	<p>From</p> <p>The form data are represented by bit data in decimals.</p> <div style="text-align: center;"> D7 D6 D5 D4 D3 D2 D1 D0 <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> 0: Flip, 0: Elbow Above, 0: Front Side, 0: R<180, 0: T<180, 0: S<180, Reserved </div> <div style="width: 45%; text-align: right;"> 1: No-Flip 1: Elbow Under 1: Back Side 1: R>=180 1: T>=180 1: S>=180 </div> </div> </div>

REMARKS	<div><div><div><div><div><div></div><div>Target position in the specified coordinate system</div><div>Robots with 7 axes</div></div><div><div>P[0]</div><div>X-axis coordinate (unit: mm)</div><div>X-axis coordinate (unit: mm)</div></div><div><div>P[1]</div><div>Y-axis coordinate (unit: mm)</div><div>Y-axis coordinate (unit: mm)</div></div><div><div>P[2]</div><div>Z-axis coordinate (unit: mm)</div><div>Z-axis coordinate (unit: mm)</div></div><div><div>P[3]</div><div>Wrist angle Rx (unit: °)</div><div>Wrist angle Rx (unit: °)</div></div><div><div>P[4]</div><div>Wrist angle Ry (unit: °)</div><div>Wrist angle Ry (unit: °)</div></div><div><div>P[5]</div><div>Wrist angle Rz (unit: °)</div><div>Wrist angle Rz (unit: °)</div></div><div><div>P[6]</div><div>7th axis pulse number (mm for traveling axis)</div><div>Re (unit: °)</div></div><div><div>P[7]</div><div>8th axis pulse number (mm for traveling axis)</div><div>8th axis pulse number (mm for traveling axis)</div></div><div><div>P[8]</div><div>9th axis pulse number (mm for traveling axis)</div><div>9th axis pulse number (mm for traveling axis)</div></div><div><div>P[9]</div><div>10th axis pulse number</div><div>10th axis pulse number (mm for traveling axis)</div></div><div><div>P[10]</div><div>11th axis pulse number</div><div>11th axis pulse number</div></div><div><div>P[11]</div><div>12th axis pulse number</div><div>12th axis pulse number</div></div><div><div>P[12]</div><div>-</div><div>13th axis pulse number</div></div></div></div><div><div>*Set "0" for data P[6] to P[11] if the system has no external axis. Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div></div></div>
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)
REFERENCE	"BscMovEx2" "BscMovlEx" "BscImovEx2"

- BscMovl

FUNCTION	Moves robot with linear motion to a target position in a specified frame system.
FORMAT	_declspec(dllexport) short APIENTRY BscMovl(short nCid,char *vtype,double spd,char *frameName,short rconf,short toolno,double *p);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular</p> <p>spd Move speed (0.1 to ****.* mm/s, 0.1 to ***.* ° /s)</p> <p>*frameName Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1...</p> <p>rconf Form</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <hr/> <p>OUT (Return)</p> <p>None</p> <hr/> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>
REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <div style="text-align: center;"> D7 D6 D5 D4 D3 D2 D1 D0 </div> <div style="margin-left: 400px;"> 0:Flip, 1:No-Flip 0:Elbow Above, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved </div> <p>* With the ERC or ERC2, the data from D3 to D7 are disregarded. * With the MRC or MRC2, the data D5 and D7 are disregarded.</p>

REMARKS	<p>Target Position The target positions are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in the specified coordinate system</th></tr> </thead> <tbody> <tr> <td>P[0]</td><td>X-axis coordinate (unit: mm)</td></tr> <tr> <td>P[1]</td><td>Y-axis coordinate (unit: mm)</td></tr> <tr> <td>P[2]</td><td>Z-axis coordinate (unit: mm)</td></tr> <tr> <td>P[3]</td><td>Wrist angle Rx (unit: °)</td></tr> <tr> <td>P[4]</td><td>Wrist angle Ry (unit: °)</td></tr> <tr> <td>P[5]</td><td>Wrist angle Rz (unit: °)</td></tr> <tr> <td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td></tr> <tr> <td>P[9]</td><td>10th axis pulse number</td></tr> <tr> <td>P[10]</td><td>11th axis pulse number</td></tr> <tr> <td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in the specified coordinate system	P[0]	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	P[7]	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in the specified coordinate system																										
P[0]	X-axis coordinate (unit: mm)																										
P[1]	Y-axis coordinate (unit: mm)																										
P[2]	Z-axis coordinate (unit: mm)																										
P[3]	Wrist angle Rx (unit: °)																										
P[4]	Wrist angle Ry (unit: °)																										
P[5]	Wrist angle Rz (unit: °)																										
P[6]	7th axis pulse number (mm for traveling axis)																										
P[7]	8th axis pulse number (mm for traveling axis)																										
P[8]	9th axis pulse number (mm for traveling axis)																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	<p>FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)</p> <p>* Refer to the BscMovIEx for the FS100/DX100 with 7 axes or more.</p>																										
REFERENCE	"BscMov" "BscMovEx" "BscImov"																										

■ BscMovlEx

FUNCTION	Moves robot with linear motion to a target position in a specified frame system. (robots with 7 axes or more)
FORMAT	_declspec(dllexport) short APIENTRY BscMovlEx(short nCid,char *vtype,double spd,char *frameName,long rconf,short toolno,double *p,short axisNum);
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular</p> <p>spd Move speed (0.1 to ****.* mm/s, 0.1 to ***.* ° /s)</p> <p>*frameName Coordinate name; BASE : Base coordinate; ROBOT : Robot coordinate; UF1 : User coordinate1...</p> <p>rconf Form</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>axisNum Number of axis</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>
REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <div style="text-align: center;"> D7 D6 D5 D4 D3 D2 D1 D0 0:Flip, 1:No-Flip 0:Elbow Above, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved </div>

REMARKS	Target Position The target positions are represented as follows.																																										
	<table><tr><th></th><th>Target position in the specified coordinate system</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>X-axis coordinate (unit: mm)</td><td>X-axis coordinate (unit: mm)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm)</td><td>Y-axis coordinate (unit: mm)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm)</td><td>Z-axis coordinate (unit: mm)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit: °)</td><td>Wrist angle Rx (unit: °)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit: °)</td><td>Wrist angle Ry (unit: °)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit: °)</td><td>Wrist angle Rz (unit: °)</td></tr><tr><td>P[6]</td><td>7th axis pulse number (mm for traveling axis)</td><td>Re (unit: °)</td></tr><tr><td>P[7]</td><td>8th axis pulse number (mm for traveling axis)</td><td>8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[8]</td><td>9th axis pulse number (mm for traveling axis)</td><td>9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table>		Target position in the specified coordinate system	Robots with 7 axes	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
		Target position in the specified coordinate system	Robots with 7 axes																																								
	P[0]	X-axis coordinate (unit: mm)	X-axis coordinate (unit: mm)																																								
	P[1]	Y-axis coordinate (unit: mm)	Y-axis coordinate (unit: mm)																																								
	P[2]	Z-axis coordinate (unit: mm)	Z-axis coordinate (unit: mm)																																								
	P[3]	Wrist angle Rx (unit: °)	Wrist angle Rx (unit: °)																																								
	P[4]	Wrist angle Ry (unit: °)	Wrist angle Ry (unit: °)																																								
	P[5]	Wrist angle Rz (unit: °)	Wrist angle Rz (unit: °)																																								
	P[6]	7th axis pulse number (mm for traveling axis)	Re (unit: °)																																								
	P[7]	8th axis pulse number (mm for traveling axis)	8th axis pulse number (mm for traveling axis)																																								
	P[8]	9th axis pulse number (mm for traveling axis)	9th axis pulse number (mm for traveling axis)																																								
	P[9]	10th axis pulse number	10th axis pulse number (mm for traveling axis)																																								
	P[10]	11th axis pulse number	11th axis pulse number																																								
	P[11]	12th axis pulse number	12th axis pulse number																																								
P[12]	-	13th axis pulse number																																									
*Set "0" for data P[6] to P[11] if the system has no external axis. Set "0" for data P[7] to P[12] if the system has robots with 7 axes.																																											
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)																																										
REFERENCE	"BscMovEx2" "BscMovjEx" "BscImovEx2"																																										

■ BscOPLock

FUNCTION	Interlocks the robot.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscOPLock(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Interlock Status Once interlock is set, all but the following are interlocked.
	With FS100/DX100/NX100 * Emergency stop, hold from the programming pendant. * Input signals except I/O mode change, external start, external servo ON, cycle change, I/O prohibited, P.P/PANEL prohibited, and master call Interlock cannot be accomplished when the programming pendant is in the editing mode, or when file access is operating by other functions.
	With XRC * Hold from the programming pendant. * Hold, emergency stop from the playback box. * Input signal other than I/O 404x, 405x and 409x. (including external hold, external servo OFF) Interlock cannot be accomplished when the programming pendant is in the editing mode, or when file access is operating by other functions.
	With MRC * Hold from the programming pendant. * Hold, emergency stop from the playback box. * Input signal other than I/O 404x and 405x. (including external hold, external servo OFF) Interlock cannot be accomplished when the programming pendant is in the editing mode, or when file access is operating by other functions.
	With ERC * Start and hold buttons of panel operation. * Emergency stop button of panel operation. * Servo power ON button of panel operation.

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscOPUnLock"

■ BscOPUnLock

FUNCTION	Releases the robot interlocked status.
FORMAT	_declspec(dllexport) short APIENTRY BscOPUnLock(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscOPLock"

■ BscPMov

FUNCTION	Moves robot to a specified pulse position.																										
FORMAT	_declspec(dllexport) short APIENTRY BscPMov(short nCid,char *movtype,char *vtype,double spd,short toolno,double *p);																										
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*movvtype Motion type; MOVJ : Joint; MOVL : Linear</p> <p>*vtype Move speed selection; V : Control point; VR : Position angular VJ : Joint speed</p> <p>spd Move speed (0.1 to ****.* mm/s, 0.1 to ***.* ° /s, 0.01 to 100.00%)</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																										
REMARKS	<p>Target Position</p> <p>The target position data are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in units of pulses</th></tr> </thead> <tbody> <tr><td>P[0]</td><td>S-axis pulse number</td></tr> <tr><td>P[1]</td><td>L-axis pulse number</td></tr> <tr><td>P[2]</td><td>U-axis pulse number</td></tr> <tr><td>P[3]</td><td>R-axis pulse number</td></tr> <tr><td>P[4]</td><td>B-axis pulse number</td></tr> <tr><td>P[5]</td><td>T-axis pulse number</td></tr> <tr><td>P[6]</td><td>7th axis pulse number</td></tr> <tr><td>P[7]</td><td>8th axis pulse number</td></tr> <tr><td>P[8]</td><td>9th axis pulse number</td></tr> <tr><td>P[9]</td><td>10th axis pulse number</td></tr> <tr><td>P[10]</td><td>11th axis pulse number</td></tr> <tr><td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in units of pulses	P[0]	S-axis pulse number	P[1]	L-axis pulse number	P[2]	U-axis pulse number	P[3]	R-axis pulse number	P[4]	B-axis pulse number	P[5]	T-axis pulse number	P[6]	7th axis pulse number	P[7]	8th axis pulse number	P[8]	9th axis pulse number	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in units of pulses																										
P[0]	S-axis pulse number																										
P[1]	L-axis pulse number																										
P[2]	U-axis pulse number																										
P[3]	R-axis pulse number																										
P[4]	B-axis pulse number																										
P[5]	T-axis pulse number																										
P[6]	7th axis pulse number																										
P[7]	8th axis pulse number																										
P[8]	9th axis pulse number																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port) * Refer to the BscPMovEx for the FS100/DX100 with 7 axes or more.																										
REFERENCE	"BscPMovj" "BscPMovi"																										

■ BscPMovEx

FUNCTION	Moves robot to a specified pulse position. (robots with 7 axes or more)																																										
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPMovEx(short nCid,char *movtype,char *vtype,double spd,short toolno,double *p,short axisNum);</code>																																										
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *movvtype Motion type; MOVJ : Joint, MOVL : Linear *vtype Move speed selection; V : TCP speed VR : Play speed of the posture VJ : Joint speed spd Move speed (0.1 to ****.* mm/s, 0.1 to ***.* °/s, 0.01 to 100.00%) toolno Tool number *p Head pointer to the target position storage area axisNum Number of axis</div><div>OUT (Return) None</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>																																										
REMARKS	<div><div>Target Position The target position data are represented as follows.</div><table><tr><th></th><th>Target position in units of pulses</th><th>Robots with 7 axes</th></tr><tr><td>P[0]</td><td>S-axis pulse number</td><td>S-axis pulse number</td></tr><tr><td>P[1]</td><td>L-axis pulse number</td><td>L-axis pulse number</td></tr><tr><td>P[2]</td><td>U-axis pulse number</td><td>U-axis pulse number</td></tr><tr><td>P[3]</td><td>R-axis pulse number</td><td>R-axis pulse number</td></tr><tr><td>P[4]</td><td>B-axis pulse number</td><td>B-axis pulse number</td></tr><tr><td>P[5]</td><td>T-axis pulse number</td><td>T-axis pulse number</td></tr><tr><td>P[6]</td><td>7th axis pulse number</td><td>E-axis pulse number</td></tr><tr><td>P[7]</td><td>8th axis pulse number</td><td>8th axis pulse number</td></tr><tr><td>P[8]</td><td>9th axis pulse number</td><td>9th axis pulse number</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></table><div>*Set "0" for data P[6] to P[11] if the system has no external axis. Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div>		Target position in units of pulses	Robots with 7 axes	P[0]	S-axis pulse number	S-axis pulse number	P[1]	L-axis pulse number	L-axis pulse number	P[2]	U-axis pulse number	U-axis pulse number	P[3]	R-axis pulse number	R-axis pulse number	P[4]	B-axis pulse number	B-axis pulse number	P[5]	T-axis pulse number	T-axis pulse number	P[6]	7th axis pulse number	E-axis pulse number	P[7]	8th axis pulse number	8th axis pulse number	P[8]	9th axis pulse number	9th axis pulse number	P[9]	10th axis pulse number	10th axis pulse number	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in units of pulses	Robots with 7 axes																																									
P[0]	S-axis pulse number	S-axis pulse number																																									
P[1]	L-axis pulse number	L-axis pulse number																																									
P[2]	U-axis pulse number	U-axis pulse number																																									
P[3]	R-axis pulse number	R-axis pulse number																																									
P[4]	B-axis pulse number	B-axis pulse number																																									
P[5]	T-axis pulse number	T-axis pulse number																																									
P[6]	7th axis pulse number	E-axis pulse number																																									
P[7]	8th axis pulse number	8th axis pulse number																																									
P[8]	9th axis pulse number	9th axis pulse number																																									
P[9]	10th axis pulse number	10th axis pulse number																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)																																										
REFERENCE	"BscPMovjEx" "BscPMovlEx"																																										

■ BscPMovj

FUNCTION	Moves robot to a specified pulse position with joint motion.																										
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPMovj(short nCid,double spd,short toolno,double *p);</code>																										
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>spd Move speed (0.01 to 100.00%)</p> <p>toolno Tool number</p> <p>*p Head pointer to the target position storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																										
REMARKS	<p>Target Position</p> <p>The target position data are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in units of pulses</th></tr> </thead> <tbody> <tr><td>P[0]</td><td>S-axis pulse number</td></tr> <tr><td>P[1]</td><td>L-axis pulse number</td></tr> <tr><td>P[2]</td><td>U-axis pulse number</td></tr> <tr><td>P[3]</td><td>R-axis pulse number</td></tr> <tr><td>P[4]</td><td>B-axis pulse number</td></tr> <tr><td>P[5]</td><td>T-axis pulse number</td></tr> <tr><td>P[6]</td><td>7th axis pulse number</td></tr> <tr><td>P[7]</td><td>8th axis pulse number</td></tr> <tr><td>P[8]</td><td>9th axis pulse number</td></tr> <tr><td>P[9]</td><td>10th axis pulse number</td></tr> <tr><td>P[10]</td><td>11th axis pulse number</td></tr> <tr><td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in units of pulses	P[0]	S-axis pulse number	P[1]	L-axis pulse number	P[2]	U-axis pulse number	P[3]	R-axis pulse number	P[4]	B-axis pulse number	P[5]	T-axis pulse number	P[6]	7th axis pulse number	P[7]	8th axis pulse number	P[8]	9th axis pulse number	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in units of pulses																										
P[0]	S-axis pulse number																										
P[1]	L-axis pulse number																										
P[2]	U-axis pulse number																										
P[3]	R-axis pulse number																										
P[4]	B-axis pulse number																										
P[5]	T-axis pulse number																										
P[6]	7th axis pulse number																										
P[7]	8th axis pulse number																										
P[8]	9th axis pulse number																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	<p>FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)</p> <p>ERC (Serial Port)</p> <p>* Refer to the BscPMovjEx for the FS100/DX100 with 7 axes or more.</p>																										
REFERENCE	"BscPMovl" "BscPMov"																										

■ BscPMovjEx

FUNCTION	Moves robot to a specified pulse position with joint motion. (robots with 7 axes or more)																																										
FORMAT	_declspec(dllexport) short APIENTRY BscPMovjEx(short nCid,double spd,short toolno,double *p,short axisNum);																																										
ARGUMENTS	<div><div>IN (Transfer)</div><div>nCid Communication handler ID number spd Move speed (0.01 to 100.00%) toolno Tool number *p Head pointer to the target position storage area axisNum Number of axis</div></div> <div><div>OUT (Return)</div><div>None</div></div> <div><div>Return Value</div><div>0 : Normal completion Others : Error codes</div></div>																																										
REMARKS	<div><div>Target Position</div><div>The target position data are represented as follows.</div><table><thead><tr><th></th><th>Target position in units of pulses</th><th>Robots with 7 axes</th></tr></thead><tbody><tr><td>P[0]</td><td>S-axis pulse number</td><td>S-axis pulse number</td></tr><tr><td>P[1]</td><td>L-axis pulse number</td><td>L-axis pulse number</td></tr><tr><td>P[2]</td><td>U-axis pulse number</td><td>U-axis pulse number</td></tr><tr><td>P[3]</td><td>R-axis pulse number</td><td>R-axis pulse number</td></tr><tr><td>P[4]</td><td>B-axis pulse number</td><td>B-axis pulse number</td></tr><tr><td>P[5]</td><td>T-axis pulse number</td><td>T-axis pulse number</td></tr><tr><td>P[6]</td><td>7th axis pulse number</td><td>E-axis pulse number</td></tr><tr><td>P[7]</td><td>8th axis pulse number</td><td>8th axis pulse number</td></tr><tr><td>P[8]</td><td>9th axis pulse number</td><td>9th axis pulse number</td></tr><tr><td>P[9]</td><td>10th axis pulse number</td><td>10th axis pulse number</td></tr><tr><td>P[10]</td><td>11th axis pulse number</td><td>11th axis pulse number</td></tr><tr><td>P[11]</td><td>12th axis pulse number</td><td>12th axis pulse number</td></tr><tr><td>P[12]</td><td>-</td><td>13th axis pulse number</td></tr></tbody></table><div>*Set "0" for data P[6] to P[11] if the system has no external axis. Set "0" for data P[7] to P[12] if the system has robots with 7 axes.</div></div>		Target position in units of pulses	Robots with 7 axes	P[0]	S-axis pulse number	S-axis pulse number	P[1]	L-axis pulse number	L-axis pulse number	P[2]	U-axis pulse number	U-axis pulse number	P[3]	R-axis pulse number	R-axis pulse number	P[4]	B-axis pulse number	B-axis pulse number	P[5]	T-axis pulse number	T-axis pulse number	P[6]	7th axis pulse number	E-axis pulse number	P[7]	8th axis pulse number	8th axis pulse number	P[8]	9th axis pulse number	9th axis pulse number	P[9]	10th axis pulse number	10th axis pulse number	P[10]	11th axis pulse number	11th axis pulse number	P[11]	12th axis pulse number	12th axis pulse number	P[12]	-	13th axis pulse number
	Target position in units of pulses	Robots with 7 axes																																									
P[0]	S-axis pulse number	S-axis pulse number																																									
P[1]	L-axis pulse number	L-axis pulse number																																									
P[2]	U-axis pulse number	U-axis pulse number																																									
P[3]	R-axis pulse number	R-axis pulse number																																									
P[4]	B-axis pulse number	B-axis pulse number																																									
P[5]	T-axis pulse number	T-axis pulse number																																									
P[6]	7th axis pulse number	E-axis pulse number																																									
P[7]	8th axis pulse number	8th axis pulse number																																									
P[8]	9th axis pulse number	9th axis pulse number																																									
P[9]	10th axis pulse number	10th axis pulse number																																									
P[10]	11th axis pulse number	11th axis pulse number																																									
P[11]	12th axis pulse number	12th axis pulse number																																									
P[12]	-	13th axis pulse number																																									
CONTROLLER	FS100, DX100 (Serial Port, Ethernet)																																										
REFERENCE	"BscPMovIEx" "BscPMovEx"																																										

■ BscPMovl

FUNCTION	Moves robot to a specified pulse position with linear motion.																										
FORMAT	_declspec(dllexport) short APIENTRY BscPMovl(short nCid,char *vtype,double spd,short toolno,double *p);																										
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number *vtype Move speed selection; V : Control point; VR : Position angular spd Move speed (0.1 to ****.*mm/s, 0.1 to ***.* ° /s) toolno Tool number *p Head pointer to the target position storage area</p> <hr/> <p>OUT (Return)</p> <p>None</p> <hr/> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>																										
REMARKS	<p>Target Position</p> <p>The target position data are represented as follows.</p> <table border="1"> <thead> <tr> <th></th><th>Target position in units of pulses</th></tr> </thead> <tbody> <tr><td>P[0]</td><td>S-axis pulse number</td></tr> <tr><td>P[1]</td><td>L-axis pulse number</td></tr> <tr><td>P[2]</td><td>U-axis pulse number</td></tr> <tr><td>P[3]</td><td>R-axis pulse number</td></tr> <tr><td>P[4]</td><td>B-axis pulse number</td></tr> <tr><td>P[5]</td><td>T-axis pulse number</td></tr> <tr><td>P[6]</td><td>7th axis pulse number</td></tr> <tr><td>P[7]</td><td>8th axis pulse number</td></tr> <tr><td>P[8]</td><td>9th axis pulse number</td></tr> <tr><td>P[9]</td><td>10th axis pulse number</td></tr> <tr><td>P[10]</td><td>11th axis pulse number</td></tr> <tr><td>P[11]</td><td>12th axis pulse number</td></tr> </tbody> </table> <p>*Set "0" for data P[7] to P[11] if the system has no external axis.</p>		Target position in units of pulses	P[0]	S-axis pulse number	P[1]	L-axis pulse number	P[2]	U-axis pulse number	P[3]	R-axis pulse number	P[4]	B-axis pulse number	P[5]	T-axis pulse number	P[6]	7th axis pulse number	P[7]	8th axis pulse number	P[8]	9th axis pulse number	P[9]	10th axis pulse number	P[10]	11th axis pulse number	P[11]	12th axis pulse number
	Target position in units of pulses																										
P[0]	S-axis pulse number																										
P[1]	L-axis pulse number																										
P[2]	U-axis pulse number																										
P[3]	R-axis pulse number																										
P[4]	B-axis pulse number																										
P[5]	T-axis pulse number																										
P[6]	7th axis pulse number																										
P[7]	8th axis pulse number																										
P[8]	9th axis pulse number																										
P[9]	10th axis pulse number																										
P[10]	11th axis pulse number																										
P[11]	12th axis pulse number																										
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port) * Refer to the BscPMovIEx for the FS100/DX100 with 7 axes or more.																										
REFERENCE	"BscPMovj" "BscPMov"																										

- BscPMovlEx

FUNCTION	Moves robot to a specified pulse position with linear motion.
FORMAT	_declspec(dllexport) short APIENTRY BscPMovlEx(short nCid,char *vtype,double spd,short toolno,double *p,short axisNum);
ARGUMENTS	<div><div>IN (Transfer)</div><div>nCid Communication handler ID number *vtype Move speed selection; V : Control point; </div></div>

■ BscPutUFrame

FUNCTION	Sets a specified user frame data.																		
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPutUFrame(short nCid,char *ufname,double *p);</code>																		
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*ufname Storage pointer of the user coordinate name to be written in.</p> <p>*p Head pointer to the user coordinate data storage area</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																		
REMARKS	<p>Restrictions</p> <p>This function does not support the robot with 7 axes or more.</p> <p>User Coordinate Name</p> <p>The following coordinate names correspond to the user coordinate numbers.</p> <table border="1"> <thead> <tr> <th>User Coordinate Name</th><th>Specified Name</th></tr> </thead> <tbody> <tr><td>User coordinate 1</td><td>UF1</td></tr> <tr><td>User coordinate 2</td><td>UF2</td></tr> <tr><td>User coordinate 3</td><td>UF3</td></tr> <tr><td>⋮</td><td>⋮</td></tr> <tr><td>User coordinate 21</td><td>UF21</td></tr> <tr><td>User coordinate 22</td><td>UF22</td></tr> <tr><td>User coordinate 23</td><td>UF23</td></tr> <tr><td>User coordinate 24</td><td>UF24</td></tr> </tbody> </table> <p>* User coordinate numbers 9 to 24 are effective only for NX100/XRC/MRC.</p>	User Coordinate Name	Specified Name	User coordinate 1	UF1	User coordinate 2	UF2	User coordinate 3	UF3	⋮	⋮	User coordinate 21	UF21	User coordinate 22	UF22	User coordinate 23	UF23	User coordinate 24	UF24
User Coordinate Name	Specified Name																		
User coordinate 1	UF1																		
User coordinate 2	UF2																		
User coordinate 3	UF3																		
⋮	⋮																		
User coordinate 21	UF21																		
User coordinate 22	UF22																		
User coordinate 23	UF23																		
User coordinate 24	UF24																		

REMARKS	<div><div>Variable Types</div><div>Coordinate values of the user coordinate system specified with the user coordinate number are assigned to the user coordinate data as follows.</div><table><thead><tr><th>Variables</th><th>Coordinate System</th><th>Meaning</th></tr></thead><tbody><tr><td>P[0]</td><td rowspan="7">ORG</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[6]</td><td>Form</td></tr><tr><td>P[7]</td><td rowspan="7">XX</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[8]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[9]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[10]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[11]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[12]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[13]</td><td>Form</td></tr><tr><td>P[14]</td><td rowspan="7">XY</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[15]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[16]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[17]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[18]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[19]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places)</td></tr><tr><td>P[20]</td><td>Form</td></tr><tr><td>P[21]</td><td colspan="2">Tool number</td></tr><tr><td>P[22]</td><td colspan="2">7th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[23]</td><td colspan="2">8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[24]</td><td colspan="2">9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[25]</td><td colspan="2">10th axis pulse number</td></tr><tr><td>P[26]</td><td colspan="2">11th axis pulse number</td></tr><tr><td>P[27]</td><td colspan="2">12th axis pulse number</td></tr></tbody></table></div>	Variables	Coordinate System	Meaning	P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[1]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[2]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[3]	Wrist angle Rx (unit:°, effective down to 2 decimal places)	P[4]	Wrist angle Ry (unit:°, effective down to 2 decimal places)	P[5]	Wrist angle Rz (unit:°, effective down to 2 decimal places)	P[6]	Form	P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[8]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[9]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[10]	Wrist angle Rx (unit:°, effective down to 2 decimal places)	P[11]	Wrist angle Ry (unit:°, effective down to 2 decimal places)	P[12]	Wrist angle Rz (unit:°, effective down to 2 decimal places)	P[13]	Form	P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[15]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[16]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[17]	Wrist angle Rx (unit:°, effective down to 2 decimal places)	P[18]	Wrist angle Ry (unit:°, effective down to 2 decimal places)	P[19]	Wrist angle Rz (unit:°, effective down to 2 decimal places)	P[20]	Form	P[21]	Tool number		P[22]	7th axis pulse number (mm for traveling axis)		P[23]	8th axis pulse number (mm for traveling axis)		P[24]	9th axis pulse number (mm for traveling axis)		P[25]	10th axis pulse number		P[26]	11th axis pulse number		P[27]	12th axis pulse number	
Variables	Coordinate System	Meaning																																																																				
P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[1]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[2]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[3]		Wrist angle Rx (unit:°, effective down to 2 decimal places)																																																																				
P[4]		Wrist angle Ry (unit:°, effective down to 2 decimal places)																																																																				
P[5]		Wrist angle Rz (unit:°, effective down to 2 decimal places)																																																																				
P[6]		Form																																																																				
P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[8]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[9]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[10]		Wrist angle Rx (unit:°, effective down to 2 decimal places)																																																																				
P[11]		Wrist angle Ry (unit:°, effective down to 2 decimal places)																																																																				
P[12]		Wrist angle Rz (unit:°, effective down to 2 decimal places)																																																																				
P[13]		Form																																																																				
P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[15]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[16]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[17]		Wrist angle Rx (unit:°, effective down to 2 decimal places)																																																																				
P[18]		Wrist angle Ry (unit:°, effective down to 2 decimal places)																																																																				
P[19]		Wrist angle Rz (unit:°, effective down to 2 decimal places)																																																																				
P[20]		Form																																																																				
P[21]	Tool number																																																																					
P[22]	7th axis pulse number (mm for traveling axis)																																																																					
P[23]	8th axis pulse number (mm for traveling axis)																																																																					
P[24]	9th axis pulse number (mm for traveling axis)																																																																					
P[25]	10th axis pulse number																																																																					
P[26]	11th axis pulse number																																																																					
P[27]	12th axis pulse number																																																																					
	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div><div>0:Flip,</div><div>0:Elbow Abov,</div><div>0:Front Side,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div></div></div><div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div></div>																																																																					
CONTROLLER	NX100, XRC, MRC (Serial Port, Ethernet), ERC (Serial Port)																																																																					
REFERENCE	"BscGetUFrame" "BscPutUFrameEx2"																																																																					

■ BscPutUFrameEx2

FUNCTION	Sets a specified user frame data.																				
FORMAT	_declspec(dllexport) short APIENTRY BscPutUFrame(short nCid,short ctype,char *ufname,double *p);																				
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>ctype Controller selection; 0 : ERC, 1 : MRC, 2 : XRC, 3 : NX100 , 4 : DX100, 5 : FS100</p> <p>*ufname Storage pointer of the user coordinate name to be written in.</p> <p>*p Head pointer to the user coordinate data storage area</p> <hr/> <p>OUT (Return)</p> <p>None</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>																				
REMARKS	<p>Restrictions</p> <p>This function does not support the robot with 7 axes or more.</p>																				
	<p>User Coordinate Name</p> <p>The following coordinate names correspond to the user coordinate numbers.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>User Coordinate Name</th><th>Specified Name</th></tr> </thead> <tbody> <tr> <td>User coordinate 1</td><td>UF1</td></tr> <tr> <td>User coordinate 2</td><td>UF2</td></tr> <tr> <td>User coordinate 3</td><td>UF3</td></tr> <tr> <td>:</td><td>:</td></tr> <tr> <td>:</td><td>:</td></tr> <tr> <td>User coordinate 61</td><td>UF61</td></tr> <tr> <td>User coordinate 62</td><td>UF62</td></tr> <tr> <td>User coordinate 63</td><td>UF63</td></tr> <tr> <td>User coordinate 64</td><td>UF64</td></tr> </tbody> </table> <p>* User coordinate numbers 9 to 16 are effective only for FS100. * User coordinate numbers 9 to 64 are effective only for DX100. * User coordinate numbers 9 to 24 are effective only for NX100/XRC/MRC.</p>	User Coordinate Name	Specified Name	User coordinate 1	UF1	User coordinate 2	UF2	User coordinate 3	UF3	:	:	:	:	User coordinate 61	UF61	User coordinate 62	UF62	User coordinate 63	UF63	User coordinate 64	UF64
User Coordinate Name	Specified Name																				
User coordinate 1	UF1																				
User coordinate 2	UF2																				
User coordinate 3	UF3																				
:	:																				
:	:																				
User coordinate 61	UF61																				
User coordinate 62	UF62																				
User coordinate 63	UF63																				
User coordinate 64	UF64																				

REMARKS	<div><div>Variable Types</div><div>Coordinate values of the user coordinate system specified with the user coordinate number are assigned to the user coordinate data as follows.</div><table><thead><tr><th>Variables</th><th>Coordinate System</th><th>Meaning</th></tr></thead><tbody><tr><td>P[0]</td><td rowspan="7">ORG</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[1]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[2]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[3]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[4]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[5]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[6]</td><td>Form</td></tr><tr><td>P[7]</td><td rowspan="7">XX</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[8]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[9]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[10]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[11]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[12]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[13]</td><td>Form</td></tr><tr><td>P[14]</td><td rowspan="7">XY</td><td>X-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[15]</td><td>Y-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[16]</td><td>Z-axis coordinate (unit: mm, effective down to 3 decimal places)</td></tr><tr><td>P[17]</td><td>Wrist angle Rx (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[18]</td><td>Wrist angle Ry (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[19]</td><td>Wrist angle Rz (unit:°, effective down to 2 decimal places) *1</td></tr><tr><td>P[20]</td><td>Form</td></tr><tr><td>P[21]</td><td colspan="2">Tool number</td></tr><tr><td>P[22]</td><td colspan="2">7th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[23]</td><td colspan="2">8th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[24]</td><td colspan="2">9th axis pulse number (mm for traveling axis)</td></tr><tr><td>P[25]</td><td colspan="2">10th axis pulse number</td></tr><tr><td>P[26]</td><td colspan="2">11th axis pulse number</td></tr><tr><td>P[27]</td><td colspan="2">12th axis pulse number</td></tr></tbody></table><div><div>*1 DX100, FS100 is effective down to 4 decimal places.</div></div></div>	Variables	Coordinate System	Meaning	P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[1]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[2]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[3]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[4]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[5]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[6]	Form	P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[8]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[9]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[10]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[11]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[12]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[13]	Form	P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)	P[15]	Y-axis coordinate (unit: mm, effective down to 3 decimal places)	P[16]	Z-axis coordinate (unit: mm, effective down to 3 decimal places)	P[17]	Wrist angle Rx (unit:°, effective down to 2 decimal places) *1	P[18]	Wrist angle Ry (unit:°, effective down to 2 decimal places) *1	P[19]	Wrist angle Rz (unit:°, effective down to 2 decimal places) *1	P[20]	Form	P[21]	Tool number		P[22]	7th axis pulse number (mm for traveling axis)		P[23]	8th axis pulse number (mm for traveling axis)		P[24]	9th axis pulse number (mm for traveling axis)		P[25]	10th axis pulse number		P[26]	11th axis pulse number		P[27]	12th axis pulse number	
Variables	Coordinate System	Meaning																																																																				
P[0]	ORG	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[1]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[2]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[3]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[4]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[5]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[6]		Form																																																																				
P[7]	XX	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[8]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[9]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[10]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[11]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[12]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[13]		Form																																																																				
P[14]	XY	X-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[15]		Y-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[16]		Z-axis coordinate (unit: mm, effective down to 3 decimal places)																																																																				
P[17]		Wrist angle Rx (unit:°, effective down to 2 decimal places) *1																																																																				
P[18]		Wrist angle Ry (unit:°, effective down to 2 decimal places) *1																																																																				
P[19]		Wrist angle Rz (unit:°, effective down to 2 decimal places) *1																																																																				
P[20]		Form																																																																				
P[21]	Tool number																																																																					
P[22]	7th axis pulse number (mm for traveling axis)																																																																					
P[23]	8th axis pulse number (mm for traveling axis)																																																																					
P[24]	9th axis pulse number (mm for traveling axis)																																																																					
P[25]	10th axis pulse number																																																																					
P[26]	11th axis pulse number																																																																					
P[27]	12th axis pulse number																																																																					
	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>↑</div><div>↑</div><div>↑</div><div>↑</div><div>↑</div><div>↑</div><div>↑</div><div>↑</div></div><div><div>0: Flip,</div><div>0: Elbow Above,</div><div>0: Front Side,</div><div>0: R<180,</div><div>0: T<180,</div><div>0: S<180,</div><div>Reserved</div><div>1: No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1: R>=180</div><div>1: T>=180</div><div>1: S>=180</div></div></div></div></div>																																																																					
	<div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div>																																																																					
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																																																					
REFERENCE	"BscGetUFrame" "BscPutUFrame"																																																																					

■ BscPutVarData

FUNCTION	Sets variable data.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPutVarData(short nCid,short type,short varno,double *p);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number type Variable Types varno Variable number *p Head pointer to the numeric variable storage area</p> <p>OUT (Return) None</p> <p>Return Value 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <p>Variable Types The variable types are represented as follows. 0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type 4 : Robot axis position type 5 : Base axis position type 6 : Station axis position type (pulse type only)</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	8	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	8	B-axis Pulses	T-axis Pulses	Tool Number	-	-
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See the following for details on the coordinate system types and form.

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

REMARKS	<div>Coordinate Types NX100/XRC/MRC</div> <div>The following coordinate names correspond to the coordinate type data.</div> <table><thead><tr><th>Coordinate type</th><th>Coordinate name</th><th>Coordinate type</th><th>Coordinate name</th></tr></thead><tbody><tr><td>0</td><td>Base coordinate</td><td>14</td><td>User coordinate 13</td></tr><tr><td>1</td><td>Robot coordinate</td><td>15</td><td>User coordinate 14</td></tr><tr><td>2</td><td>User coordinate 1</td><td>16</td><td>User coordinate 15</td></tr><tr><td>3</td><td>User coordinate 2</td><td>17</td><td>User coordinate 16</td></tr><tr><td>4</td><td>User coordinate 3</td><td>18</td><td>User coordinate 17</td></tr><tr><td>5</td><td>User coordinate 4</td><td>19</td><td>User coordinate 18</td></tr><tr><td>6</td><td>User coordinate 5</td><td>20</td><td>User coordinate 19</td></tr><tr><td>7</td><td>User coordinate 6</td><td>21</td><td>User coordinate 20</td></tr><tr><td>8</td><td>User coordinate 7</td><td>22</td><td>User coordinate 21</td></tr><tr><td>9</td><td>User coordinate 8</td><td>23</td><td>User coordinate 22</td></tr><tr><td>10</td><td>User coordinate 9</td><td>24</td><td>User coordinate 23</td></tr><tr><td>11</td><td>User coordinate 10</td><td>25</td><td>User coordinate 24</td></tr><tr><td>12</td><td>User coordinate 11</td><td>26</td><td>Tool coordinate</td></tr><tr><td>13</td><td>User coordinate 12</td><td>27</td><td>Master tool coordinate</td></tr></tbody></table>	Coordinate type	Coordinate name	Coordinate type	Coordinate name	0	Base coordinate	14	User coordinate 13	1	Robot coordinate	15	User coordinate 14	2	User coordinate 1	16	User coordinate 15	3	User coordinate 2	17	User coordinate 16	4	User coordinate 3	18	User coordinate 17	5	User coordinate 4	19	User coordinate 18	6	User coordinate 5	20	User coordinate 19	7	User coordinate 6	21	User coordinate 20	8	User coordinate 7	22	User coordinate 21	9	User coordinate 8	23	User coordinate 22	10	User coordinate 9	24	User coordinate 23	11	User coordinate 10	25	User coordinate 24	12	User coordinate 11	26	Tool coordinate	13	User coordinate 12	27	Master tool coordinate
	Coordinate type	Coordinate name	Coordinate type	Coordinate name																																																									
0	Base coordinate	14	User coordinate 13																																																										
1	Robot coordinate	15	User coordinate 14																																																										
2	User coordinate 1	16	User coordinate 15																																																										
3	User coordinate 2	17	User coordinate 16																																																										
4	User coordinate 3	18	User coordinate 17																																																										
5	User coordinate 4	19	User coordinate 18																																																										
6	User coordinate 5	20	User coordinate 19																																																										
7	User coordinate 6	21	User coordinate 20																																																										
8	User coordinate 7	22	User coordinate 21																																																										
9	User coordinate 8	23	User coordinate 22																																																										
10	User coordinate 9	24	User coordinate 23																																																										
11	User coordinate 10	25	User coordinate 24																																																										
12	User coordinate 11	26	Tool coordinate																																																										
13	User coordinate 12	27	Master tool coordinate																																																										
	<div>Form</div> <div>The form data are represented by bit data in decimals.</div> <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><table><tbody><tr><td>0: Flip,</td><td>1: No-Flip</td></tr><tr><td>0: Elbow Above,</td><td>1: Elbow Under</td></tr><tr><td>0: Front Side,</td><td>1: Back Side</td></tr><tr><td>0: R<180,</td><td>1: R>=180</td></tr><tr><td>0: T<180,</td><td>1: T>=180</td></tr><tr><td>0: S<180,</td><td>1: S>=180</td></tr><tr><td>Reserved</td><td></td></tr></tbody></table></div></div> <div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div>	0: Flip,	1: No-Flip	0: Elbow Above,	1: Elbow Under	0: Front Side,	1: Back Side	0: R<180,	1: R>=180	0: T<180,	1: T>=180	0: S<180,	1: S>=180	Reserved																																															
0: Flip,	1: No-Flip																																																												
0: Elbow Above,	1: Elbow Under																																																												
0: Front Side,	1: Back Side																																																												
0: R<180,	1: R>=180																																																												
0: T<180,	1: T>=180																																																												
0: S<180,	1: S>=180																																																												
Reserved																																																													
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)																																																												
REFERENCE	"BscGetVarData" "BscGetVarData2"																																																												

■ BscPutVarData2

FUNCTION	Sets variable data. (robot with 7 axes or more)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPutVarData2(short nCid,short type,short varno,double *p,short axisNum);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number type Variable Types varno Variable number *p Head pointer to the numeric variable storage area axisNum Number of axis</p> <hr/> <p>OUT (Return)</p> <p>None</p> <hr/> <p>Return Value</p> <p>0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100/XRC/MRC.</p> <hr/> <p>Variable Types</p> <p>The variable types are represented as follows.</p> <p>0 : Byte type 1 : Integer type 2 : Double-precision type 3 : Real type 4 : Robot axis position type 5 : Base axis position type 6 : Station axis position type (pulse type only)</p>

REMARKS

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[0]	P[1]	P[2]	P[3]	P[4]
0	-	1	Byte				
1	-	1	Integer	-	-	-	-
2	-	1	Double	-	-	-	-
3	-	1	Real	-	-	-	-
4	Pulse	10	0	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
4	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
5	Pulse	8	0	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
5	XYZ	10	1	Coordinate Type	X-axis (mm)	Y-axis (mm)	Z-axis (mm)
6	Pulse	8	0	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Data Type (Pulse/XYZ)	Number of values	Content				
			P[5]	P[6]	P[7]	P[8]	P[9]
0	-	1					
1	-	1	-	-	-	-	-
2	-	1	-	-	-	-	-
3	-	1	-	-	-	-	-
4	Pulse	10	B-axis Pulses	T-axis Pulses	7thaxis Pulses	8thaxis Pulses	Tool Number
4	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
5	Pulse	8	Base Axis-5 Pulses	Base Axis-6 Pulses	Tool Number	-	-
5	XYZ	10	Rx Angle(deg)	Ry Angle(deg)	Rz Angle(deg)	Form	Tool Number
6	Pulse	8	Station Axis-5 Pulses	Station Axis-6 Pulses	Tool Number	-	-

The robot axis position and base axis position type variables include the pulse type and XYZ type, according to the first return value. The station axis position type variable contains the pulse type only. See the following for details on the coordinate system types and form.

REMARKS

Coordinate Types**FS100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	10	User coordinate 9
1	Robot coordinate	11	User coordinate 10
2	User coordinate 1	12	User coordinate 11
3	User coordinate 2	13	User coordinate 12
4	User coordinate 3	14	User coordinate 13
5	User coordinate 4	15	User coordinate 14
6	User coordinate 5	16	User coordinate 15
7	User coordinate 6	17	User coordinate 16
8	User coordinate 7	18	Tool coordinate
9	User coordinate 8	19	Master tool coordinate

Coordinate Types**DX100**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	:	:
1	Robot coordinate	:	:
2	User coordinate 1	63	User coordinate 62
3	User coordinate 2	64	User coordinate 63
:	:	65	Tool coordinate
:	:	66	Master tool coordinate

Coordinate Types**NX100/XRC**

The following coordinate names correspond to the coordinate type data.

Coordinate type	Coordinate name	Coordinate type	Coordinate name
0	Base coordinate	14	User coordinate 13
1	Robot coordinate	15	User coordinate 14
2	User coordinate 1	16	User coordinate 15
3	User coordinate 2	17	User coordinate 16
4	User coordinate 3	18	User coordinate 17
5	User coordinate 4	19	User coordinate 18
6	User coordinate 5	20	User coordinate 19
7	User coordinate 6	21	User coordinate 20
8	User coordinate 7	22	User coordinate 21
9	User coordinate 8	23	User coordinate 22
10	User coordinate 9	24	User coordinate 23
11	User coordinate 10	25	User coordinate 24
12	User coordinate 11	26	Tool coordinate
13	User coordinate 12	27	Master tool coordinate

REMARKS	<p>Form</p> <p>The form data are represented by bit data in decimals.</p> <p>D7 D6 D5 D4 D3 D2 D1 D0</p> <p> 0:Flip, 1:No-Flip 0:Elbow Abov, 1: Elbow Under 0:Front Side, 1: Back Side 0:R<180, 1:R>=180 0:T<180, 1:T>=180 0:S<180, 1:S>=180 Reserved </p>
CONTROLLER	FS100, DX100, NX100, XRC (Serial Port, Ethernet)
REFERENCE	"BscGetVarData" "BscGetVarData2"

■ BscStartJob

FUNCTION	Starts job. (A job to be started has the job name which is selected last.)
FORMAT	_declspec(dllexport) short APIENTRY BscStartJob(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion 1 : Current job name not specified Others : Error codes
REMARKS	Call Condition The BscSelectJob function must be called up and the current job name must be specified before executing this function. To restart a job during startup that has been held by the "BscHoldOn" function, release the hold by "BscHoldOff" function to call up the BscContinueJob function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscContinueJob" "BscSelectJob"

■ BscSelectJob

FUNCTION	Selects a job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSelectJob(short nCid,char *name);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *name Job name storage pointer
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Job name The character string for the job name is restricted as follows. Character string with 30 characters maximum. (8 characters that can be used in the MS-DOS) Specify "" instead of the job name to select all the jobs.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscFindFirstMaster" "BscDeleteJob" "BscSetMasterJob" "BscSetLineNumber" "BscStartJob"

■ BscSelectMode

FUNCTION	Selects mode. (Teach or Play)
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSelectMode(short nCid,short mode);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number mode Selected mode
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Selected Mode The selected mode is represented as follows. 1 : Teach 2 : Play
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscSelLoopCycle

FUNCTION	Changes the cycle mode to auto mode.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSelLoopCycle(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscSelStepCycle" "BscSelOneCycle"

■ BscSelOneCycle

FUNCTION	Changes the cycle mode to 1-cycle mode.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSelOneCycle(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscSelStepCycle" "BscSelLoopCycle"

■ BscSelStepCycle

FUNCTION	Changes the cycle mode to step mode.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSelStepCycle(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscSelOneCycle" "BscSelLoopCycle"

■ BscSetLineNumber

FUNCTION	Sets a line number of current job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetLineNumber(short nCid,short line);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number line Line number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscSelectJob"

■ BscSetMasterJob

FUNCTION	Sets a job as a master job.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetMasterJob(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Call Condition The BscSelectJob function must be called up and the registered job must be specified before executing this function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDeleteJob" "BscSelectJob"

■ BscReset


FUNCTION	Resets a robot alarm.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReset(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscCancel"

■ BscSetCtrlGroup

FUNCTION	Sets a control group.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetCtrlGroup(short nCid,short groupno);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number groupno Control group information to be set
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
REMARKS	Restrictions This function is effective only for transmission with the MRC. Refer to BscSetCtrlGroupDX for the transmission with the DX100. Refer to BscSetCtrlGroupXrc for the transmission with the FS100/ NX100/XRC. When the power supply of robot controller is started up, robot 1, base 1, and station 1 (when a base and a stations exist) are speci- fied. In a system with a base axis (such as travel axis), when the manipulator with this base axis is specified, this base axis is auto- matically specified. The following settings can not be made. - Selection of control axis which does not exist - Simultaneous specification of R1 and R2 - Specification of multiple number of stations Control Group Information The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot 1) D1 : R2 (Robot 2) D2 : S1 (Station 1) D3 : S2 (Station 2) D4 : S3 (Station 3) D5 : S4 (Station 4) D6 : S5 (Station 5) D7 : S6 (Station 6)</div></div>
CONTROLLER	MRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BsclsCtrlGroupDX" "BscSetCtrlGroupXrc" "BscGetCtrlGroupXrc" "BsclsCtrlGroupXrc" "BsclsTaskInfXrc" "BscGetCtrlGroup" "BsclsCtrlGroup" "BsclsTaskInf" "BscChangeTask"

■ BscSetCtrlGroupXrc

FUNCTION	Sets a control group.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetCtrlGroupXrc(short nCid,short groupno1,short groupno2);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number groupno1 Control group information to be set (robot axis) groupno2 Control group information to be set (station axis)</div><div>OUT (Return) groupno1 Control group information to be set (robot axis) groupno2 Control group information to be set (station axis)</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>
REMARKS	<div><div>Restrictions This function is effective only for transmission with the FS100/ NX100/XRC. Refer to BscSetCtrlGroupDX for the transmission with the DX100. Refer to BscSetCtrlGroup for transmission with the MRC. When the power supply of robot controller is started up, robot 1, base 1, and station 1 (when a base and a stations exist) are specified. In a system with a base axis (such as travel axis), when the manipulator with this base axis is specified, this base axis is automatically specified. The following settings can not be made. - Selection of control axis which does not exist - Simultaneous specification of R1 and R2 - Specification of multiple number of stations</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot 1) D1 : R2 (Robot 2) D2 : R3 (Robot 3) D3 : R4 (Robot 4)</div></div></div></div>

REMARKS	Control Group Information (Station Axis) The control group information is represented by bit data in decimals. D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0  D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12)
CONTROLLER	FS100, NX100, XRC (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscSetCtrlGroupDX" "BscIsCtrlGroupDX" "BscGetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInfXrc" "BscSetCtrlGroup" "BscGetCtrlGroup" "BscIsCtrlGroup" "BscIsTaskInf" "BscChangeTask"

■ BscSetCtrlGroupDX

FUNCTION	Sets a control group.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetCtrlGroupDX(short nCid,long groupno1,long groupno2);</code>
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number groupno1 Control group information to be set (robot axis) groupno2 Control group information to be set (station axis)</div><div>OUT (Return) groupno1 Control group information to be set (robot axis) groupno2 Control group information to be set (station axis)</div><div>Return Value 0 : Normal completion Others : Error codes</div></div>
REMARKS	<div><div>Restrictions This function is effective only for transmission with the DX100. Refer to BscSetCtrlGroupXrc for transmission with the FS100/NX100/XRC. Refer to BscSetCtrlGroup for transmission with the MRC. When the power supply of robot controller is started up, robot 1, base 1, and station 1 (when a base and a stations exist) are specified. In a system with a base axis (such as travel axis), when the manipulator with this base axis is specified, this base axis is automatically specified. The following settings can not be made. - Selection of control axis which does not exist - Simultaneous specification of R1 and R2 - Specification of multiple number of stations</div><div>Control Group Information (Robot Axis) The control group information is represented by bit data in decimals. <div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>D0 : R1 (Robot 1) D1 : R2 (Robot 2) D2 : R3 (Robot 3) D3 : R4 (Robot 4) : : D7 : R8 (Robot 8)</div></div></div></div>

REMARKS	Control Group Information (Station Axis) The control group information is represented by bit data in decimals. <div style="text-align: center;"> D23 ... D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0 </div> <div style="text-align: center;"> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> ... <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> <div style="border: 1px solid black; display: inline-block; width: 15px; height: 15px;"></div> </div> D0 : S1 (Station1) D1 : S2 (Station2) D2 : S3 (Station3) D3 : S4 (Station4) D4 : S5 (Station5) D5 : S6 (Station6) D6 : S7 (Station7) D7 : S8 (Station8) D8 : S9 (Station9) D9 : S10 (Station10) D10 : S11 (Station11) D11 : S12 (Station12) : : D23 : S24 (Station24)
CONTROLLER	DX100 (Serial Port, Ethernet)
REFERENCE	"BscGetCtrlGroupDX" "BscIsCtrlGroupDX" "BscSetCtrlGroupXrc" "BscGetCtrlGroupXrc" "BscIsCtrlGroupXrc" "BscIsTaskInfXrc" "BscSetCtrlGroup" "BscGetCtrlGroup" "BscIsCtrlGroup" "BscIsTaskInf" "BscChangeTask"

■ BscServoOff

FUNCTION	Sets servo power supply OFF.
FORMAT	_declspec(dllexport) short APIENTRY BscServoOff(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscServoOn" "BscIsServo"

■ BscServoOn

FUNCTION	Sets servo power supply ON.
FORMAT	_declspec(dllexport) short APIENTRY BscServoOn(short nCid);
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Error codes
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscServoOff" "BsclsServo"

7.4 DCI Function

Job save, load, or variable load, save are automatically accomplished when the robot is under playback mode, by preparing the functions corresponding to the instructions.

Reads the robot status (current position, alarm, error, servo status, etc.), and controls the system (start, hold, job call, etc.).

The following functions are available.

- BscDCILoadSave
- BscDCILoadSaveOnce
- BscDCIGetPos
- BscDCIGetPos2
- BscDCIGetVarData
- BscDCIGetVarDataEx
- BscDCIPutPos
- BscDCIPutPos2
- BscDCIPutVarData
- BscDCIPutVarDataEx

■ BscDCILoadSave

FUNCTION	Loads or saves a job with DCI instruction.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCILoadSave(short nCid,short timec);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number timec Waiting time for sending/receiving (sec)
	OUT (Return) None
	Return Value -1 : Failed to send/receive Others : Number of received jobs
REMARKS	Number of Sending/Receiving This function retries communication of the sending/receiving signal until the specified waiting time comes.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDCILoadSaveOnce" "BscDCIGetPos" "BscDCIPutPos" "BscDCIGetPos2" "BscDCIPutPos2" "BscDCIGetVarData" "BscDCIPutVarData"

■ BscDCILoadSaveOnce

FUNCTION	Loads or saves a job with DCI instruction.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCILoadSaveOnce(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value -1 : Failed to send/receive Others : Number of received jobs
REMARKS	Number of Sending/Receiving This function waits indefinitely until sending/receiving request is sent from the robot. Communication is accomplished a single time when the request arrives.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDCILoadSave" "BscDCIGetPos" "BscDCIPutPos" "BscDCIGetPos2" "BscDCIPutPos2" "BscDCIGetVarData" "BscDCIPutVarData"

■ BscDCIGetPos

FUNCTION	Gets a variable with DCI instruction.																																				
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCIGetPos(short nCid,short *type,short *rconf,double *p);</code>																																				
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *type Variable type number (pointer) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area</div><div>OUT (Return) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area</div><div>Return Value -1 : Failed to send Others : Variable type number</div></div>																																				
REMARKS	<div><div>Variable Type Number The variable type number is represented as follows.</div><table><tr><th>Variable Contents</th><th>FS100/DX100/NX100/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td colspan="2">Byte type</td></tr><tr><td>2</td><td colspan="2">Integer type</td></tr><tr><td>3</td><td colspan="2">Double-precision type</td></tr><tr><td>4</td><td colspan="2">Real type</td></tr><tr><td>5</td><td colspan="2">Robot axis position type (pulse)</td></tr><tr><td>6</td><td colspan="2">Robot axis position type (XYZ)</td></tr><tr><td>7</td><td colspan="2">Base axis position type (pulse)</td></tr><tr><td>8</td><td colspan="2">Base axis position type (XYZ)</td></tr><tr><td>9</td><td>Station axis position type (pulse)</td><td>-</td></tr></table></div> <div><div>Variable type number and the storage area, number of values.</div><table><tr><th>Variable Type Number</th><th>Storage area</th><th>Number of values</th></tr><tr><td>1 ~ 9</td><td>p</td><td>6</td></tr></table></div>	Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC	1	Byte type		2	Integer type		3	Double-precision type		4	Real type		5	Robot axis position type (pulse)		6	Robot axis position type (XYZ)		7	Base axis position type (pulse)		8	Base axis position type (XYZ)		9	Station axis position type (pulse)	-	Variable Type Number	Storage area	Number of values	1 ~ 9	p	6
Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC																																			
1	Byte type																																				
2	Integer type																																				
3	Double-precision type																																				
4	Real type																																				
5	Robot axis position type (pulse)																																				
6	Robot axis position type (XYZ)																																				
7	Base axis position type (pulse)																																				
8	Base axis position type (XYZ)																																				
9	Station axis position type (pulse)	-																																			
Variable Type Number	Storage area	Number of values																																			
1 ~ 9	p	6																																			

REMARKS	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>0:Flip,</div><div>0:Elbow Abov,</div><div>0:Front Side,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div></div><div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div></div></div></div><div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div></div>																																																																																						
	<div><div>Content of the numeric variable storage area</div><div>Depending on the variable type, the numeric variable storage area contains the number of values indicated below.</div><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="6">Content</th></tr><tr><th>P[0]</th><th>P[1]</th><th>P[2]</th><th>P[3]</th><th>P[4]</th><th>P[5]</th></tr><tr><td>1</td><td>1</td><td>Byte</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>1</td><td>Integer</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>Double</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>Real</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>6</td><td>S-axis Pulses</td><td>L-axis Pulses</td><td>U-axis Pulses</td><td>R-axis Pulses</td><td>B-axis Pulses</td><td>T-axis Pulses</td></tr><tr><td>6</td><td>6</td><td>X-axis (mm)</td><td>Y-axis (mm)</td><td>Z-axis (mm)</td><td>Wrist angle Rx (deg)</td><td>Wrist angle Ry (deg)</td><td>Wrist angle Rz (deg)</td></tr><tr><td>7</td><td>6</td><td>Base Axis-1 Pulses</td><td>Base Axis-2 Pulses</td><td>Base Axis-3 Pulses</td><td>Base Axis-4 Pulses</td><td>Base Axis-5 Pulses</td><td>Base Axis-6 Pulses</td></tr><tr><td>8</td><td>6</td><td>Base Axis-1 (mm)</td><td>Base Axis-2 (mm)</td><td>Base Axis-3 (mm)</td><td>Base Axis-4 (mm)</td><td>Base Axis-5 (mm)</td><td>Base Axis-6 (mm)</td></tr><tr><td>9</td><td>6</td><td>Station Axis-1 Pulses</td><td>Station Axis-2 Pulses</td><td>Station Axis-3 Pulses</td><td>Station Axis-4 Pulses</td><td>Station Axis-5 Pulses</td><td>Station Axis-6 Pulses</td></tr></table></div>	Variable Type Number	Number of values	Content						P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	1	1	Byte						2	1	Integer	-	-	-	-	-	3	1	Double	-	-	-	-	-	4	1	Real	-	-	-	-	-	5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses	6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)	7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses	8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)	9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses
Variable Type Number	Number of values			Content																																																																																			
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]																																																																																
1	1	Byte																																																																																					
2	1	Integer	-	-	-	-	-																																																																																
3	1	Double	-	-	-	-	-																																																																																
4	1	Real	-	-	-	-	-																																																																																
5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses																																																																																
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)																																																																																
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses																																																																																
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)																																																																																
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses																																																																																
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																																																																						
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData"																																																																																						

■ BscDCIGetPos2

FUNCTION	Gets a variable with DCI instruction. (robots with 7 axes or more)																																				
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCIGetPos2(short nCid,short *type,short *rconf,double *p,short *axisNum);</code>																																				
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *type Variable type number (pointer) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area *axisNum Number of axis (pointer)</div><div>OUT (Return) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area</div><div>Return Value -1 : Failed to send Others : Variable type number</div></div>																																				
REMARKS	<div><div>Variable Type Number The variable type number is represented as follows.</div><table><tr><th>Variable Contents</th><th>FS100/DX100/NX100/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td>Byte type</td><td>Byte type</td></tr><tr><td>2</td><td>Integer type</td><td>Integer type</td></tr><tr><td>3</td><td>Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td>Real type</td><td>Real type</td></tr><tr><td>5</td><td>Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td>Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td>Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td>Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td>Station axis position type (pulse)</td><td>-</td></tr></table></div> <div><div>Variable type number and the storage area, number of values.</div><table><tr><th>Variable Type Number</th><th>Storage area</th><th>Number of values</th></tr><tr><td>1 ~ 9</td><td>p</td><td>8</td></tr></table></div>	Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC	1	Byte type	Byte type	2	Integer type	Integer type	3	Double-precision type	Double-precision type	4	Real type	Real type	5	Robot axis position type (pulse)	Robot axis position type (pulse)	6	Robot axis position type (XYZ)	Robot axis position type (XYZ)	7	Base axis position type (pulse)	External axis position type (pulse)	8	Base axis position type (XYZ)	External axis position type (XYZ)	9	Station axis position type (pulse)	-	Variable Type Number	Storage area	Number of values	1 ~ 9	p	8
Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC																																			
1	Byte type	Byte type																																			
2	Integer type	Integer type																																			
3	Double-precision type	Double-precision type																																			
4	Real type	Real type																																			
5	Robot axis position type (pulse)	Robot axis position type (pulse)																																			
6	Robot axis position type (XYZ)	Robot axis position type (XYZ)																																			
7	Base axis position type (pulse)	External axis position type (pulse)																																			
8	Base axis position type (XYZ)	External axis position type (XYZ)																																			
9	Station axis position type (pulse)	-																																			
Variable Type Number	Storage area	Number of values																																			
1 ~ 9	p	8																																			

REMARKS	<div><div>Form</div><div>The form data are represented by bit data in decimals.</div><div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>0:Flip,</div><div>0:Elbow Above,</div><div>0:Front Side,</div><div>0:R<180,</div><div>0:T<180,</div><div>0:S<180,</div><div>Reserved</div></div><div><div>1:No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1:R>=180</div><div>1:T>=180</div><div>1:S>=180</div></div></div></div><div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div></div> <tr><td></td><td><div><div>Content of the numeric variable storage area</div><div>Depending on the variable type, the numeric variable storage area contains the number of values indicated below.</div><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[0]</th><th>P[1]</th><th>P[2]</th><th>P[3]</th></tr><tr><td>1</td><td>1</td><td>Byte</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>Integer</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>Double</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>Real</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>S-axis Pulses</td><td>L-axis Pulses</td><td>U-axis Pulses</td><td>R-axis Pulses</td></tr><tr><td>6</td><td>6</td><td>X-axis (mm)</td><td>Y-axis (mm)</td><td>Z-axis (mm)</td><td>Wrist angle Rx (deg)</td></tr><tr><td>7</td><td>6</td><td>Base Axis-1 Pulses</td><td>Base Axis-2 Pulses</td><td>Base Axis-3 Pulses</td><td>Base Axis-4 Pulses</td></tr><tr><td>8</td><td>6</td><td>Base Axis-1 (mm)</td><td>Base Axis-2 (mm)</td><td>Base Axis-3 (mm)</td><td>Base Axis-4 (mm)</td></tr><tr><td>9</td><td>6</td><td>Station Axis-1 Pulses</td><td>Station Axis-2 Pulses</td><td>Station Axis-3 Pulses</td><td>Station Axis-4 Pulses</td></tr></table><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[4]</th><th>P[5]</th><th>P[6]</th><th>P[7]</th></tr><tr><td>1</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>B-axis Pulses</td><td>T-axis Pulses</td><td>7th axis Pulses</td><td>8th axis Pulses</td></tr><tr><td>6</td><td>6</td><td>Wrist angle Ry (deg)</td><td>Wrist angle Rz (deg)</td><td>-</td><td>-</td></tr><tr><td>7</td><td>6</td><td>Base Axis-5 Pulses</td><td>Base Axis-6 Pulses</td><td>-</td><td>-</td></tr><tr><td>8</td><td>6</td><td>Base Axis-5 (mm)</td><td>Base Axis-6 (mm)</td><td>-</td><td>-</td></tr><tr><td>9</td><td>6</td><td>Station Axis-5 Pulses</td><td>Station Axis-6 Pulses</td><td>-</td><td>-</td></tr></table></div></td></tr> <tr><td>CONTROLLER</td><td>FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)</td></tr> <tr><td>REFERENCE</td><td>"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData"</td></tr>		<div><div>Content of the numeric variable storage area</div><div>Depending on the variable type, the numeric variable storage area contains the number of values indicated below.</div><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[0]</th><th>P[1]</th><th>P[2]</th><th>P[3]</th></tr><tr><td>1</td><td>1</td><td>Byte</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>Integer</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>Double</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>Real</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>S-axis Pulses</td><td>L-axis Pulses</td><td>U-axis Pulses</td><td>R-axis Pulses</td></tr><tr><td>6</td><td>6</td><td>X-axis (mm)</td><td>Y-axis (mm)</td><td>Z-axis (mm)</td><td>Wrist angle Rx (deg)</td></tr><tr><td>7</td><td>6</td><td>Base Axis-1 Pulses</td><td>Base Axis-2 Pulses</td><td>Base Axis-3 Pulses</td><td>Base Axis-4 Pulses</td></tr><tr><td>8</td><td>6</td><td>Base Axis-1 (mm)</td><td>Base Axis-2 (mm)</td><td>Base Axis-3 (mm)</td><td>Base Axis-4 (mm)</td></tr><tr><td>9</td><td>6</td><td>Station Axis-1 Pulses</td><td>Station Axis-2 Pulses</td><td>Station Axis-3 Pulses</td><td>Station Axis-4 Pulses</td></tr></table><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[4]</th><th>P[5]</th><th>P[6]</th><th>P[7]</th></tr><tr><td>1</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>B-axis Pulses</td><td>T-axis Pulses</td><td>7th axis Pulses</td><td>8th axis Pulses</td></tr><tr><td>6</td><td>6</td><td>Wrist angle Ry (deg)</td><td>Wrist angle Rz (deg)</td><td>-</td><td>-</td></tr><tr><td>7</td><td>6</td><td>Base Axis-5 Pulses</td><td>Base Axis-6 Pulses</td><td>-</td><td>-</td></tr><tr><td>8</td><td>6</td><td>Base Axis-5 (mm)</td><td>Base Axis-6 (mm)</td><td>-</td><td>-</td></tr><tr><td>9</td><td>6</td><td>Station Axis-5 Pulses</td><td>Station Axis-6 Pulses</td><td>-</td><td>-</td></tr></table></div>	Variable Type Number	Number of values	Content				P[0]	P[1]	P[2]	P[3]	1	1	Byte	-	-	-	2	1	Integer	-	-	-	3	1	Double	-	-	-	4	1	Real	-	-	-	5	8	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Variable Type Number	Number of values	Content				P[4]	P[5]	P[6]	P[7]	1	1	-	-	-	-	2	1	-	-	-	-	3	1	-	-	-	-	4	1	-	-	-	-	5	8	B-axis Pulses	T-axis Pulses	7 th axis Pulses	8 th axis Pulses	6	6	Wrist angle Ry (deg)	Wrist angle Rz (deg)	-	-	7	6	Base Axis-5 Pulses	Base Axis-6 Pulses	-	-	8	6	Base Axis-5 (mm)	Base Axis-6 (mm)	-	-	9	6	Station Axis-5 Pulses	Station Axis-6 Pulses	-	-	CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)	REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData"
	<div><div>Content of the numeric variable storage area</div><div>Depending on the variable type, the numeric variable storage area contains the number of values indicated below.</div><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[0]</th><th>P[1]</th><th>P[2]</th><th>P[3]</th></tr><tr><td>1</td><td>1</td><td>Byte</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>Integer</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>Double</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>Real</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>S-axis Pulses</td><td>L-axis Pulses</td><td>U-axis Pulses</td><td>R-axis Pulses</td></tr><tr><td>6</td><td>6</td><td>X-axis (mm)</td><td>Y-axis (mm)</td><td>Z-axis (mm)</td><td>Wrist angle Rx (deg)</td></tr><tr><td>7</td><td>6</td><td>Base Axis-1 Pulses</td><td>Base Axis-2 Pulses</td><td>Base Axis-3 Pulses</td><td>Base Axis-4 Pulses</td></tr><tr><td>8</td><td>6</td><td>Base Axis-1 (mm)</td><td>Base Axis-2 (mm)</td><td>Base Axis-3 (mm)</td><td>Base Axis-4 (mm)</td></tr><tr><td>9</td><td>6</td><td>Station Axis-1 Pulses</td><td>Station Axis-2 Pulses</td><td>Station Axis-3 Pulses</td><td>Station Axis-4 Pulses</td></tr></table><table><tr><th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="4">Content</th></tr><tr><th>P[4]</th><th>P[5]</th><th>P[6]</th><th>P[7]</th></tr><tr><td>1</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>2</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>3</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>4</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td></tr><tr><td>5</td><td>8</td><td>B-axis Pulses</td><td>T-axis Pulses</td><td>7th axis Pulses</td><td>8th axis Pulses</td></tr><tr><td>6</td><td>6</td><td>Wrist angle Ry (deg)</td><td>Wrist angle Rz (deg)</td><td>-</td><td>-</td></tr><tr><td>7</td><td>6</td><td>Base Axis-5 Pulses</td><td>Base Axis-6 Pulses</td><td>-</td><td>-</td></tr><tr><td>8</td><td>6</td><td>Base Axis-5 (mm)</td><td>Base Axis-6 (mm)</td><td>-</td><td>-</td></tr><tr><td>9</td><td>6</td><td>Station Axis-5 Pulses</td><td>Station Axis-6 Pulses</td><td>-</td><td>-</td></tr></table></div>	Variable Type Number	Number of values			Content				P[0]	P[1]	P[2]	P[3]	1	1	Byte	-	-	-	2	1	Integer	-	-	-	3	1	Double	-	-	-	4	1	Real	-	-	-	5	8	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses			Variable Type Number	Number of values	Content				P[4]	P[5]	P[6]	P[7]	1	1	-	-	-	-	2	1	-	-	-	-	3	1	-	-	-	-	4	1	-	-	-	-	5	8	B-axis Pulses	T-axis Pulses	7 th axis Pulses	8 th axis Pulses	6	6	Wrist angle Ry (deg)	Wrist angle Rz (deg)	-	-	7	6	Base Axis-5 Pulses	Base Axis-6 Pulses	-	-	8	6	Base Axis-5 (mm)	Base Axis-6 (mm)	-	-	9	6	Station Axis-5 Pulses	Station Axis-6 Pulses	-	-		
Variable Type Number	Number of values			Content																																																																																																																																			
		P[0]	P[1]	P[2]	P[3]																																																																																																																																		
1	1	Byte	-	-	-																																																																																																																																		
2	1	Integer	-	-	-																																																																																																																																		
3	1	Double	-	-	-																																																																																																																																		
4	1	Real	-	-	-																																																																																																																																		
5	8	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses																																																																																																																																		
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)																																																																																																																																		
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses																																																																																																																																		
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)																																																																																																																																		
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses																																																																																																																																		
Variable Type Number	Number of values	Content																																																																																																																																					
		P[4]	P[5]	P[6]	P[7]																																																																																																																																		
1	1	-	-	-	-																																																																																																																																		
2	1	-	-	-	-																																																																																																																																		
3	1	-	-	-	-																																																																																																																																		
4	1	-	-	-	-																																																																																																																																		
5	8	B-axis Pulses	T-axis Pulses	7 th axis Pulses	8 th axis Pulses																																																																																																																																		
6	6	Wrist angle Ry (deg)	Wrist angle Rz (deg)	-	-																																																																																																																																		
7	6	Base Axis-5 Pulses	Base Axis-6 Pulses	-	-																																																																																																																																		
8	6	Base Axis-5 (mm)	Base Axis-6 (mm)	-	-																																																																																																																																		
9	6	Station Axis-5 Pulses	Station Axis-6 Pulses	-	-																																																																																																																																		
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																																																																																																																						
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData"																																																																																																																																						

■ BscDCIGetVarData

FUNCTION	Gets a variable with DCI instruction.																																														
FORMAT	_declspec(dllexport) short APIENTRY BscDCIGetVarData(short nCid,short *type,short *rconf,double *p, char *str);																																														
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *type Variable type number (pointer) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area																																														
	OUT (Return) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area																																														
	Return Value -1 : Failed to send Others : Variable type number																																														
	REMARKS																																														
	Restrictions String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.																																														
	Variable Type Number The variable type number is represented as follows.																																														
	<table><tr><th>Variable Contents</th><th>FS100/DX100/ NX100 (v3.0 and after)</th><th>NX100 (before v3.0)/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td colspan="2">Byte type</td><td>Byte type</td></tr><tr><td>2</td><td colspan="2">Integer type</td><td>Integer type</td></tr><tr><td>3</td><td colspan="2">Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td colspan="2">Real type</td><td>Real type</td></tr><tr><td>5</td><td colspan="2">Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td colspan="2">Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td colspan="2">Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td colspan="2">Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td colspan="2">Station axis position type (pulse)</td><td>-</td></tr><tr><td>10</td><td>String type</td><td>-</td><td>-</td></tr></table>			Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC	1	Byte type		Byte type	2	Integer type		Integer type	3	Double-precision type		Double-precision type	4	Real type		Real type	5	Robot axis position type (pulse)		Robot axis position type (pulse)	6	Robot axis position type (XYZ)		Robot axis position type (XYZ)	7	Base axis position type (pulse)		External axis position type (pulse)	8	Base axis position type (XYZ)		External axis position type (XYZ)	9	Station axis position type (pulse)		-	10	String type	-	-
Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC																																												
1	Byte type		Byte type																																												
2	Integer type		Integer type																																												
3	Double-precision type		Double-precision type																																												
4	Real type		Real type																																												
5	Robot axis position type (pulse)		Robot axis position type (pulse)																																												
6	Robot axis position type (XYZ)		Robot axis position type (XYZ)																																												
7	Base axis position type (pulse)		External axis position type (pulse)																																												
8	Base axis position type (XYZ)		External axis position type (XYZ)																																												
9	Station axis position type (pulse)		-																																												
10	String type	-	-																																												
	<div><div><div>NOTE</div></div><div>String type is valid only for the NX100 V3.0 and later, or for the FS100/DX100.</div></div>																																														

REMARKS

Variable type number and the storage area, number of values.

Variable Type Number	Storage area	Number of values
1 ~ 9	p	6
10	str	16



When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17

Declaration in C++: char S_Variable[17]

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



0:Flip, 1:No-Flip
 0:Elbow Above, 1: Elbow Under
 0:Front Side, 1: Back Side
 0:R<180, 1:R>=180
 0:T<180, 1:T>=180
 0:S<180, 1:S>=180
 Reserved

* With the ERC or ERC2, the data from D3 to D7 are disregarded.

* With the MRC or MRC2, the data D5 and D7 are disregarded.

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Number of values	Content					
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]
1	1	Byte					
2	1	Integer	-	-	-	-	-
3	1	Double	-	-	-	-	-
4	1	Real	-	-	-	-	-
5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses

Content of the character variable storage area


Variable Type Number	Number of values	Content
		str
10	16	String

CONTROLLER

FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet)
ERC (Serial Port)

REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData"
-----------	--

■ BscDCIGetVarDataEx

FUNCTION	Gets a variable with DCI instruction.(robots with 7 axes or more)																																												
FORMAT	_declspec(dllexport) short APIENTRY BscDCIGetVarDataEx(short nCid,short *type,long *rconf,double *p, char *str,short *axisNum);																																												
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number *type Variable type number (pointer) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area *axisNum Number of axis (pointer)</div><div>OUT (Return) *rconf Form data (pointer) *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area *axisNum Number of axis (pointer)</div><div>Return Value -1 : Failed to send Others : Variable type number</div></div>																																												
REMARKS	<div><div>Restrictions String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</div><div><div>Variable Type Number The variable type number is represented as follows.</div><table><tr><th>Variable Contents</th><th>FS100/DX100/ NX100 (v3.0 and after)</th><th>NX100 (before v3.0)/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td colspan="2">Byte type</td><td>Byte type</td></tr><tr><td>2</td><td colspan="2">Integer type</td><td>Integer type</td></tr><tr><td>3</td><td colspan="2">Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td colspan="2">Real type</td><td>Real type</td></tr><tr><td>5</td><td colspan="2">Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td colspan="2">Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td colspan="2">Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td colspan="2">Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td colspan="2">Station axis position type (pulse)</td><td>-</td></tr><tr><td>10</td><td>String type</td><td>-</td><td>-</td></tr></table></div><div><div><div>NOTE</div><div><ul style="list-style-type: none">• If the robot controller includes 7-axis robot, P-variable specification is for 7-axis robot.• String type is valid only for the FS100/NX100 V3.0 and later, or for the DX100.</div></div></div></div>	Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC	1	Byte type		Byte type	2	Integer type		Integer type	3	Double-precision type		Double-precision type	4	Real type		Real type	5	Robot axis position type (pulse)		Robot axis position type (pulse)	6	Robot axis position type (XYZ)		Robot axis position type (XYZ)	7	Base axis position type (pulse)		External axis position type (pulse)	8	Base axis position type (XYZ)		External axis position type (XYZ)	9	Station axis position type (pulse)		-	10	String type	-	-
Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC																																										
1	Byte type		Byte type																																										
2	Integer type		Integer type																																										
3	Double-precision type		Double-precision type																																										
4	Real type		Real type																																										
5	Robot axis position type (pulse)		Robot axis position type (pulse)																																										
6	Robot axis position type (XYZ)		Robot axis position type (XYZ)																																										
7	Base axis position type (pulse)		External axis position type (pulse)																																										
8	Base axis position type (XYZ)		External axis position type (XYZ)																																										
9	Station axis position type (pulse)		-																																										
10	String type	-	-																																										

REMARKS

Variable type number and the storage area, number of values.

Variable Type Number	Storage area	Number of values
1 ~ 9	p	7
10	str	16



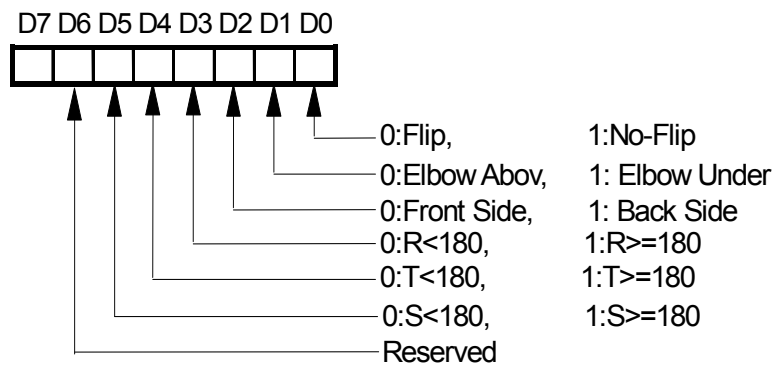
When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17

Declaration in C++: char S_Variable[17]

Form

The form data are represented by bit data in decimals.



- * With the ERC or ERC2, the data from D3 to D7 are disregarded.
- * With the MRC or MRC2, the data D5 and D7 are disregarded.

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Number of values	Content						
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]
1	1	Byte	-	-	-	-	-	-
2	1	Integer	-	-	-	-	-	-
3	1	Double	-	-	-	-	-	-
4	1	Real	-	-	-	-	-	-
5	7	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses	E-axis Pulses
6	7	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)	Re (deg)
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses	-
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)	-
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses	-

Content of the character variable storage area

Variable Type Number	Number of values	Content
		str
10	16	String

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIPutPos" "BscDCIPutPos2" "BscDCIPutVarData" "BscDCIPutVarDataEx"

■ BscDCIPutPos

FUNCTION	Sets a variable with DCI instruction.																																												
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCIPutPos(short nCid,short type,short rconf,double *p);</code>																																												
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number type Variable type number rconf Form data *p Head pointer to the numeric variable storage area</div><div>OUT (Return) None</div><div>Return Value -1 : Failed to send Others : Normal completion</div></div>																																												
REMARKS	<div><div>Variable Type Number The variable type number is represented as follows.</div><table><thead><tr><th>Variable Contents</th><th>FS100/DX100/NX100/ XRC/MRC</th><th>ERC</th></tr></thead><tbody><tr><td>1</td><td>Byte type</td><td>Byte type</td></tr><tr><td>2</td><td>Integer type</td><td>Integer type</td></tr><tr><td>3</td><td>Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td>Real type</td><td>Real type</td></tr><tr><td>5</td><td>Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td>Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td>Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td>Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td>Station axis position type (pulse)</td><td>-</td></tr></tbody></table></div> <div><div>Variable type number and the storage area, number of values.</div><table><thead><tr><th>Variable Type Number</th><th>Storage area</th><th>Number of values</th></tr></thead><tbody><tr><td>1 ~ 9</td><td>p</td><td>6</td></tr></tbody></table></div> <div><div>Form The form data are represented by bit data in decimals.</div><div><div>D7 D6 D5 D4 D3 D2 D1 D0</div><div><table><tbody><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></tbody></table><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div>0: Flip,</div><div>0: Elbow Above,</div><div>0: Front Side,</div><div>0: R<180,</div><div>0: T<180,</div><div>0: S<180,</div><div>Reserved</div><div>1: No-Flip</div><div>1: Elbow Under</div><div>1: Back Side</div><div>1: R>=180</div><div>1: T>=180</div><div>1: S>=180</div></div></div></div></div></div> <div><div>* With the ERC or ERC2, the data from D3 to D7 are disregarded.</div><div>* With the MRC or MRC2, the data D5 and D7 are disregarded.</div></div>	Variable Contents	FS100/DX100/NX100/ XRC/MRC	ERC	1	Byte type	Byte type	2	Integer type	Integer type	3	Double-precision type	Double-precision type	4	Real type	Real type	5	Robot axis position type (pulse)	Robot axis position type (pulse)	6	Robot axis position type (XYZ)	Robot axis position type (XYZ)	7	Base axis position type (pulse)	External axis position type (pulse)	8	Base axis position type (XYZ)	External axis position type (XYZ)	9	Station axis position type (pulse)	-	Variable Type Number	Storage area	Number of values	1 ~ 9	p	6								
Variable Contents	FS100/DX100/NX100/ XRC/MRC	ERC																																											
1	Byte type	Byte type																																											
2	Integer type	Integer type																																											
3	Double-precision type	Double-precision type																																											
4	Real type	Real type																																											
5	Robot axis position type (pulse)	Robot axis position type (pulse)																																											
6	Robot axis position type (XYZ)	Robot axis position type (XYZ)																																											
7	Base axis position type (pulse)	External axis position type (pulse)																																											
8	Base axis position type (XYZ)	External axis position type (XYZ)																																											
9	Station axis position type (pulse)	-																																											
Variable Type Number	Storage area	Number of values																																											
1 ~ 9	p	6																																											

REMARKS	<p>Content of the numeric variable storage area Depending on the variable type, the numeric variable storage area contains the number of values indicated below.</p> <table border="1"> <thead> <tr> <th rowspan="2">Variable Type Number</th><th rowspan="2">Number of values</th><th colspan="6">Content</th></tr> <tr> <th>P[0]</th><th>P[1]</th><th>P[2]</th><th>P[3]</th><th>P[4]</th><th>P[5]</th></tr> </thead> <tbody> <tr> <td>1</td><td>1</td><td>Byte</td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>2</td><td>1</td><td>Integer</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr> <td>3</td><td>1</td><td>Double</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr> <td>4</td><td>1</td><td>Real</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> <tr> <td>5</td><td>6</td><td>S-axis Pulses</td><td>L-axis Pulses</td><td>U-axis Pulses</td><td>R-axis Pulses</td><td>B-axis Pulses</td><td>T-axis Pulses</td></tr> <tr> <td>6</td><td>6</td><td>X-axis (mm)</td><td>Y-axis (mm)</td><td>Z-axis (mm)</td><td>Wrist angle Rx (deg)</td><td>Wrist angle Ry (deg)</td><td>Wrist angle Rz (deg)</td></tr> <tr> <td>7</td><td>6</td><td>Base Axis-1 Pulses</td><td>Base Axis-2 Pulses</td><td>Base Axis-3 Pulses</td><td>Base Axis-4 Pulses</td><td>Base Axis-5 Pulses</td><td>Base Axis-6 Pulses</td></tr> <tr> <td>8</td><td>6</td><td>Base Axis-1 (mm)</td><td>Base Axis-2 (mm)</td><td>Base Axis-3 (mm)</td><td>Base Axis-4 (mm)</td><td>Base Axis-5 (mm)</td><td>Base Axis-6 (mm)</td></tr> <tr> <td>9</td><td>6</td><td>Station Axis-1 Pulses</td><td>Station Axis-2 Pulses</td><td>Station Axis-3 Pulses</td><td>Station Axis-4 Pulses</td><td>Station Axis-5 Pulses</td><td>Station Axis-6 Pulses</td></tr> </tbody> </table>							Variable Type Number	Number of values	Content						P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	1	1	Byte						2	1	Integer	-	-	-	-	-	3	1	Double	-	-	-	-	-	4	1	Real	-	-	-	-	-	5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses	6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)	7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses	8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)	9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses
Variable Type Number	Number of values	Content																																																																																											
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]																																																																																						
1	1	Byte																																																																																											
2	1	Integer	-	-	-	-	-																																																																																						
3	1	Double	-	-	-	-	-																																																																																						
4	1	Real	-	-	-	-	-																																																																																						
5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses																																																																																						
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)																																																																																						
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses																																																																																						
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)																																																																																						
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses																																																																																						
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)																																																																																												
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIGetPos" "BscDCIGetPos2" "BscDCIGetVarData"																																																																																												

■ BscDCIPutPos2

FUNCTION	Sets a variable with DCI instruction. (robots with 7 axes or more)																																				
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDCIPutPos2(short nCid,short type,short rconf,double *p,short axisNum);</code>																																				
ARGUMENTS	IN (Transfer) nCid Communication handler ID number type Variable type number rconf Form data *p Head pointer to the numeric variable storage area axisNum Number of axis																																				
	OUT (Return) None																																				
	Return Value -1 : Failed to send Others : Variable type number																																				
REMARKS	Variable Type Number The variable type number is represented as follows. <table border="1"><thead><tr><th>Variable Contents</th><th>FS100/DX100/NX100/XRC/MRC</th><th>ERC</th></tr></thead><tbody><tr><td>1</td><td>Byte type</td><td>Byte type</td></tr><tr><td>2</td><td>Integer type</td><td>Integer type</td></tr><tr><td>3</td><td>Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td>Real type</td><td>Real type</td></tr><tr><td>5</td><td>Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td>Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td>Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td>Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td>Station axis position type (pulse)</td><td>-</td></tr></tbody></table> Variable type number and the storage area, number of values. <table border="1"><thead><tr><th>Variable Type Number</th><th>Storage area</th><th>Number of values</th></tr></thead><tbody><tr><td>1 ~ 9</td><td>p</td><td>8</td></tr></tbody></table>	Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC	1	Byte type	Byte type	2	Integer type	Integer type	3	Double-precision type	Double-precision type	4	Real type	Real type	5	Robot axis position type (pulse)	Robot axis position type (pulse)	6	Robot axis position type (XYZ)	Robot axis position type (XYZ)	7	Base axis position type (pulse)	External axis position type (pulse)	8	Base axis position type (XYZ)	External axis position type (XYZ)	9	Station axis position type (pulse)	-	Variable Type Number	Storage area	Number of values	1 ~ 9	p	8
Variable Contents	FS100/DX100/NX100/XRC/MRC	ERC																																			
1	Byte type	Byte type																																			
2	Integer type	Integer type																																			
3	Double-precision type	Double-precision type																																			
4	Real type	Real type																																			
5	Robot axis position type (pulse)	Robot axis position type (pulse)																																			
6	Robot axis position type (XYZ)	Robot axis position type (XYZ)																																			
7	Base axis position type (pulse)	External axis position type (pulse)																																			
8	Base axis position type (XYZ)	External axis position type (XYZ)																																			
9	Station axis position type (pulse)	-																																			
Variable Type Number	Storage area	Number of values																																			
1 ~ 9	p	8																																			

REMARKS

Form

The form data are represented by bit data in decimals.

D7 D6 D5 D4 D3 D2 D1 D0



- 0:Flip, 1:No-Flip
- 0:Elbow Above, 1: Elbow Under
- 0:Front Side, 1: Back Side
- 0:R<180, 1:R>=180
- 0:T<180, 1:T>=180
- 0:S<180, 1:S>=180
- Reserved

- * With the ERC or ERC2, the data from D3 to D7 are disregarded.
- * With the MRC or MRC2, the data D5 and D7 are disregarded.

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Number of values	Content			
		P[0]	P[1]	P[2]	P[3]
1	1	Byte	-	-	-
2	1	Integer	-	-	-
3	1	Double	-	-	-
4	1	Real	-	-	-
5	8	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses

Variable Type Number	Number of values	Content			
		P[4]	P[5]	P[6]	P[7]
1	1	-	-	-	-
2	1	-	-	-	-
3	1	-	-	-	-
4	1	-	-	-	-
5	8	B-axis Pulses	T-axis Pulses	7 th axis Pulses	8 th axis Pulses
6	6	Wrist angle Ry (deg)	Wrist angle Rz (deg)	-	-
7	6	Base Axis-5 Pulses	Base Axis-6 Pulses	-	-
8	6	Base Axis-5 (mm)	Base Axis-6 (mm)	-	-
9	6	Station Axis-5 Pulses	Station Axis-6 Pulses	-	-

CONTROLLER

FS100, DX100, NX100, XRC (Serial Port, Ethernet)

REFERENCE

"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIGetPos" "BscDCIGetPos2" "BscDCIGetVarData"

■ BscDCIPutVarData

FUNCTION	Sets a variable with DCI instruction.																																												
FORMAT	_declspec(dllexport) short APIENTRY BscDCIPutVarData(short nCid,short type,short rconf,double *p, char *str);																																												
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number type Variable type number rconf Form data *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area</div><div>OUT (Return) None</div><div>Return Value -1 : Failed to send Others : Variable type number</div></div>																																												
REMARKS	<div><div>Restrictions String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</div><div><div>Variable Type Number The variable type number is represented as follows.</div><table><tr><th>Variable Contents</th><th>FS100/DX100/ NX100 (v3.0 and after)</th><th>NX100 (before v3.0)/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td colspan="2">Byte type</td><td>Byte type</td></tr><tr><td>2</td><td colspan="2">Integer type</td><td>Integer type</td></tr><tr><td>3</td><td colspan="2">Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td colspan="2">Real type</td><td>Real type</td></tr><tr><td>5</td><td colspan="2">Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td colspan="2">Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td colspan="2">Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td colspan="2">Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td colspan="2">Station axis position type (pulse)</td><td>-</td></tr><tr><td>10</td><td>String type</td><td>-</td><td>-</td></tr></table></div><div><div><div><div><div></div><div>NOTE</div></div><div></div></div><div>String type is valid only for the NX100 V3.0 and later, or for the FS100/DX100.</div></div></div></div>	Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC	1	Byte type		Byte type	2	Integer type		Integer type	3	Double-precision type		Double-precision type	4	Real type		Real type	5	Robot axis position type (pulse)		Robot axis position type (pulse)	6	Robot axis position type (XYZ)		Robot axis position type (XYZ)	7	Base axis position type (pulse)		External axis position type (pulse)	8	Base axis position type (XYZ)		External axis position type (XYZ)	9	Station axis position type (pulse)		-	10	String type	-	-
Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC																																										
1	Byte type		Byte type																																										
2	Integer type		Integer type																																										
3	Double-precision type		Double-precision type																																										
4	Real type		Real type																																										
5	Robot axis position type (pulse)		Robot axis position type (pulse)																																										
6	Robot axis position type (XYZ)		Robot axis position type (XYZ)																																										
7	Base axis position type (pulse)		External axis position type (pulse)																																										
8	Base axis position type (XYZ)		External axis position type (XYZ)																																										
9	Station axis position type (pulse)		-																																										
10	String type	-	-																																										

REMARKS

Variable type number and the storage area, number of values.

Variable Type Number	Storage area	Number of values
1 ~ 9	p	6
10	str	16

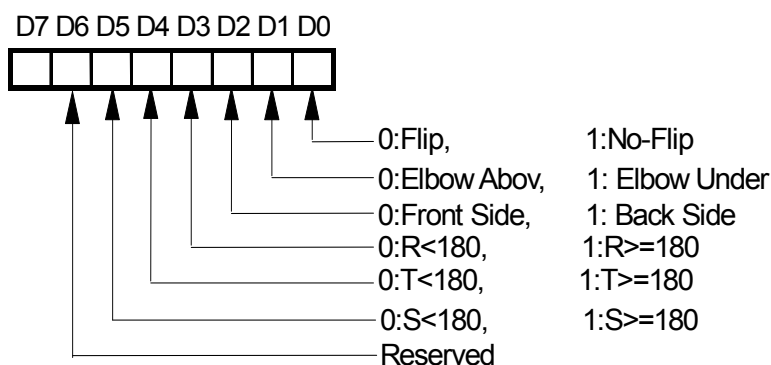


When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17
Declaration in C++: char S_Variable[17]

Form

The form data are represented by bit data in decimals.



- * With the ERC or ERC2, the data from D3 to D7 are disregarded.
- * With the MRC or MRC2, the data D5 and D7 are disregarded.

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Number of values	Content					
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]
1	1	Byte	-	-	-	-	-
2	1	Integer	-	-	-	-	-
3	1	Double	-	-	-	-	-
4	1	Real	-	-	-	-	-
5	6	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses
6	6	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses

Content of the character variable storage area

Variable Type Number	Number of values	Content
		str
10	16	String

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIGetPos" "BscDCIGetPos2" "BscDCIGetVarData"

■ BscDCIPutVarDataEx

FUNCTION	Sets a variable with DCI instruction.(robots with 7 axes or more)																																												
FORMAT	_declspec(dllexport) short APIENTRY BscDCIPutVarDataEx(short nCid,short type,long rconf,double *p, char *str,short axisNum);																																												
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number type Variable type number rconf Form data *p Head pointer to the numeric variable storage area *str Head pointer to the character variable storage area axisNum Number of axis</div><div>OUT (Return) None</div><div>Return Value -1 : Failed to send Others : Variable type number</div></div>																																												
REMARKS	<div><div>Restrictions String variables can only be used with the FS100/DX100 or NX100 ver3.0 or later.</div><div><div>Variable Type Number The variable type number is represented as follows.</div><table><tr><th>Variable Contents</th><th>FS100/DX100/ NX100 (v3.0 and after)</th><th>NX100 (before v3.0)/XRC/MRC</th><th>ERC</th></tr><tr><td>1</td><td colspan="2">Byte type</td><td>Byte type</td></tr><tr><td>2</td><td colspan="2">Integer type</td><td>Integer type</td></tr><tr><td>3</td><td colspan="2">Double-precision type</td><td>Double-precision type</td></tr><tr><td>4</td><td colspan="2">Real type</td><td>Real type</td></tr><tr><td>5</td><td colspan="2">Robot axis position type (pulse)</td><td>Robot axis position type (pulse)</td></tr><tr><td>6</td><td colspan="2">Robot axis position type (XYZ)</td><td>Robot axis position type (XYZ)</td></tr><tr><td>7</td><td colspan="2">Base axis position type (pulse)</td><td>External axis position type (pulse)</td></tr><tr><td>8</td><td colspan="2">Base axis position type (XYZ)</td><td>External axis position type (XYZ)</td></tr><tr><td>9</td><td colspan="2">Station axis position type (pulse)</td><td>-</td></tr><tr><td>10</td><td>String type</td><td>-</td><td>-</td></tr></table></div><div><div><div><div>NOTE</div></div><div><ul style="list-style-type: none">• If the robot controller includedes 7-axis robot, P-variable specification is for 7-axis robot.• String type is valid only for the NX100 V3.0 and later, or for the FS100/DX100.</div></div></div></div>	Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC	1	Byte type		Byte type	2	Integer type		Integer type	3	Double-precision type		Double-precision type	4	Real type		Real type	5	Robot axis position type (pulse)		Robot axis position type (pulse)	6	Robot axis position type (XYZ)		Robot axis position type (XYZ)	7	Base axis position type (pulse)		External axis position type (pulse)	8	Base axis position type (XYZ)		External axis position type (XYZ)	9	Station axis position type (pulse)		-	10	String type	-	-
Variable Contents	FS100/DX100/ NX100 (v3.0 and after)	NX100 (before v3.0)/XRC/MRC	ERC																																										
1	Byte type		Byte type																																										
2	Integer type		Integer type																																										
3	Double-precision type		Double-precision type																																										
4	Real type		Real type																																										
5	Robot axis position type (pulse)		Robot axis position type (pulse)																																										
6	Robot axis position type (XYZ)		Robot axis position type (XYZ)																																										
7	Base axis position type (pulse)		External axis position type (pulse)																																										
8	Base axis position type (XYZ)		External axis position type (XYZ)																																										
9	Station axis position type (pulse)		-																																										
10	String type	-	-																																										

REMARKS

Variable type number and the storage area, number of values.

Variable Type Number	Storage area	Number of values
1 ~ 9	p	7
10	str	16

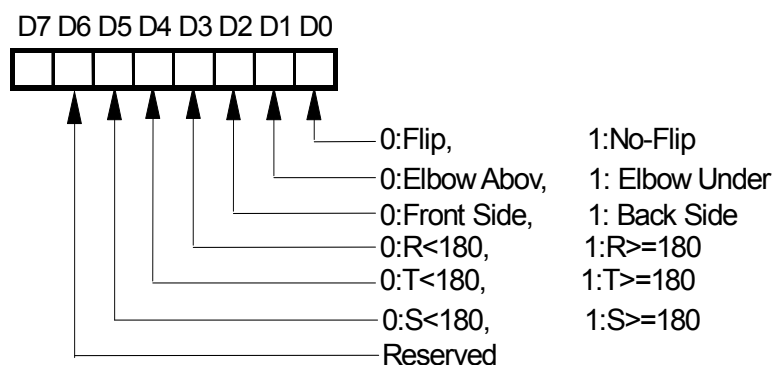


When this function is used to receive a string type variable make sure that the character variable storage area is allocated for 17 characters.

Declaration in Visual Basic: Dim S_Variable As String *17
Declaration in C++: char S_Variable[17]

Form

The form data are represented by bit data in decimals.



- * With the ERC or ERC2, the data from D3 to D7 are disregarded.
- * With the MRC or MRC2, the data D5 and D7 are disregarded.

Content of the numeric variable storage area

Depending on the variable type, the numeric variable storage area contains the number of values indicated below.

Variable Type Number	Number of values	Content						
		P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]
1	1	Byte	-	-	-	-	-	-
2	1	Integer	-	-	-	-	-	-
3	1	Double	-	-	-	-	-	-
4	1	Real	-	-	-	-	-	-
5	7	S-axis Pulses	L-axis Pulses	U-axis Pulses	R-axis Pulses	B-axis Pulses	T-axis Pulses	E-axis Pulses
6	7	X-axis (mm)	Y-axis (mm)	Z-axis (mm)	Wrist angle Rx (deg)	Wrist angle Ry (deg)	Wrist angle Rz (deg)	Re (deg)
7	6	Base Axis-1 Pulses	Base Axis-2 Pulses	Base Axis-3 Pulses	Base Axis-4 Pulses	Base Axis-5 Pulses	Base Axis-6 Pulses	-
8	6	Base Axis-1 (mm)	Base Axis-2 (mm)	Base Axis-3 (mm)	Base Axis-4 (mm)	Base Axis-5 (mm)	Base Axis-6 (mm)	-
9	6	Station Axis-1 Pulses	Station Axis-2 Pulses	Station Axis-3 Pulses	Station Axis-4 Pulses	Station Axis-5 Pulses	Station Axis-6 Pulses	-

Content of the character variable storage area

Variable Type Number	Number of values	Content
		str
10	16	String

CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscDCILoadSave" "BscDCILoadSaveOnce" "BscDCIGetPos" "BscDCIGetPos2" "BscDCIGetVarData" "BscDCIGetVarDataEx"

7.5 I/O Signal Read/Write Function

Reads or writes the I/O signals.

The following functions are available.

- BscReadIO
- BscReadIO2
- BscWriteIO
- BscWriteIO2

■ BscReadIO

FUNCTION	Reads specified count of coil status. Up to 256 coil numbers can be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReadIO(short nCid,short add,short num,short *stat);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number add Read starting address number num Number of read signals (up to 256) *stat Coil status</p> <p>OUT (Return) *stat Coil status</p> <p>Return Value -1 : Header number error 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the XRC/MRC. Refer to the BscReadIO2 for transmission with the FS100/DX100/NX100.</p> <p>Unnecessary Signals All unnecessary signals are set to 0 unless the number of the read data items is a multiple of 8.</p>

REMARKS	List of I/O Signals				
	MRC				
	Signal	Signal Range	Name	Read	Write
	0xxx	0010 to 0167 (128)	Robot universal input		x
	1xxx	1010 to 1167 (128)	Robot universal output		x
	2xxx	2010 to 2187 (144)	External input		x
	3xxx	3010 to 3187 (144)	External output		x
	4xxx	4010 to 4167 (128)	Robot specific input		x
	5xxx	5010 to 5247 (192)	Robot specific output		x
	6xxx	6010 to 6047 (32)	Timer/counter	x	x
	7xxx	7010 to 7327 (256)	Auxiliary relay		x
	8xxx	8010 to 8087 (64)	Control status signal		x
	82xx	8210 to 8247 (32)	Pseudo input signal		x
	9xxx	9010 to 9167 (128)	DL input		
	XRC				
	Signal	Signal Range	Name	Read	Write
	0xxx	0010 to 0247 (192)	Robot universal input		x
	1xxx	1010 to 1247 (192)	Robot universal output		x
	2xxx	2010 to 2327 (256)	External input		x
	3xxx	3010 to 3327 (256)	External output		x
	4xxx	4010 to 4287 (224)	Robot specific input		x
	5xxx	5010 to 5387 (304)	Robot specific output		x
	6xxx	-	-	x	x
	7xxx	7010 to 7887 (704)	Auxiliary relay		x
	8xxx	8010 to 8127 (96)	Control status signal		x
	82xx	8210 to 8247 (32)	Pseudo input signal		x
	9xxx	9010 to 9167 (128)	Network input		
CONTROLLER	XRC, MRC (Serial Port, Ethernet)				
REFERENCE	"BscWriteIO" "BscReadIO2" "BscWriteIO2"				

■ BscReadIO2

FUNCTION	Reads specified count of coil status. Up to 256 coil numbers can be specified.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReadIO2(short nCid,DWORD add,short num,short *stat);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number add Read starting address number num Number of read signals (up to 256) *stat Coil status</p> <p>OUT (Return)</p> <p>*stat Coil status</p> <p>Return Value</p> <p>-1 : Header number error 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions</p> <p>This function is effective only for transmission with the FS100/DX100/NX100. Refer to the BscReadIO for transmission with the XRC/ MRC.</p> <p>Unnecessary Signals</p> <p>All unnecessary signals are set to 0 unless the number of the read data items is a multiple of 8.</p>

REMARKS

List of I/O Signals
FS100

Signal	Signal Range	Name	Read	Write
0xxxx	00010 to 01287 (1024)	Robot universal input		x
1xxxx	10010 to 11287 (1024)	Robot universal output		x
2xxxx	20010 to 21287 (1024)	External input		x
25xxx	25010 to 26287 (1024)	Network input		
3xxxx	30010 to 31287 (1024)	External output		x
35xxx	35010 to 36287 (1024)	Network output		x
4xxxx	40010 to 41607 (1280)	Robot specific input		x
5xxxx	50010 to 52007 (1600)	Robot specific output		x
6xxxx	-	-	x	x
7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
8xxxx	80010 to 80647 (512)	Control status signal		x
82xxx	82010 to 82207 (160)	Pseudo input signal		x

DX100

Signal	Signal Range	Name	Read	Write
0xxxx	00010 to 02567 (2048)	Robot universal input		x
1xxxx	10010 to 12567 (2048)	Robot universal output		x
2xxxx	20010 to 22567 (2048)	External input		x
25xxx	25010 to 27567 (2048)	Network input		
3xxxx	30010 to 32567 (2048)	External output		x
35xxx	35010 to 37567 (2048)	Network output		x
4xxxx	40010 to 41607 (1280)	Robot specific input		x
5xxxx	50010 to 52007 (1600)	Robot specific output		x
6xxxx	-	-	x	x
7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
8xxxx	80010 to 80647 (512)	Control status signal		x
82xxx	82010 to 82207 (160)	Pseudo input signal		x

NX100

Signal	Signal Range	Name	Read	Write
0xxxx	00010 to 01287 (1024)	Robot universal input		x
1xxxx	10010 to 11287 (1024)	Robot universal output		x
2xxxx	20010 to 21287 (1024)	External input		x
22xxx	22010 to 23287 (1024)	Network input		
3xxxx	30010 to 31287 (1024)	External output		x
32xxx	32010 to 33287 (1024)	Network output		x
4xxxx	40010 to 40807 (640)	Robot specific input		x
5xxxx	50010 to 51007 (800)	Robot specific output		x
6xxxx	-	-	x	x
7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
8xxxx	80010 to 80647 (512)	Control status signal		x
82xxx	82010 to 82127 (96)	Pseudo input signal		x

CONTROLLER

FS100, DX100, NX100 (Serial Port, Ethernet)

REFERENCE

"BscWriteIO2" "BscReadIO" "BscWriteIO"

■ BscWriteIO

FUNCTION	Writes specified count of coil status. Up to 256 coil numbers can be specified. Address numbers to be written are only of Nos. 9000's.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscWriteIO(short nCid,short add,short num,short *stat);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number add Read starting address number num Number of read signals (up to 256) *stat Coil status</p> <p>OUT (Return) *stat Coil status</p> <p>Return Value -1 : Header number error 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the XRC/MRC. Refer to the BscWriteIO2 for transmission with the FS100/DX100/NX100.</p> <p>Unnecessary Signals All unnecessary data exist in the last data unless the number of the written data items is a multiple of 8.</p>

REMARKS	List of I/O Signals				
	MRC				
	Signal	Signal Range	Name	Read	Write
	0xxx	0010 to 0167 (128)	Robot universal input		x
	1xxx	1010 to 1167 (128)	Robot universal output		x
	2xxx	2010 to 2187 (144)	External input		x
	3xxx	3010 to 3187 (144)	External output		x
	4xxx	4010 to 4167 (128)	Robot specific input		x
	5xxx	5010 to 5247 (192)	Robot specific output		x
	6xxx	6010 to 6047 (32)	Timer/counter	x	x
	7xxx	7010 to 7327 (256)	Auxiliary relay		x
	8xxx	8010 to 8087 (64)	Control status signal		x
	82xx	8210 to 8247 (32)	Pseudo input signal		x
	9xxx	9010 to 9167 (128)	DL input		
	XRC				
	Signal	Signal Range	Name	Read	Write
	0xxx	0010 to 0247 (192)	Robot universal input		x
	1xxx	1010 to 1247 (192)	Robot universal output		x
	2xxx	2010 to 2327 (256)	External input		x
	3xxx	3010 to 3327 (256)	External output		x
	4xxx	4010 to 4287 (224)	Robot specific input		x
	5xxx	5010 to 5387 (304)	Robot specific output		x
	6xxx	-	-	x	x
	7xxx	7010 to 7887 (704)	Auxiliary relay		x
	8xxx	8010 to 8127 (96)	Control status signal		x
	82xx	8210 to 8247 (32)	Pseudo input signal		x
	9xxx	9010 to 9167 (128)	Network input		
CONTROLLER	XRC, MRC (Serial Port, Ethernet)				
REFERENCE	"BscReadIO" "BscReadIO2" "BscWriteIO2"				

■ BscWriteIO2

FUNCTION	Writes specified count of coil status. Up to 256 coil numbers can be specified. Address numbers to be written are only of Nos. 22000's (NX100) or Nos.25000's (FS100/DX100).
FORMAT	<code>_declspec(dllexport) short APIENTRY BscWriteIO2(short nCid,DWORD add,short num,short *stat);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number add Read starting address number num Number of read signals (up to 256) *stat Coil status</p> <p>OUT (Return) *stat Coil status</p> <p>Return Value -1 : Header number error 0 : Normal completion Others : Error codes</p>
REMARKS	<p>Restrictions This function is effective only for transmission with the FS100/DX100/NX100. Refer to the BscWriteIO for transmission with the XRC/ MRC.</p> <p>Unnecessary Signals All unnecessary data exist in the last data unless the number of the written data items is a multiple of 8.</p>

REMARKS	List of I/O Signals				
	FS100				
	Signal	Signal Range	Name	Read	Write
	0xxxx	00010 to 01287 (1024)	Robot universal input		x
	1xxxx	10010 to 11287 (1024)	Robot universal output		x
	2xxxx	20010 to 21287 (1024)	External input		x
	25xxx	25010 to 26287 (1024)	Network input		
	3xxxx	30010 to 31287 (1024)	External output		x
	35xxx	35010 to 36287 (1024)	Network output		x
	4xxxx	40010 to 41607 (1280)	Robot specific input		x
	5xxxx	50010 to 52007 (1600)	Robot specific output		x
	6xxxx	-	-	x	x
	7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
	8xxxx	80010 to 80647 (512)	Control status signal		x
	82xxx	82010 to 82207 (160)	Pseudo input signal		x
	DX100				
	Signal	Signal Range	Name	Read	Write
	0xxxx	00010 to 02567 (2048)	Robot universal input		x
	1xxxx	10010 to 12567 (2048)	Robot universal output		x
	2xxxx	20010 to 22567 (2048)	External input		x
	25xxx	25010 to 27567 (2048)	Network input		
	3xxxx	30010 to 32567 (2048)	External output		x
	35xxx	35010 to 37567 (2048)	Network output		x
	4xxxx	40010 to 41607 (1280)	Robot specific input		x
	5xxxx	50010 to 52007 (1600)	Robot specific output		x
	6xxxx	-	-	x	x
	7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
	8xxxx	80010 to 80647 (512)	Control status signal		x
	82xxx	82010 to 82207 (160)	Pseudo input signal		x
	NX100				
	Signal	Signal Range	Name	Read	Write
	0xxxx	00010 to 01287 (1024)	Robot universal input		x
	1xxxx	10010 to 11287 (1024)	Robot universal output		x
	2xxxx	20010 to 21287 (1024)	External input		x
	22xxx	22010 to 23287 (1024)	Network input		
	3xxxx	30010 to 31287 (1024)	External output		x
	32xxx	32010 to 33287 (1024)	Network output		x
	4xxxx	40010 to 40807 (640)	Robot specific input		x
	5xxxx	50010 to 51007 (800)	Robot specific output		x
	6xxxx	-	-	x	x
	7xxxx	70010 to 79997 (7992)	Auxiliary relay		x
	8xxxx	80010 to 80647 (512)	Control status signal		x
	82xxx	82010 to 82127 (96)	Pseudo input signal		x
CONTROLLER	FS100, DX100, NX100 (Serial Port, Ethernet)				
REFERENCE	"BscReadIO2" "BscReadIO" "BscWriteIO"				



7.6 Other Functions

The following functions are also available.

- BscClose
- BscCommand
- BscConnect
- BscDisconnect
- BscDiskFreeSizeGet
- BscEnforcedClose
- BscGets
- BscInBytes
- BscOpen
- BscOutBytes
- BscPuts
- BscReConnect
- BscReStartJob
- BscSetBreak
- BscSetCom
- BscSetCondBSC
- BscSetEServerMode
- BscSetEther
- BscStatus

■ BscClose

FUNCTION	Releases a communication handler.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscClose(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Failed to release
REMARKS	Call Condition It is necessary to disconnect the communications lines BscDisConnect function before calling this function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscOpen" "BscDisConnect"

■ BscCommand

FUNCTION	Sends a transmission command.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscCommand(short nCid,char *h_buf,char *d_buf,short fforever);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *h_buf Header character string pointer *d_buf Data character string pointer fforever Robot response; 0 : No wait; 1 : Wait</p> <p>OUT (Return) None</p> <p>Return Value -1 : Failed to send Others : Normal completion</p>
REMARKS	<p>Header Character String The header character string is represented by the header number and sub code number, in that order.</p> <p>Data Character String The data character string is represented by the data queue plus \r (carriage return) at the end.</p> <p><Example> When sending the "SERVO ON" command If Header number 01 Sub code number 000, then Header character string 01,000 Data character string SVON 1\r</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscConnect

FUNCTION	Connects communications lines.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscConnect(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Error 1 : Normal completion
REMARKS	Call Condition Before calling this function, it is necessary to set the communications lines with BscOpen function followed by BscSetCom function (serial port) or BscSetEther function (Ethernet) or BscSetEServer-Mode function (Ethernet Server).
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	" BscOpen " " BscSetCom "(Serial Port) " BscSetEther "(Ethernet) " BscDisconnect "

■ BscDisconnect

FUNCTION	Disconnects communications lines.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDisconnect(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Error 1 : Normal completion
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscClose" "BscConnect"

■ BscDiskFreeSizeGet

FUNCTION	Gets free capacity of the specified drive.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscDiskFreeSizeGet(short nCid,short dno,long *dsize);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number dno Drive number 1:A to 26:Z dsize Free capacity pointer</p> <p>OUT (Return) dsize Free capacity pointer</p> <p>Return Value 0 : Error 1 : Normal completion</p>
REMARKS	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscEnforcedClose

FUNCTION	Closes compulsorily.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscEnforcedClose(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion Others : Failed to release
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscOpen" "BscDisConnect"

■ BscGets

FUNCTION	Sends a character string by transmission in TTY level.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscGets(short nCid,char *bufptr,WORD bsize,WORD *plengets);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*bufptr Received character string pointer</p> <p>bsize Maximum character count</p> <p>plengets Received character count</p> <p>OUT (Return)</p> <p>*bufptr Received character string pointer</p> <p>Return Value</p> <p>Received character count</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscPuts" "BscInBytes" "BscOutBytes"

■ BscInBytes

FUNCTION	Returns the number of characters which are received by transmission in TTY level.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscInBytes(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value Received character count
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGets" "BscPuts" "BscOutBytes"

■ BscOpen

FUNCTION	Gets a communication handler.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscOpen(char *path,short mode);</code>
ARGUMENTS	<p>IN (Transfer) *path Communication current directory storage pointer mode Communication type : 1 (=0x01) : serial port, 16 (=0x10) : Ethernet 256 (=0x100) : Ethernet Server</p> <p>OUT (Return) None</p> <p>Return Value -1 : Acquisition Failure -8 : License Error Others : Communication handler ID number</p>
REMARKS	<p>Call Condition By calling the BscSetCom function (serial port) or BscSetEther (Ethernet) BscSetEServerMode(Ethernet Server) BscConnect function after calling this function, communications can be started.</p> <p>Type of Communications Only 1 (=0x01): serial port or 16 (=0x10): Ethernet or 256 (=0x100): Ethernet Server can be used. For any other values, an error occurs.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscClose" "BscSetCom"(Serial Port) "BscSetEther"(Ethernet) "BscSetEServerMode"(Ethernet Server) "BscConnect"

■ BscOutBytes

FUNCTION	Returns the remaining number of characters which are sent by transmission in TTY level.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscOutBytes(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value Sending character count
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGets" "BscPuts" "BscInBytes"

■ BscPuts

FUNCTION	Sends a character string by transmission in TTY level.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscPuts(short nCid,char *bufptr,WORD length);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number *bufptr Sending character string pointer length Sending character count
	OUT (Return) None
	Return Value Sending character count
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscGets" "BscInBytes" "BscOutBytes"

■ BscReConnect

FUNCTION	Connects communications lines again.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReConnect(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Error 1 : Normal completion
REMARKS	Call Condition Before calling this function, it is necessary to set the communications lines with the BscOpen function and BscSetCom function (serial port), or BscSetEther function (Ethernet) or BscSetEServer-Mode function (Ethernet Server).
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	" BscOpen " " BscSetCom "(Serial Port) " BscSetEther "(Ethernet) " BscDisconnect "

■ BscReStartJob

FUNCTION	Starts job again.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscReStartJob(short nCid);</code>
ARGUMENTS	IN (Transfer) nCid Communication handler ID number
	OUT (Return) None
	Return Value 0 : Normal completion 1 : Current job name not specified Others : Error codes
REMARKS	Call Condition The BscSelectJob function must be called up and the current job name must be specified before executing this function. To restart a job during startup that has been held by the BscHoldOn function, release the hold by BscHoldOff function to call up the BscContinueJob function.
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	"BscContinueJob" "BscSelectJob"

■ BscSetBreak

FUNCTION	Cancels transmission.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetBreak(short nCid,short flg);</code>
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number flg Forced completion flag ; 0 : No forced completion, 1 : Forced completion</p> <p>OUT (Return) None</p> <p>Return Value -1 : Communication handler error 0 : Normal completion</p>
REMARKS	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

■ BscSetCom

FUNCTION	Sets communications parameters of the serial port.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetCom(short nCid, short port,DWORD baud,short parity,short clen,short stp);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>port Communication port number 1 : COM1, 2 : COM2, 3 : COM3, 4 : COM4, ..., 255 : COM255</p> <p>baud Baud rate 150, 300, 600, 1200, 2400, 4800, 9600, 19200</p> <p>parity Parity 0: None, 1: Odd, 2: Even</p> <p>clen Data length 7: 7 bits, 8: 8 bits</p> <p>stp Stop bit 0 : 1 bit, 1 : 1.5 bits, 2 : 2 bits</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>0 : Error</p> <p>1 : Normal completion</p>
REMARKS	<p>Call Condition</p> <p>Before calling this function, it is necessary to get the communication handler of the serial port with BscOpen function. After calling this function, communications can be done using the BscConnect function.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC, ERC (Serial Port)
REFERENCE	"BscOpen" "BscConnect"

■ BscSetCondBSC

FUNCTION	Sets a communication control timer or retry counter.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetCondBSC(short nCid,short timerA,short timerB,short rtyR,short rtyW);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number timerA Timer A (control timer, unit: msec) timerB Timer B (text timer, unit: msec) rtyR Sequence retry counter rtyW Text retry counter</p> <p>OUT (Return)</p> <p>None</p> <p>Return Value</p> <p>-1 : Communication handler error 0 : Normal completion</p>
REMARKS	<p>Initial Value</p> <p>timerA 10000(msec) timerB 30000(msec) rtyR 3 rtyW 3</p> <p>Note: This function is used to change the parameters of MOTOCOM32 on the personal computer. To change the robot controller transmission parameters (control timers, retry counter), use the programming pendant of the robot controller.</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)

- BscSetEServerMode

FUNCTION	Sets the communication parameters for Ethernet Server function.
FORMAT	_declspec(dllexport) short APIENTRY BscSetEServerMode(short nCid, char *IPAddr, short Mode);
ARGUMENTS	<p>IN (Transfer) nCid Communication handler ID number *IPAddr IP address of receiver Mode Server communication mode 1: Server mode, -1: Exclusive mode</p> <p>OUT (Return) None</p> <p>Return Value 0 : Error 1 : Normal completion</p>
REMARKS	<p>Call Condition Before calling this function, it is necessary to get the communication handler of the serial port with BscOpen function. After calling this function, communications can be done using the BscConnect function.</p> <p>Mode Specification This function specifies the Ethernet Server mode. The following modes are available: Server Mode: The network connection between the controller and application is ended after each command. Because of this, multiple applications can communicate simultaneously to the same controller via the network connection. Exclusive Mode: The network connection with the controller is exclusive to a single application. After the BscConnect function is called, the network connection is maintained until the BscDisConnect function is called. Because the differences between these network connection modes are processed inside MOTOCOM32, it is not necessary to make any change to the connection method on application side, other than to appoint the mode with this function.</p> <p>Restrictions DCI function is not supported with Ethernet Server function communication. The function is only available for communication with the FS100/DX100/NX100.</p>
CONTROLLER	FS100, DX100, NX100 (Ethernet)
REFERENCE	"BscOpen" "BscConnect"

■ BscSetEther

FUNCTION	Sets parameters for Ethernet communications.										
FORMAT	<code>_declspec(dllexport) short APIENTRY BscSetEther(short nCid, char *IPAddr, short mode, HWND hWnd);</code>										
ARGUMENTS	<div><div>IN (Transfer) nCid Communication handler ID number IPAddr IP address of receiver mode Execution mode 0: Client, 1: Stand alone hWnd Window handle</div><div>OUT (Return) None</div><div>Return Value 0 : Error 1 : Normal completion</div></div>										
REMARKS	<div><div>Call Condition Before calling this function, it is necessary to get the communication handler of the serial port with BscOpen function. After calling this function, communications can be done using the BscConnect function.</div><div><div>Execution Mode and IP Address of Receiver Select the corresponding "mode" argument to the communications function to be used. That "mode" argument determines whether the application to be operated by personal computer is to be client or server.</div><table><tr><td>Function</td><td>Mode (Personal Computer)</td><td>IPAddr</td></tr><tr><td>Host Control</td><td>0 (Client)</td><td>Must be always set.</td></tr><tr><td>DCI</td><td rowspan="2">1 (Server)</td><td rowspan="2">Can be omitted.</td></tr><tr><td>Stand Alone</td></tr></table><div>When the personal computer is set to server (mode = 1), setting FS100/DX100/NX100/XRC/MRC IP address to the IP address (IPAddr) determines that the server is a specified client server.</div><div><Example> Client: BscSetEther(nCid, "192.168.10.10", 0, hWnd); Specified client server (IP address must be always written.): BscSetEther(nCid, "192.168.10.10", 1, hWnd); Some client servers (IP address is not written.): BscSetEther(nCid, "", 1, hWnd);</div></div></div>	Function	Mode (Personal Computer)	IPAddr	Host Control	0 (Client)	Must be always set.	DCI	1 (Server)	Can be omitted.	Stand Alone
Function	Mode (Personal Computer)	IPAddr									
Host Control	0 (Client)	Must be always set.									
DCI	1 (Server)	Can be omitted.									
Stand Alone											
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Ethernet)										
REFERENCE	"BscOpen" "BscConnect"										

■ BscStatus

FUNCTION	Reads the status.
FORMAT	<code>_declspec(dllexport) short APIENTRY BscStatus(short nCid,char *hpt,char *dpt,unsigned short sz,char *rbuf);</code>
ARGUMENTS	<p>IN (Transfer)</p> <p>nCid Communication handler ID number</p> <p>*hpt Header character string pointer</p> <p>*dpt Sending data character string pointer</p> <p>sz Sending data character string size</p> <p>*rbuf Received data character string pointer</p> <hr/> <p>OUT (Return)</p> <p>*rbuf Received data character string pointer</p> <hr/> <p>Return Value</p> <p>0 : Normal completion</p> <p>Others : Error codes</p>
CONTROLLER	FS100, DX100, NX100, XRC, MRC (Serial Port, Ethernet) ERC (Serial Port)
REFERENCE	" BscGetStatus "

7.7 DLL Functions Corresponding to Transmission-related Key words

7.7.1 DLL Functions Related to Transmission Commands

■ Read/Monitoring System

RALARM	BscGetError BscGetError2 BscGetFirstAlarm BscGetNextAlarm BsclsErrorCode
RPOS	BsclsLoc
RPOSJ	BsclsLoc BscGetPulsePos
RSTATS	BscGetStatus BsclsCycle BsclsServo BsclsTeachMode BsclsPlayMode BsclsRemoteMode BsclsHold BsclsAlarm BsclsError
RJSEQ	BsclsJobName BsclsJobLine BsclsJobStep
RPOSC	BsclsRobotPos BscGetCartPos
JWAIT	BscJobWait
RGROUP	BscGetCtrlGroupDX BsclsCtrlGroupDX BscGetCtrlGroupXrc BsclsCtrlGroupXrc BsclsTaskInfXrc BscGetCtrlGroup BsclsCtrlGroup BsclsTaskInf

■ Read/Data Access System

RJDIR	BscFindFirst BscFindFirstMaster BscFindNext BscFindNextMaster
RUFRAME	BscGetUFrame
UPLOAD	BscUpload BscUploadEx
SAVEV	BscGetVarData BscGetVarData2 BscHostGetVarData BscHostGetVarDataM BscGetVarDataEx

■ Operation System

HOLD	BscHoldOn BscHoldOff
RESET	BscReset
CANCEL	BscCancel
MODE	BscSelectMode
CYCLE	BscSelStepCycle BscSelOneCycle BscSelLoopCycle
HLOCK	BscOPLock BscOPUnLock
MDSP	BscMDSP
SVON	BscServoOn BscServoOff
CGROUP	BscSetCtrlGroupDX BscSetCtrlGroupXrc BscSetCtrlGroup
CTASK	BscChangeTask

■ Editing System

DELETE	BscDeleteJob
WUFRAME	BscPutUFrame BscPutUFrameEx2
CVTRJ	BscConvertJobP2R
DOWNLOAD	BscDownload BscDownloadEx
CVTSJ	BscConvertJobR2P
LOADV	BscPutVarData BscPutVarData2 BscHostPutVarData BscHostPutVarDataM BscPutVarDataEx

■ Job Selection System

SETMJ	BscSetMasterJob
JSEQ	BscSetLineNumber

■ Startup System

START	BscStartJob BscContinueJob
MOVJ	BscMovj BscMovjEx BscMov BscMovEx BscMovEx2
MOVL	BscMovl BscMovlEx BscMov BscMovEx BscMovEx2
IMOV	BscImov BscImovEx BscImovEx2 BscMov BscMovEx BscMovEx2

PMOVJ	BscPMovj BscPMovjEx BscPMov BscPMovEx
PMOVL	BscPMovl BscPMovlEx BscPMov BscPMovEx

■ Other DLL Functions

BscCommand
BscStatus

7.7.2 DLL Functions Related to DCI Function

LOADJ	BscDCILoadSave BscDCILoadSaveOnce
SAVEJ	BscDCILoadSave BscDCILoadSaveOnce
LOADV	BscDCIGetPos BscDCIGetPos2 BscDCIGetVarData BscDCIGetVarDataEx
SAVEV	BscDCIPutPos BscDCIPutPos2 BscDCIPutVarData BscDCIPutVarDataEx

7.7.3 DLL Functions Related to I/O Read/Write

I/O Read	BscReadIO BscReadIO2
I/O Write	BscWriteIO BscWriteIO2

7.7.4 DLL Functions Related to Personal Computer Communications Port

Port Connection	<code>BscOpen</code> <code>BscSetCom</code> <code>BscSetEServerMode</code> <code>BscSetEther</code> <code>BscConnect</code> <code>BscReConnect</code>
Port Disconnection	<code>BscClose</code> <code>BscDisConnect</code> <code>BscEnforcedClose</code>
Transmission Parameter Setting	<code>BscSetCondBSC</code>

7.7.5 Other DLL Functions

`BscDiskFreeSizeGet`
`BscGets`
`BscInBytes`
`BscOutBytes`
`BscPuts`
`BscSetBreak`

MOTOCOM32

OPERATION MANUAL

HEAD OFFICE

2-1 Kurosakishiroishi, Yahatanishi-ku, Kitakyushu 806-0004 Japan
Phone +81-93-645-7745 Fax +81-93-645-7746

YASKAWA America Inc. MOTOMAN Robotics Division
805 Liberty Lane, West Carrollton, OH 45449, U.S.A.
Phone +1-937-847-6200 Fax +1-937-847-6277

YASKAWA Nordic AB
Verkstadsгатan 2, PO Box 504, SE-385 25 Torsås, Sweden
Phone +46-480-417-800 Fax +46-486-414-10

YASKAWA Europe GmbH Robotics Div.
Kammerfeldstr. 1, 80591 Allershausen, Germany
Phone +49-8166-90-0 Fax +49-8166-90-103

YASKAWA Electric Korea Co., Ltd
9F, KYOBO Securities Bldg., 26-4, Yeoido-Dong Yeoungpo-ku, Seoul, KOREA
Phone +82-2-784-7844 Fax +82-2-784-8495

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.
252/246, 4th Floor. Muang Thai-Phatra Office Tower II Rachadaphisek Road, Huaykwang Bangkok, 10320 Thailand
Phone +66-2-693-2200 Fax +66-2-693-4200

Shougang MOTOMAN Robot Co. Ltd.
No.7, Yongchang-North Road, Beijing E&T Development Area, China 100176
Phone +86-10-6788-0548 Fax +86-10-6788-0548-813

YASKAWA ELECTRIC (SHANGHAI) Co., Ltd.
No.18Xizang Zhong Road, 17F, Harbour Ring Plaza, Shanghai 200001, CHINA
Phone +86-21-5385-0655 Fax +86-21-5385-2770

YASKAWA Robotics India Ltd.
#426, Udyog Vihar, Phase- IV, Gurgaon, Haryana, India
Phone +91-124-475-8500 Fax +91-124-414-8016

Specifications are subject to change without notice
for ongoing product modifications and improvements.

YASKAWA ELECTRIC CORPORATION



© Printed in Japan May 2008 08-03