NX100 OPTIONS
INSTRUCTIONS

FOR ETHERNET SERVER FUNCTION:
ADVANCED FEATURE OF NX100 ETHERNET FUNCTION

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

MOTOMAN INSTRUCTIONS

MOTOMAN-□□□ INSTRUCTIONS
NX100 INSTRUCTIONS
NX100 OPERATOR'S MANUAL
NX100 MAINTENANCE MANUAL

The NX100 operator's manual above corresponds to specific usage.
Be sure to use the appropriate manual.

YASKAWA

MANUAL NO.

HW0483202

<table>
<tr><td align="center">❗ MANDATORY</td></tr>
</table>

- **This manual explains the Ethernet server function of the NX100 system. Read this manual carefully and be sure to understand its contents before handling the NX100.**

- **General items related to safety are listed in the Section 1: Safety of the NX100 Instructions. To ensure correct and safe operation, carefully read the NX100 Instructions before reading this manual.**

<table>
<tr><td align="center">⚠ CAUTION</td></tr>
</table>

- **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.**

ii

Notes for Safe Operation

Read this manual carefully before installation, operation, maintenance, or inspection of the NX100.

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.

> **NOTE** To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as "CAUTION" and "WARNING".

iii

⚠ WARNING

- **Before operating the manipulator, check that servo power is turned OFF when the emergency stop buttons on the front door of the NX100 and programming pendant are pressed.**
  When the servo power is turned OFF, the SERVO ON LED on the programming pendant is turned OFF.

  Injury or damage to machinery may result if the emergency stop circuit cannot stop the manipulator during an emergency. The manipulator should not be used if the emergency stop buttons do not function.
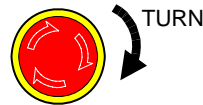
  Emergency Stop Button

- **Once the emergency stop button is released, clear the cell of all items which could interfere with the operation of the manipulator. Then turn the servo power ON**

  Injury may result from unintentional or unexpected manipulator motion.

  TURN

  Release of Emergency Stop

- **Observe the following precautions when performing teaching operations within the P-point maximum envelope of the manipulator:**
  - View the manipulator from the front whenever possible.
  - Always follow the predetermined operating procedure.
  - Ensure that you have a safe place to retreat in case of emergency.

  Improper or unintended manipulator operation may result in injury.

- **Confirm that no person is present in the P-point maximum envelope of the manipulator and that you are in a safe location before:**
  - Turning ON the NX100 power
  - Moving the manipulator with the programming pendant
  - Running the system in the check mode
  - Performing automatic operations

  Injury may result if anyone enters the P-point maximum envelope of the manipulator during operation. Always press an emergency stop button immediately if there is a problem. The emergency stop buttons are located on the right of the front door of the NX100 and the programming pendant.

<div style="border: 1px solid;">

⚠ CAUTION

- **Perform the following inspection procedures prior to conducting manipulator teaching. If problems are found, repair them immediately, and be sure that all other necessary processing has been performed.**
  -Check for problems in manipulator movement.
  -Check for damage to insulation and sheathing of external wires.

- **Always return the programming pendant to the hook on the NX100 cabinet after use.**

  The programming pendant can be damaged if it is left in the manipulator's work area, on the floor, or near fixtures.

- **Read and understand the Explanation of the Warning Labels in the NX100 Instructions before operating the manipulator.**

</div>

Definition of Terms Used Often in This Manual

The MOTOMAN manipulator is the YASKAWA industrial robot product.
The manipulator usually consists of the controller, the programming pendant, and supply cables.
In this manual, the equipment is designated as follows.

Equipment	Manual Designation
NX100 Controller	NX100
NX100 Programming Pendant	Programming Pendant
Cable between the manipulator and the controller	Manipulator Cable

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

Equipment		Manual Designation
Programming Pendant	Character Keys	The keys which have characters printed on them are denoted with [ ]. ex. [ENTER]
	Symbol Keys	The keys which have a symbol printed on them are not denoted with [ ] but depicted with a small picture. ex. page key The cursor key is an exception, and a picture is not shown.
	Axis Keys Numeric Keys	"Axis Keys" and "Numeric Keys" are generic names for the keys for axis operation and number input.
	Keys pressed simultaneously	When two keys are to be pressed simultaneously, the keys are shown with a "+" sign between them, ex. [SHIFT]+[COORD]
	Displays	The menu displayed in the programming pendant is denoted with { }. ex. {JOB}

Description of the Operation Procedure

In the explanation of the operation procedure, the expression "Select • • • " means that the cursor is moved to the object item and the SELECT key is pressed.

Registered Trademarks

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

vi

1 Introduction

2 System Configuration

3 Transmission Procedure

4 Function Checkout Procedure

5 Restrictions

1 Introduction

The Ethernet server function enables communication using more simple protocols instead of using conventional protocols such as BSC protocols, in the host control of the NX100 Ethernet function.
The Ethernet server function is an option function: advanced feature of the NX100 Ethernet function.
This manual explains settings and related information which are necessary for the use of this function.

1.1 Features

1.1.1 Request of Multiple Processes

The Ethernet server function manages multiple connections, which allows to perform sequential processing after receiving a multiple-processing request.

> **NOTE** The Ethernet server function performs sequential processing every time it receives a request. This function cannot handle a simultaneous parallel processing of multiple requests.

1.1.2 Simplification of Communication Procedure

The Ethernet server function uses TCP for the lower layer communication protocol. Using of the connection control function of TCP simplifies connection control processing in the upper layer and communication procedure.

1.1.3 Communication with Other Than MOTOCOM

The communication procedure for the Ethernet server function is simplified, which enables users to prepare for all communication programs.

> **NOTE** The communication procedure with the conventional NX100 Ethernet function was complicated; the procedure had not been disclosed. For this reason, the application "Motocom.dll" that mainly preforms communication procedure had been a must for the personal computer.

> **NOTE** The latest MOTOCOM allows the Ethernet server function to communicate in the system.

1-1

1.2 System Outline

The Ethernet server function is an option function: advanced feature of the host control function for the NX100 Ethernet function. When the Ethernet server function is used, the system needs to be preliminary kept in such a state that the NX100 Enthernet function can be used.

1.3 Applicable Version

The Ethernet server function can be used for the standard version NX100 with the system No. NS3.30.00A (US)-00 or greater.
For the NX100 other than standard version, contact your Yaskawa representative to see if the Ethernet server function can be used for the NX100 other than standard version.

2 System Configuration

Configure the following for the use of Ethernet server function.

2.1 Setting of Ethernet Function

The Ethernet server function is designed as an option function: advanced feature of the host control function for the NX100 Ethernet function. Therefore users need to configure the system in such a state that the NX100 Ethernet host control function can be used.
For details, refer to "3 Ethernet Function Settings" of "NX100 OPTIONS INSTRUCTIONS FOR ETHERNET FUNCTION (FOR NS3.00 OR GREATER)".

2.2 Setting of Ethernet Server Function

Set the parameters to enable the Ethernet server function.
(Users should not change the parameter settings themselves: contact your Yaskawa representative.)

2.3 Setting of Transmission Parameters

The parameters related to transmissions with the Ethernet server function are listed in the following tables.
For details, refer to "NX100 OPTIONS INSTRUCTIONS FOR DATA TRANSMISSION FUNCTION".

Make sure to change the value of the following parameter.

Parameter No.	Contents	Set Value	Default
RS000	Specifies the protocol for the standard port.	2 (BSC)	3 (FC1)

2.3 Setting of Transmission Parameters

Set the following parameters to their default values.

Parameter No.	Contents	Default
RS029	Loading of job/variable during playback	1: Enable
RS030	Specifies the data bits.	8: 8 bits
RS031	Specifies the stop bits.	0: 1 bit
RS032	Specifies the parity.	2: Even parity
RS033	Specifies the baud rate.	7: 9600 bauds
RS036	Specifies the enquiry character retry count for invalid responses or nonresponses.	10: 10 counts
RS037	Specifies the text data transmission retry count for block check error (NAK reception)	3: 3 counts
RS038	Specifies the block check method.	0: Checksum

Adjust the following parameters to stabilize the transmissions with the Ethernet server function as required.

Parameter No.	Contents		Set Value	Default
RS034	Timer A:	Timer for monitoring the sequence. Controls the response waiting time for the invalid responses or nonresponses.	100	30
	Units:	0.1 sec (setting range: 0 to 100)		
RS035	Timer B:	Timer for monitoring text reception. Controls the monitoring time to wait for the text termination character.	200	200
	Units:	0.1 sec (setting range: 0 to 255)		

3 Transmission Procedure

> **NOTE** In the Ethernet server function, processings other than socket connection are performed by the transmission of ASCII character string. The processings are not correctly performed if the ASCII character string is not in accordance with the ASCII code. Use case-sensitive ASCII character string.

3.1 Outline

The command transmission with the NX100 Ethernet server function proceeds as follows.

1. Connects a socket on the TCP port 80 of the NX100 from the host computer.
2. The host computer sends a START request.
3. After the START request is accepted, the NX100 sends the response to the START request.
4. The host computer sends a command.
5. The NX100 sends the response to the command sent from the host computer.
6. The host computer sends command data if required.
7. The NX100 sends an answer.
8. Closes a TCP socket from the host computer.
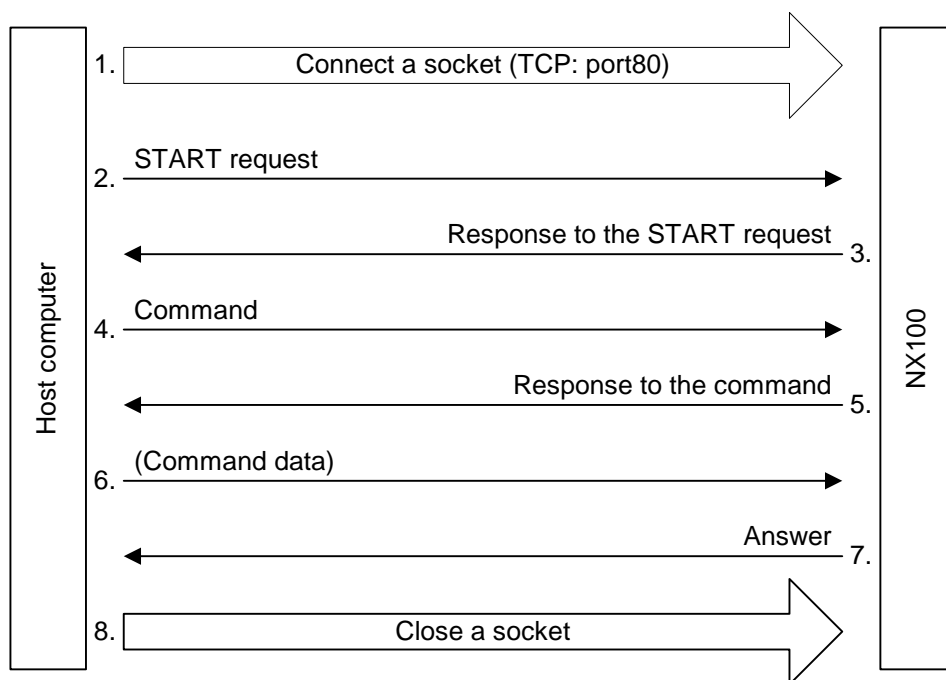
Outline of Command Transmission by Ethernet Server Function

3.1.1   Socket Connection

Connects a socket on the TCP port 80 of the NX100 from the host computer.

3.1.2   START Request

The host computer sends a character string of a START request: "CONNECT Robot_access<CR><LF>".
This character string allows for a single command processing.
If multiple commands need to be processed in a row, the host computer sends a character string: "CONNECT Robot_access Keep-Alive:n<CR><LF>". The following integer values can be specified in n.

   2 to 32767   : Executes as many commands as the number specified in n.

   -1   : Executes infinite number of commands.

3.1.3   Response to START Request

After receiving a START request from the host computer, the NX100 sends back a character string for the response to the request.
If the NX100 normally receives a START request, it sends a character string: "OK: NX Information Server(Ver).<CR><LF>" for a single command processing and "OK: NX Information Server(Ver) Keep-Alive:n.<CR><LF>" for processing of multiple commands. In each case the NX100 waits for reception of "3.1.4 Command" after sending back a character string.
If the NX100 cannot normally receive a START request, it sends back a character string: "NG: *HTTP Error Response*<CR><LF>", and closes a socket.

3.1.4   Command

The host computer sends a character string: "HOSTCTRL_REQUEST *Command Size*<CR><LF>".

■   *Command*

Use the character strings listed in "3.2 Command Details" for *Command*.

■   *Size*

For *Size*, use the ASCII character string which represents the byte number in decimal form, of character string that is sent in "3.1.6 Command Data". This byte number is calculated including the linefeed code <CR> at the end of the character string. If no command data are sent, writes "0" in *Size*.

3.1.5 Response to Command

After receiving a command from the host computer, the NX100 sends back a character string for the response to the command.

The NX100 normally sends back a character string: "OK: *Command*<CR><LF>". Furthermore, if the *Size* specified in " 3.1.4 Command " is "0", the NX100 responses according to "3.1.7 Response". If the *Size* is other than "0", the NX100 waits for " 3.1.6 Command Data ". At an error, the NX100 sends back a character string: "NG: *Message*<CR><LF>" and closes a socket.

3.1.6 Command Data

If a command is accompanied by data, the host computer sends a character string of the command data. The command data differ from *Command* to *Command*. (*Command* is not always accompanied by command data.) Make sure to add the linefeed code <CR> to the end of the command data character string.

The description method of command data differs with each *Command*. Refer to "3.2 Command Details" for details on the description method.

3.1.7 Answer

The NX100 sends back an answer in the following format according to a request from the host computer.

Classification	Subclassification	Format
At normal answer	In case that the NX100 sends back *data*	Data<CR> (Refer to "3.2 Command Details).
	In case that the NX100 answers normally (In the absence of *data*)	0000<CR><LF>
At abnormal answer	In case that the NX100 sends back an error answer	ERROR:[*Command*] is not successful (*xxxx*).<CR><LF> * *xxxx* indicates an interpreter message.
At socket error	In case that an error occurs at socket processing	ERROR:*Message*<CR><LF>

NOTE

The linefeed code of when *data* are sent back as normal answer is different from the one of the case other than normal answer with *data* sent back.
- The linefeed code of the normal answer with *data* sent back: <CR>
- The linefeed code of the case other than above: <CR><LF>

3-3

> SUPPLE-
> MENT
>
> The format when *data* are sent back as normal answer varies from each command. For details in the format, refer to "3.2 Command Details". For interpreter message, refer to "NX100 OPTIONS INSTRUCTIONS FOR DATA TRANSMISSION FUNCTION" (Manual No. RE-CKI-A445).

3.1.8   Closing of a Socket

A socket is closed.

3-4

HW0483202

3.2   Command Details

The following show how to use each command.
The answer for each command is the one at normal answer.
For details on robot control function, commands for multi-control group and independent control functions, and interpreter message list, refer to "NX100 OPTIONS INSTRUCTIONS FOR DATA TRANSMISSION FUNCTION" (Manual No. RE-CKI-A445).

3.2.1   Status Read Function

■   Read/Monitor Command

RALARM

Reads the error alarm code.

Command format: RALARM

Command data: None

Answer format: Data-1, Data-2, Data-3, Data-4, ···, Data-10
Data-1 = Error code (0 to 9999)
Data-2 = Error data (0 to 256)
Data-3 = Alarm code (0 to 9999)
Data-4 = Alarm data (0 to 256)
Data-5 = Alarm code (0 to 9999)
Data-6 = Alarm data (0 to 256)
Data-7 = Alarm code (0 to 9999)
Data-8 = Alarm data (0 to 256)
Data-9 = Alarm code (0 to 9999)
Data-10 = Alarm data (0 to 256)

RPOSJ

Reads the current position in joint coordinate system.

Command format: RPOSJ

Command data: None

Answer format: Data-1, Data-2, Data-3, Data-4, ···, Data-12
Data-1 = Number of S-axis pulses
Data-2 = Number of L-axis pulses
Data-3 = Number of U-axis pulses

Data-4 = Number of R-axis pulses
Data-5 = Number of B-axis pulses
Data-6 = Number of T-axis pulses
Data-7 = Number of 7th axis pulses
Data-8 = Number of 8th axis pulses
Data-9 = Number of 9th axis pulses
Data-10 = Number of 10th axis pulses
Data-11 = Number of 11th axis pulses
Data-12 = Number of 12th axis pulses

RPOSC

Reads the current position in a specified coordinate system. The specification with or without external axis can be made.

Command format: RPOSC

Command data: Data-1, Data-2
Data-1 = Specification of coordinate system
  0: Base coordinate
  1: Robot coordinate
  2: User coordinate 1
        .
        .
        .
  25: User coordinate 24

Data-2 = With or Without external axis
  0: Without external axis
  1: With external axis

Answer format: Data-1, Data-2, Data-3, Data-4, ···, Data-14
Data-1 = X coordinate value (unit: mm, significant 3 decimal points)
Data-2 = Y coordinate value (unit: mm, significant 3 decimal points)
Data-3 = Z coordinate value (unit: mm, significant 3 decimal points)
Data-4 = Wrist angle TX (unit: degree (°), significant 2 decimal point)
Data-5 = Wrist angle TY (unit: degree (°), significant 2 decimal point)
Data-6 = Wrist angle TZ (unit: degree (°), significant 2 decimal point)
Data-7 = Type
Data-8 = Tool number (0 to 23)
Data-9 = Numter of 7th axis pulses (for travel axis, mm)
Data-10 = Number of 8th axis pulses (for travel axis, mm)
Data-11 = Number of 9th axis pulses (for travel axis, mm)
Data-12 = Number of 10th axis pulses
Data-13 = Number of 11th axis pulses
Data-14 = Number of 12th axis pulses

3-6

• Data-9 to Data-14 are added only when "With external axis" is specified.
• If the specified user coordinate system is undefined, an error answer is sent back.
• Data of the type are represented by the following bit data coded into a decimal number.

```
        D5 D4 D3 D2 D1 D0
       ┌──┬──┬──┬──┬──┬──┐
       │  │  │  │  │  │  │
       └──┴──┴──┴──┴──┴──┘
         ▲  ▲  ▲  ▲  ▲  ▲
         │  │  │  │  │  └── 0 : Flip        1 : No flip
         │  │  │  │  └───── 0 : Upper arm   1 : Lower arm
         │  │  │  └──────── 0 : Front       1 : Back
         │  │  └─────────── 0 : R<180       1 : R>=180
         │  └────────────── 0 : T<180       1 : T>=180
         └───────────────── 0 : S<180       1 : S>=180
```
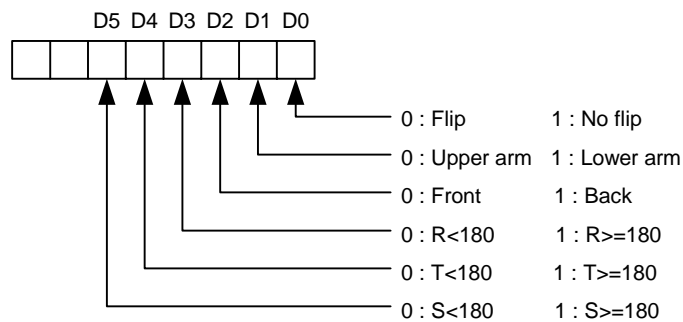
RSTATS

Reads the status of mode, cycle, operation, alarm error, and servo.

Command format: RSTATS

Command data: None

Answer format: Data-1, Data-2,

Data-1 = Represented by the following bit data coded into a decimal number

```
        D7 D6 D5 D4 D3 D2 D1 D0
       ┌──┬──┬──┬──┬──┬──┬──┬──┐
       │  │  │  │  │  │  │  │  │
       └──┴──┴──┴──┴──┴──┴──┴──┘
         ▲  ▲  ▲  ▲  ▲  ▲  ▲  ▲
         │  │  │  │  │  │  │  └── Step
         │  │  │  │  │  │  └───── 1 cycle
         │  │  │  │  │  └──────── Auto
         │  │  │  │  └─────────── Running
         │  │  │  └────────────── Safety speed operation
         │  │  └───────────────── Teach
         │  └──────────────────── Play
         └─────────────────────── Command remote
```

3-7
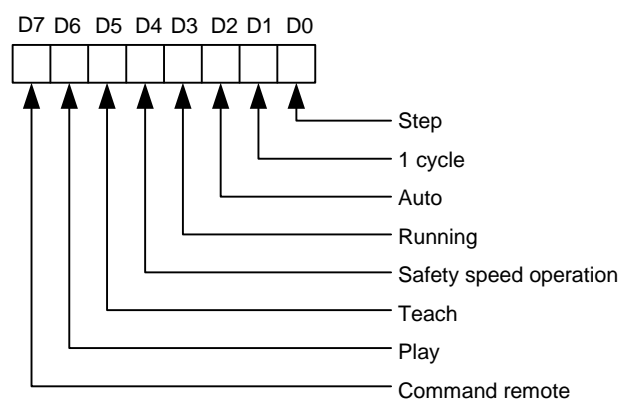
3.2 Command Details

Data-2 = Represented by the following bit data coded into a decimal number

D6 D5 D4 D3 D2 D1

Hold (programming pendant)
Hold (external)
Hold (command)
Alarm occuring
Error occuring
Servo ON
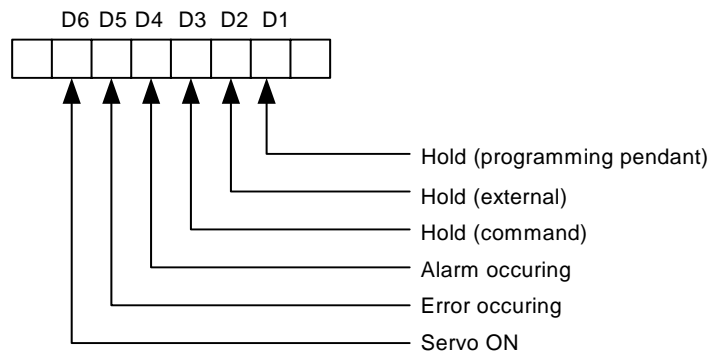
RJSEQ

Reads the current job name, line No. and step No.

Command format: RJSEQ

Command data: None

Answer format: Data-1, Data-2, Data-3

Data-1 = Read job name
Data-2 = Read line No. (0 to 9999)
Data-3 = Read step No. (0 to 9999)

JWAIT

The Ethernet server function does not support the JWAIT command.

NOTE   Use the command: JWAIT in the NX100 Ethernet function.

RGROUP

Reads the current control group set by the CGROUP command or CTASK command, and the task selection status.

Command format: RGROUP

Command data: None

Answer format: Data-1, Data-2, Data-3

Data-1 = Robot control group information
   • Represented by the following bit data coded into a decimal number
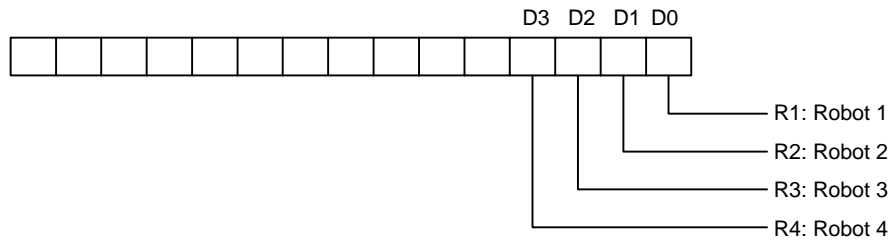
Data-2 = Station control group information
   • Represented by the following bit data coded into a decimal number
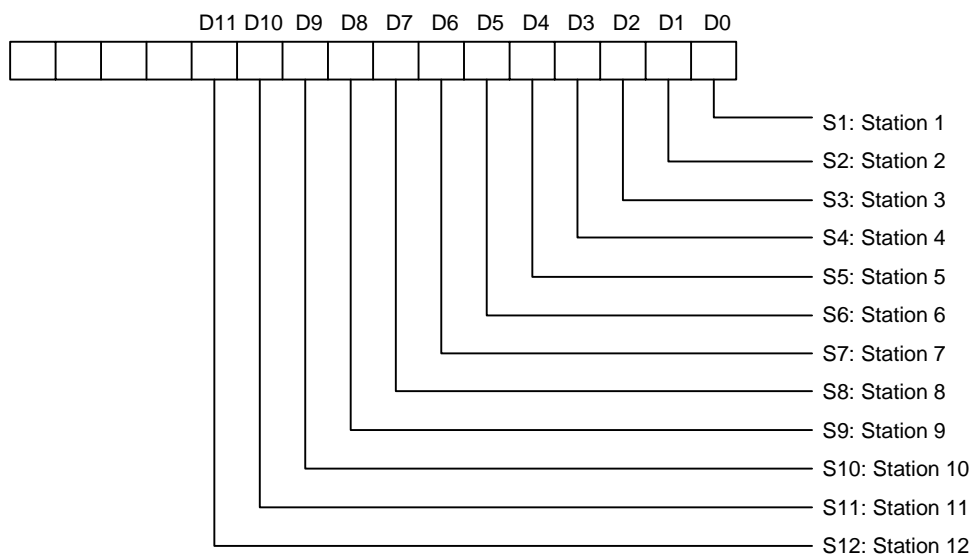
Data-3 = Task information
   In a system where independent control group is not allowed, "0" is returned.
   0: Master task
   1: Sub 1 task
   2: Sub 2 task
   3: Sub 3 task
   4: Sub 4 task
   5: Sub 5 task
   6: Sub 6 task
   7: Sub 7 task

3-9

■   Read/Data Access System Command

RJDIR

Reads all job names, or the names of jobs related to the parent job.

Command format: RJDIR

Command data: Job-Name
Job-Name = *
            = Parent job name
If Job-Name is specified with "*", RJDIR reads all the job names currently registered.
If Job-Name is specified with a parent job name, RJDIR reads the name of related jobs
excluding the parent job. If there is not related child jobs, the command returns the null list. If
the parent job has related child jobs but they are not registered in the system, an error occurs.

Answer format: Name-1, Name-2, ⋯, Name-N
Name-1 = Job name-1
Name-2 = Job name-2
            ·
            ·
   ·
Name-N = Job name-N

RUFRAME

Reads a specified user coordinate data.

Command format: RUFRAME

Command data: Data-1
Data-1 = User coordinate No.
        2: User coordinate 1
          ·
          ·
          ·
        25: User coordinate 24

Answer format: Data-1, Data-2, ⋯, Data-28
Data-1 = ORG X coordinate value (unit: mm, significant 3 decimal points)
Data-2 = ORG Y coordinate value (unit: mm, significant 3 decimal points)
Data-3 = ORG Z coordinate value (unit: mm, significant 3 decimal points)
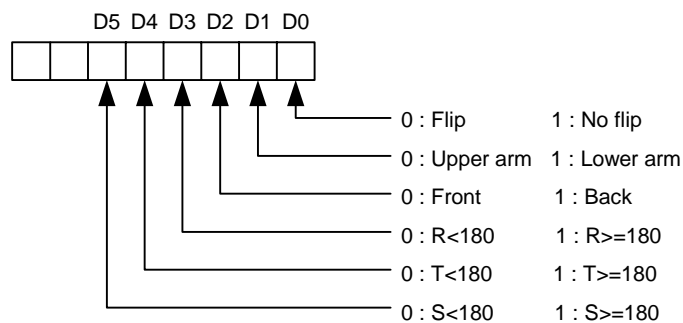Data-4 = ORG wrist angle TX (unit: degree (°), significant 2 decimal points)
Data-5 = ORG wrist angle TY (unit: degree (°), significant 2 decimal points)
Data-6 = ORG wrist angle TZ (unit: degree (°), significant 2 decimal points)
Data-7 = ORG type

Data-8 = XX X coordinate value (unit: mm, significant 3 decimal points)

Data-9 = XX Y coordinate value (unit: mm, significant 3 decimal points)

Data-10 = XX Z coordinate value (unit: mm, significant 3 decimal points)

Data-11 = XX wrist angle TX (unit: degree (°), significant 2 decimal points)

Data-12 = XX wrist angle TY (unit: degree (°), significant 2 decimal points)

Data-13 = XX wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-14 = XX type

Data-15 = XY X coordinate value (unit: mm, significant 3 decimal points)

Data-16 = XY Y coordinate value (unit: mm, significant 3 decimal points)

Data-17 = XY Z coordinate value (unit: mm, significant 3 decimal points)

Data-18 = XY wrist angle TX (unit: degree (°), significant 2 decimal points)

Data-19 = XY wrist angle TY (unit: degree (°), significant 2 decimal points)

Data-20 = XY wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-21 = XY type

Data-22 = Tool No. (0 to 23)

Data-23 = Number of 7th axis pulses (for travel axis, unit: mm)

Data-24 = Number of 8th axis pulses (for travel axis, unit: mm)

Data-25 = Number of 9th axis pulses (for travel axis, unit: mm)

Data-26 = Number of 10th axis pulses

Data-27 = Number of 11th axis pulses

Data-28 = Number of 12th axis pulses

- ORG, XX, YY coordinates are read in the base coordinate system.
- In a system having no external axis, Data-23 to Data-28 are "0".
- If the specified user coordinate system is not registered, an error answer is sent back.
- If the group axis of the specified user coordinate system is not R1, an error answer is sent back.
- If ORG, XX, and YY have different base axis data, an error answer is sent back.
- Data of the type are represented by the following bit data coded into a decimal number.


```
        D5  D4  D3  D2  D1  D0
       ┌───┬───┬───┬───┬───┬───┐
       │   │   │   │   │   │   │
       └───┴───┴───┴───┴───┴───┘
```

0 : Flip   1 : No flip

0 : Upper arm   1 : Lower arm

0 : Front   1 : Back

0 : R<180   1 : R>=180

0 : T<180   1 : T>=180

0 : S<180   1 : S>=180

3-11

SAVEV

Reads variable data.

Command format: SAVEV

Command data: Data-1, Data-2
Data-1 = Type of variables
        0: Byte type variables
        1: Integer type variables
        2: Double precision type variables
        3: Real number type variables
        4: Robot axis position type variables
        5: Base axis position type variables
        6: Station axis position type variables (only pulse type)
Data-2 = Variable No.

Answer format 1 (When the type of variables specified with the command data is 0, 1, 2, or 3)
Answer: Data-1
Data-1 = Byte value / Integer value / Double precision integer value / Real number value
      The value corresponding to the type of variables that is specified with the command
      data is read out.

Answer format 2 (When the type of variables specified with the command data is 4, 5, or 6)
Answer: Data-1, Data-2, ⋯, Data-10
Data-1 = Position data type (0: Pulse type, 1: Cartesian type)

(When the position data type is "0")
Data-2 = Number of robot S-axis pulses / Number of base 1st axis pulses / Number of station 1st axis pulses
Data-3 = Number of robot L-axis pulses / Number of base 2nd axis pulses / Number of station 2nd axis pulses
Data-4 = Number of robot U-axis pulses / Number of base 3rd axis pulses / Number of station 3rd axis pulses
Data-5 = Number of robot R-axis pulses / Number of base 4th axis pulses / Number of station 4th axis pulses
Data-6 = Number of robot B-axis pulses / Number of base 5th axis pulses / Number of station 5th axis pulses
Data-7 = Number of robot T-axis pulses / Number of base 6th axis pulses / Number of station 6th axis pulses
Data-8 = Tool No.
Data-9 = Not exist
Data-10 = Not exist

(When the position data type is "1": Only robot axis position type variables and base axis position type variables exist.)

Data-2 = Coordinate data

      0: Base coordinate

      1: Robot coordinate

      2: User coordinate 1

      3: User coordinate 2

        ·

        ·

        ·

      25: User coordinate 24

      26: Tool coordinate

      27: Master tool coordinate

Data-3 = X coordinate value / Base 1st Cartesian value (unit: mm, significant 3 decimal points)
Data-4 = Y coordinate value / Base 2nd Cartesian value (unit: mm, significant 3 decimal points)
Data-5 = Z coordinate value / Base 3rd Cartesian value (unit: mm, significant 3 decimal points)
Data-6 = Wrist angle TX coordinate value (unit: degree (°), significant 2 decimal points)
Data-7 = Wrist angle TY coordinate value (unit: degree (°), significant 2 decimal points)
Data-8 = Wrist angle TZ coordinate value (unit: degree (°), significant 2 decimal points)
Data-9 = Type

    • Data of the type are represented by the following bit data coded into a decimal number.

	D5	D4	D3	D2	D1	D0

0 : Flip   1 : No flip
0 : Upper arm 1 : Lower arm
0 : Front   1 : Back
0 : R<180   1 : R>=180
0 : T<180   1 : T>=180
0 : S<180   1 : S>=180

Data-10 = Tool No.

3-13
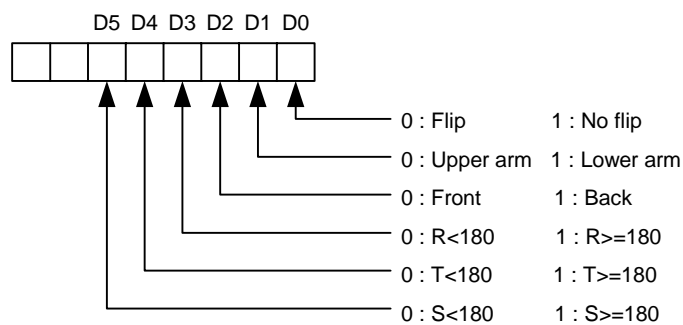
3.2.2   System Control Function

■   Operation System Command

HOLD

Turns HOLD ON/OFF.

Command format: HOLD

Command data: Data
Data = Specification of HOLD ON/OFF status
　　　　0: OFF
　　　　1: ON

Answer format: 0000 at normal answer

RESET

Resets an alarm of manipulator.
The transmission alarms can be reset only by the programming pendant.

Command format: RESET

Command data: None

Answer format: 0000 at normal answer

CANCEL

Cancels an error.

Command format: CANCEL

Command data: None

Answer format: 0000 at normal answer

MODE

Selects a mode.

Command format: MODE

Command data: Mode-No
Mode-No = Mode specification No.
      1: Teach mode
      2: Play mode

Answer format: 0000 at normal answer

> NOTE The MODE command can be used when the external mode switch is enabled on the OPERATING CONDITION window.

CYCLE

Selects cycle.

Command format: CYCLE

Command data: Cycle-No
Cycle-No = Cycle specification No.
      1: Step
      2: 1 cycle
      3: Auto

Answer format: 0000 at normal answer

SVON

Turns servo power supply ON/OFF.
To turn ON/OFF the servo power supply by this command, connect the external servo ON (EXSVON) signal 29 of the input terminal block (MTX) for the manipulator, to 30.

Command format: SVON

Command data: Data
Data = Specification of servo power supply ON/OFF status
      0: OFF
      1: ON

Answer format: 0000 at normal answer

3-15

3.2 Command Details

HLOCK

Sets an interlock between the programming pendant and I/O operation signals. While the interlock is ON, all operations except the following are prohibited.

- Emergency stop from the programming pendant
- Input signals except I/O mode change, external start, external servo ON, cycle change, I/O prohibited, PP/PANEL prohibited, and master call

HLOCK is invalid while the programming pendant is in the edit mode or accessing to a file for other function.

Command format: HLOCK

Command data: Data
Data = Interlock status setting
　　　　0: OFF
　　　　1: ON

Answer format: 0000 at normal answer

MDSP

Receives message data and displays the message in the remote display of the programming pendant. If the currently shown display is not the remote display, it is forcibly changed to the remote display to display the MDSP command message.

Command format: MDSP

Command data: Data
Data = Message to be displayed (Characters up to 30 bytes)

Answer format: 0000 at normal answer

CGROUP

Changes an objective control group of various commands used in the host control function. The NX100 can support multiple number of manipulators and stations. In this case, CGROUP is used when any control group for commands such as RPOSC is to be changed.
When the power supply is started up, robot 1, base 1, and station 1 (when a base and a stations exist) are specified.
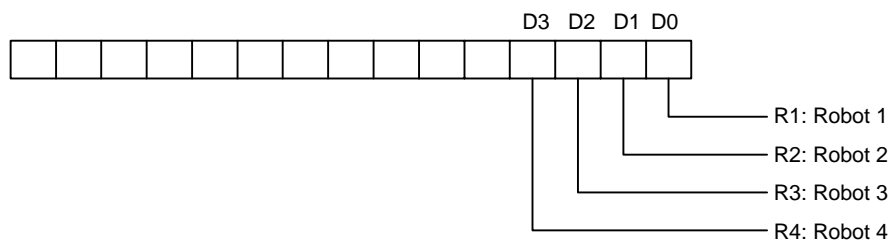
Command format: CGROUP

Command data: Data-1, Data-2

Data-1 = Robot control group specification

A control group can be specified according to the data shown below. However, the following settings cannot be made.

- Selection of control axis which does not exist
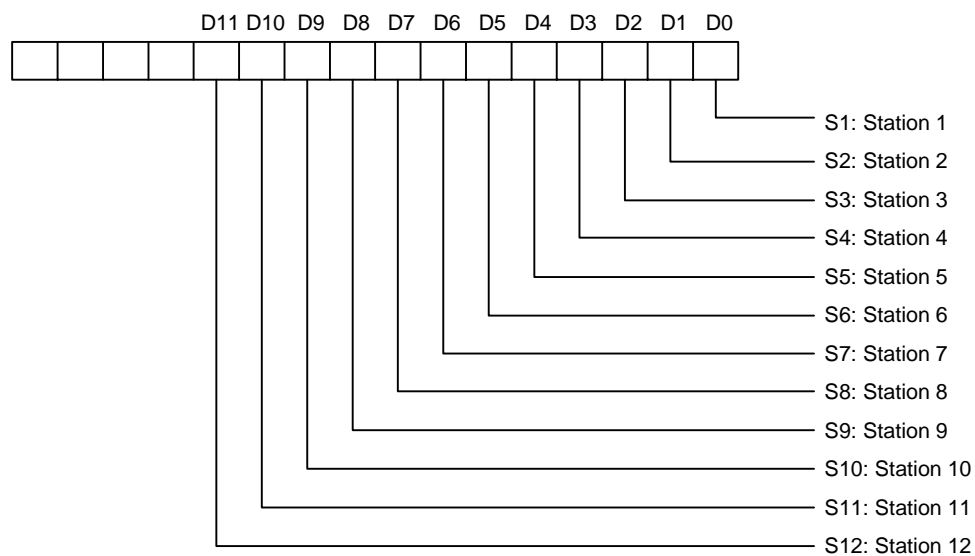- Specification of multiple robot axes

In a system with a base axis (such as travel axis), when the manipulator with this base axis is specified, this base axis is also automatically specified.

```
                                          D3  D2  D1  D0
┌──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┐
│  │  │  │  │  │  │  │  │  │  │  │  │  │  │
└──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┘
                                              └── R1: Robot 1
                                              └── R2: Robot 2
                                              └── R3: Robot 3
                                              └── R4: Robot 4
```

Data-2 = Station control group specification

A control group can be specified according to the following data. However, the following settings cannot be made.

- Selection of control axis which does not exist
- Specification of multiple station axes

```
        D11 D10 D9  D8  D7  D6  D5  D4  D3  D2  D1  D0
┌──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┬──┐
│  │  │  │  │  │  │  │  │  │  │  │  │  │  │  │
└──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┴──┘
                                              └── S1: Station 1
                                              └── S2: Station 2
                                              └── S3: Station 3
                                              └── S4: Station 4
                                              └── S5: Station 5
                                              └── S6: Station 6
                                              └── S7: Station 7
                                              └── S8: Station 8
                                              └── S9: Station 9
                                              └── S10: Station 10
                                              └── S11: Station 11
                                              └── S12: Station 12
```

Answer format: 0000 at normal answer

3-17

CTASK (OPTIONAL)

Changes the task for control in the host control function.
When the power supply is started up or in a system where an independent control is not allowed, CTASK is to be used as follows.

- When the power supply is started up, a master task is selected as a task to be controlled.
- CTASK cannot be used in a system where an independent control is not allowed.

Command format: CTASK

Command data: Data
Data = Specified task
        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

Answer format: 0000 at normal answer

■   Start-Up System Command

START

Starts a job.
If a job name is specified for an operand, the execution is started from the beginning of the job. If no job name is specified, the execution is started from the current line number of the set execution job.

Command format: START

Command data: Job-Name (Can be omitted.)
Job-Name = Starting job name

Answer format: 0000 at normal answer

MOVJ

Moves a manipulator to a specified coordinate position in joint motion.

Command format: MOVJ

Command data: Data-1, Data-2, ···, Data-16
Data-1 = Motion speed (0.01 to 100.0 %)
Data-2 = Coordinate specification
       0: Base coordinate
       1: Robot coordinate
       2: User coordinate 1
            ·
            ·
            ·
       25: User coordinate 24
Data-3 = X coordinate value (unit: mm, significant 3 decimal points)
Data-4 = Y coordinate value (unit: mm, significant 3 decimal points)
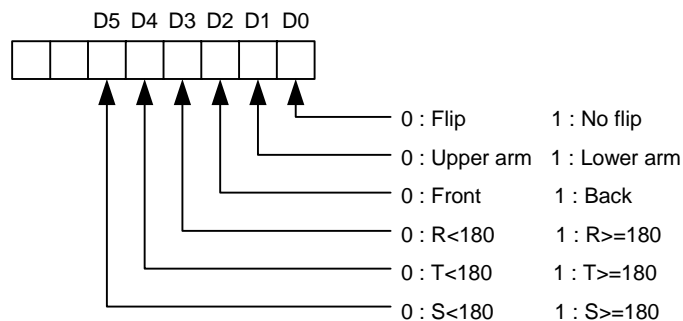Data-5 = Z coordinate value (unit: mm, significant 3 decimal points)
Data-6 = Wrist angle TX (unit: degree (°), significant 2 decimal points)
Data-7 = Wrist angle TY (unit: degree (°), significant 2 decimal points)
Data-8 = Wrist angle TZ (unit: degree (°), significant 2 decimal points)
Data-9 = Type

• Data of the type are represented by the following bit data coded into a decimal number.

	D5	D4	D3	D2	D1	D0

0 : Flip   1 : No flip
0 : Upper arm   1 : Lower arm
0 : Front   1 : Back
0 : R<180   1 : R>=180
0 : T<180   1 : T>=180
0 : S<180   1 : S>=180

Data-10 = Tool No. (0 to 23)
Data-11 = Number of 7th axis pulses (for travel axis, unit: mm)
Data-12 = Number of 8th axis pulses (for travel axis, unit: mm)
Data-13 = Number of 9th axis pulses (for travel axis, unit: mm)
Data-14 = Number of 10th axis pulses
Data-15 = Number of 11th axis pulses
Data-16 = Number of 12th axis pulses
    • In a system without external axis, Data-11 to Data-16 should be set to "0".
    • If a specified user coordinate is not defined, an error occurs.

Answer format: 0000 at normal answer

MOVL

Moves a manipulator to a specified coordinate position in linear motion.

Command format: MOVL

Command data: Data-1, Data-2, ···, Data-17
Data-1 = Motion speed selection
   0: V (speed)
   1: VR (posture speed)
Data-2 = Motion speed (0.1 to □□□. □□ mm/s, 0.1 to □□□. □° /s)
Data-3 = Coordinate specification
   0: Base coordinate
   1: Robot coordinate
   2: User coordinate 1
    ·
    ·
    ·
   25: User coordinate 24
Data-4 = X coordinate value (unit: mm, significant 3 decimal points)
Data-5 = Y coordinate value (unit: mm, significant 3 decimal points)
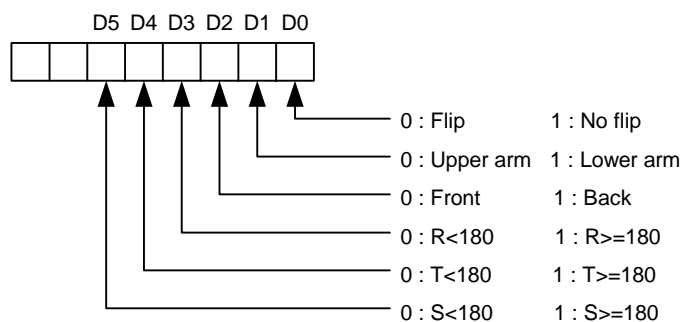Data-6 = Z coordinate value (unit: mm, significant 3 decimal points)
Data-7 = Wrist angle TX (unit: degree (°), significant 2 decimal points)
Data-8 = Wrist angle TY (unit: degree (°), significant 2 decimal points)
Data-9 = Wrist angle TZ (unit: degree (°), significant 2 decimal points)
Data-10 = Type
  • Data of the type are represented by the following bit data coded into a decimal number.

Data-11 = Tool No. (0 to 23)
Data-12 = Number of 7th axis pulses (for travel axis, unit: mm)
Data-13 = Number of 8th axis pulses (for travel axis, unit: mm)
Data-14 = Number of 9th axis pulses (for travel axis, unit: mm)
Data-15 = Number of 10th axis pulses
Data-16 = Number of 11th axis pulses
Data-17 = Number of 12th axis pulses
  • In a system without external axis, Data-12 to Data-17 should be set to "0".
  • If a specified user coordinate is not defined, an error occurs.

Answer format: 0000 at normal answer

IMOV

Moves a manipulator from the current position for a specified coordinate incremental value in linear motion.

Command format: IMOV

Command data: Data-1, Data-2, ···, Data-17
Data-1 = Motion speed selection
      0: V (speed)
      1: VR (posture speed)
Data-2 = Motion speed (0.1 to □□□. □□ mm/s, 0.1 to □□□. □° /s)
Data-3 = Coordinate specification
      0: Base coordinate
      1: Robot coordinate
      2: User coordinate 1
          ·
          ·
          ·
      25: User coordinate 24
      26: Tool coordinate
Data-4 = X coordinate incremental value (unit: mm, significant 3 decimal points)
Data-5 = Y coordinate incremental value (unit: mm, significant 3 decimal points)
Data-6 = Z coordinate incremental value (unit: mm, significant 3 decimal points)
Data-7 = Wrist angle TX incremental value (unit: degree (°), significant 2 decimal points)
Data-8 = Wrist angle TY incremental value (unit: degree (°), significant 2 decimal points)
Data-9 = Wrist angle TZ incremental value (unit: degree (°), significant 2 decimal points)
Data-10 = Reserved
Data-11 = Tool No. (0 to 23)
Data-12 = Number of 7th axis pulses (for travel axis, unit: mm)
Data-13 = Number of 8th axis pulses (for travel axis, unit: mm)
Data-14 = Number of 9th axis pulses (for travel axis, unit: mm)
Data-15 = Number of 10th axis pulses
Data-16 = Number of 11th axis pulses
Data-17 = Number of 12th axis pulses
    • In a system without external axis, Data-12 to Data-17 should be set to "0".
    • If a specified user coordinate is not defined, an error occurs.

Answer format: 0000 at normal answer

3.2 Command Details

PMOVJ

Moves a manipulator to a specified pulse position in joint motion.

Command format: PMOVJ

Command data: Data-1, Data-2, ···, Data-14
Data-1 = Motion speed (0.01 to 100.0 %)
Data-2 = Number of S-axis pulses
Data-3 = Number of L-axis pulses
Data-4 = Number of U-axis pulses
Data-5 = Number of R-axis pulses
Data-6 = Number of B-axis pulses
Data-7 = Number of T-axis pulses
Data-8 = Tool No. (0 to 23)
Data-9 = Number of 7th axis pulses
Data-10 = Number of 8th axis pulses
Data-11 = Number of 9th axis pulses
Data-12 = Number of 10th axis pulses
Data-13 = Number of 11th axis pulses
Data-14 = Number of 12th axis pulses
    • In a system without external axis, Data-9 to Data-14 should be set to "0".

Answer format: 0000 at normal answer

PMOVL

Moves a manipulator to a specified pulse position in linear motion.

Command format: PMOVL

Command data: Data-1, Data-2, ···, Data-15
Data-1 = Motion speed selection
        0: V (speed)
        1: VR (posture speed)
Data-2 = Motion speed (0.1 to □□□. □□ mm/s, 0.1 to □□□. □° /s)
Data-3 = Number of S-axis pulses
Data-4 = Number of L-axis pulses
Data-5 = Number of U-axis pulses
Data-6 = Number of R-axis pulses
Data-7 = Number of B-axis pulses
Data-8 = Number of T-axis pulses
Data-9 = Tool No. (0 to 23)
Data-10 = Number of 7th axis pulses
Data-11 = Number of 8th axis pulses
Data-12 = Number of 9th axis pulses

Data-13 = Number of 10th axis pulses

Data-14 = Number of 11th axis pulses

Data-15 = Number of 12th axis pulses

　　　• In a system without external axis, Data-10 to Data-15 should be set to "0".

Answer format: 0000 at normal answer

■　　Editing System Commands

DELETE

Deletes a specified job.

Command format: DELETE

Command data: Job-Name

Job-Name = *

　　　　　= Job name to be deleted

If Job-Name is specified with "*", the command DELETE deletes all jobs which are currently registered.

If Job-Name is specified with the job name to be deleted, the command DELETE deletes only the specified job.

Answer format: 0000 at normal answer

CVTRJ (Optional)

Converts a specified job to a relative job of a specified coordinate.

Command format: CVTRJ

Command data: Data-1, Data-2

Data-1 = Name of job to be converted

Data-2 = Conversion coordinate system specification

　　　　0: Base coordinate

　　　　1: Robot coordinate

　　　　2: User coordinate 1

　　　　　　·

　　　　　　·

　　　　　　·

　　　　25: User coordinate 24

　　　• If the specified user coordinate is not defined, an error answer is sent back.

Answer format: 0000 at normal answer

3-23

3.2 Command Details

NOTE The CVTRJ command can be used when the relative job function is enabled.

CVTSJ (Optional)

Converts a specified job to a standard job (pulse job) in a specified converting method.

Command format: CVTSJ

Command data: Data-1, Data-2, Data-3
Data-1 = Name of job to be converted
Data-2 = Converting method specification
       0: Previous step regarded (B-axis sign same)
       1: Type regarded
       2: Previous step regarded (R-axis travel amount minimum)
Data-3 = Reference position variable. Position variable No. indicating the first step conversion
      reference position when the previous step is regarded

Answer format: 0000 at normal answer

NOTE The CVTSJ command can be used when the relative job function is enabled.

WUFRAME

Writes a user coordinate data to a specified user coordinate system.

Command format: WUFRAME

Command data: Data-1, Data-2, ···, Data-29
Data-1 = User coordinate No.
      2: User coordinate 1
        ·
        ·
        ·
      25: User coordinate 24
Data-2 = ORG X coordinate value (unit: mm, significant 3 decimal points)
Data-3 = ORG Y coordinate value (unit: mm, significant 3 decimal points)
Data-4 = ORG Z coordinate value (unit: mm, significant 3 decimal points)
Data-5 = ORG wrist angle TX (unit: degree (°), significant 2 decimal points)

3-24

Data-6 = ORG wrist angle TY (unit: degree (°), significant 2 decimal points)

Data-7 = ORG wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-8 = ORG type

Data-9 = XX X coordinate value (unit: mm, significant 3 decimal points)

Data-10 = XX Y coordinate value (unit: mm, significant 3 decimal points)

Data-11 = XX Z coordinate value (unit: mm, significant 3 decimal points)

Data-12 = XX wrist angle TX (unit: degree (°), significant 2 decimal points)

Data-13 = XX wrist angle TY (unit: degree (°), significant 2 decimal points)

Data-14 = XX wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-15 = XX type

Data-16 = XY X coordinate value (unit: mm, significant 3 decimal points)

Data-17 = XY Y coordinate value (unit: mm, significant 3 decimal points)

Data-18 = XY Z coordinate value (unit: mm, significant 3 decimal points)

Data-19 = XY wrist angle TX (unit: degree (°), significant 2 decimal points)

Data-20 = XY wrist angle TY (unit: degree (°), significant 2 decimal points)

Data-21 = XY wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-22 = XY type

Data-23 = Tool Number (0 to 23)

Data-24 = Number of 7th axis pulses (for travel axis, unit: mm)

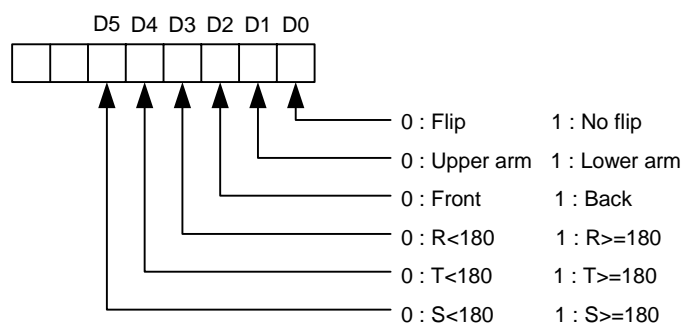Data-25= Number of 8th axis pulses (for travel axis, unit: mm)

Data-26 = Number of 9th axis pulses (for travel axis, unit: mm)

Data-27 = Number of 10th axis pulses

Data-28 = Number of 11th axis pulses

Data-29 = Number of 12th axis pulses

- ORG, XX, and XY coordinates are written in the base coordinate system.
- In a system without external axis, Data-24 to Data-29 should be set to "0".
- If the group axis of the specified user coordinate system is not R1, an error answer is sent back.
- For base-axis data of ORG, XX, and XY, the same data should be used.
- Data of the type are represented by the following bit data coded into a decimal number.


```
        D5  D4  D3  D2  D1  D0
      ┌───┬───┬───┬───┬───┬───┬───┐
      │   │   │   │   │   │   │   │
      └───┴───┴───┴───┴───┴───┴───┘
                            └─── 0 : Flip        1 : No flip
                        └─────── 0 : Upper arm   1 : Lower arm
                    └─────────── 0 : Front       1 : Back
                └─────────────── 0 : R<180       1 : R>=180
            └─────────────────── 0 : T<180       1 : T>=180
        └─────────────────────── 0 : S<180       1 : S>=180
```

Answer format: 0000 at normal answer

3-25

LOADV

Receives variable data from a host computer and writes it in a specified variable.

Command format: LOADV

Command data format 1: (When the type of variables specified with the command data is 0, 1, 2, or 3)
Command data: Data-1, Data-2, Data-3
Data-1 = Type of variables
        0: Byte type variables
        1: Integer type variables
        2: Double precision type variables
        3: Real number type variables
Data-2 = Variable No.
Data-3 = Byte value / Integer value / Double precision type integer value / Real number value
       The value corresponding to the type of variables that is specified in Data-1 is written in.

Command data format 2: (When the type of variables specified with the command data is 4, 5, or 6)
Command data: Data-1, Data-2, ⋯, Data-12
Data-1 = Type of variables
        4: Robot axis position type variables
        5: Base axis position type variables
        6: Station axis position type variables (only pulse type)
Data-2 = Variable No.
Data-3 = Position data type (0: Pulse type, 1: Cartesian type)

(When the position data type is 0)
Data-4 = Number of S-axis pulses / Number of base 1st axis pulses / Number of station 1st axis pulses
Data-5 = Number of L-axis pulses / Number of base 2nd axis pulses / Number of station 2nd axis pulses
Data-6 = Number of U-axis pulses / Number of base 3rd axis pulses / Number of station 3rd axis pulses
Data-7 = Number of R-axis pulses / Number of base 4th axis pulses / Number of station 4th axis pulses
Data-8 = Number of B-axis pulses / Number of base 5th axis pulses / Number of station 5th axis pulses
Data-9 = Number of T-axis pulses / Number of base 6th axis pulses / Number of station 6th axis pulses
Data-10 = Tool No.
Data-11 = Not exist
Data-12 = Not exist

(When the position data type is 1: Only robot axis position type variables / base axis position type variables exist.)
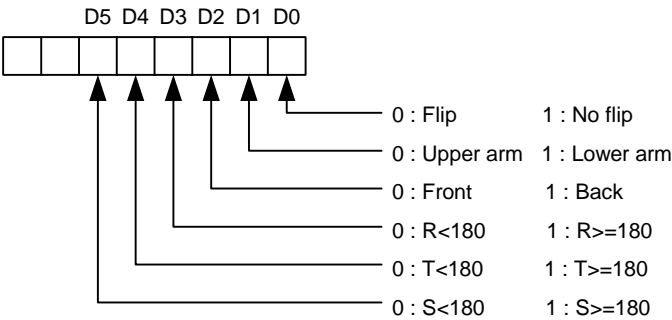Data-4 = Coordinate data
        0: Base coordinate
        1: Robot coordinate
        2: User coordinate 1
        3: User coordinate 2

 ·
 ·
 ·

  25: User coordinate 24

  26: Tool coordinate

  27: Master tool coordinate

Data-5 = X coordinate value / Base 1st axis Cartesian value (unit: mm, significant 3 decimal points)

Data-6 = Y coordinate value / Base 2nd axis Cartesian value (unit: mm, significant 3 decimal points)

Data-7 = Z coordinate value / Base 3rd axis Cartesian value (unit: mm, significant 3 decimal points)

Data-8 = Wrist angle TX (unit: degree (°), significant 2 decimal points)

Data-9 = Wrist angle TY(unit: degree (°), significant 2 decimal points)

Data-10 = Wrist angle TZ (unit: degree (°), significant 2 decimal points)

Data-11 = Type

  • Data of the type are represented by the following bit data coded into a decimal number.


```
        D5 D4 D3 D2 D1 D0
       ┌──┬──┬──┬──┬──┬──┐
       │  │  │  │  │  │  │
       └──┴──┴──┴──┴──┴──┘
         │  │  │  │  │  └─── 0 : Flip        1 : No flip
         │  │  │  │  └────── 0 : Upper arm   1 : Lower arm
         │  │  │  └───────── 0 : Front       1 : Back
         │  │  └──────────── 0 : R<180       1 : R>=180
         │  └─────────────── 0 : T<180       1 : T>=180
         └────────────────── 0 : S<180       1 : S>=180
```

Data-12 = Tool No.

Answer format: 0000 at normal answer

3.2 Command Details

■ Job Selection System Command

SETMJ

Sets a specified job as a master job.
At the same time, the specified job is set as an execution job.

Command format: SETMJ

Command data: Job-Name
Job-Name = Job name to be set

Answer format: 0000 at normal answer

JSEQ

Sets a job name and a line No.

Command format: JSEQ

Command data: Data-1, Data-2
Data-1 = Job name to be set
Data-2 = Line No. to be set

Answer format: 0000 at normal answer

3.2.3 I/O Read/Write Function

■ Read-Out of I/O Signal Status

Reads out I/O signals.

Command format: IOREAD

Command data: Data-1, Data-2
Data-1 = Contact point No. to start read-out
Data-2 = The number of contact points to be read out
- I/O data are output every eight contact points. Specify the number of contact points to be read out, in multiples of eight.

Answer format: Data-1, Data-2, ···, Data-N
Data-1 = Read-out data for the first eight contact points
Data-2 = Read-out data for the next eight contact points
.
.
.
Data-N = Read-out data for the last eight contact points
N = (Command data Data-2)/8. The command data Data-2 should be multiples of eight.

■ Write-in of I/O Signal Status

Writes in I/O signals.

Command format: IOWRITE

Command data: Data-1, Data-2, ···, Data-N
Data-1 = Contact point No. to start write-in
Data-2 = The number of contact points to be written in
Data-3 = Write-in data for the first eight contact points
Data-4 = Write-in data for the next eight contact points
.
.
.
Data-N = Write-in data for the last eight contact points
N = (Command data Data-2)/8+2. The command data Data-2 should be multiples of eight.

- I/O data are processed every eight contact points. Specify the number of contact points to be written-in, in multiples of eight.
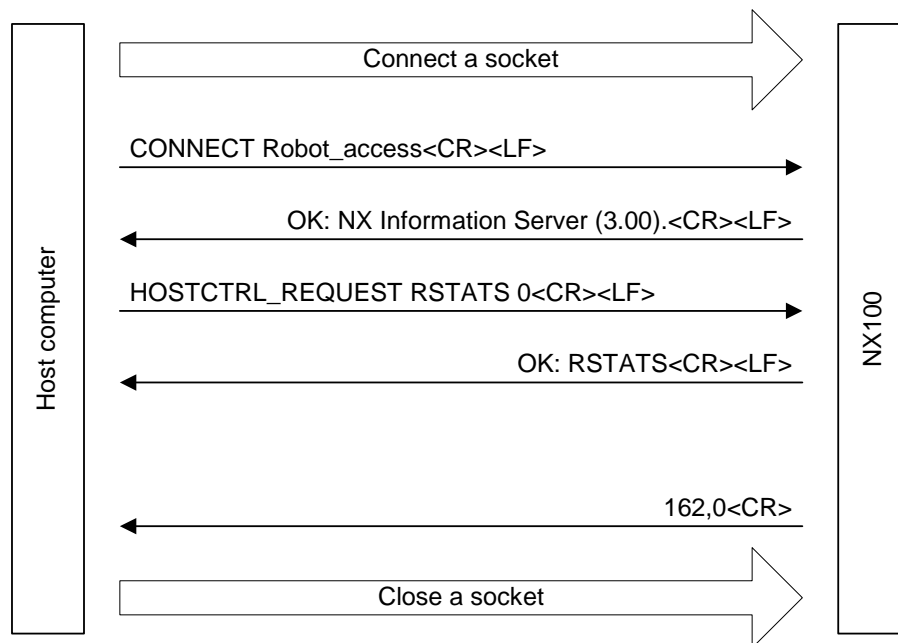- Only the network input (#22010 to #23287) can be written in.

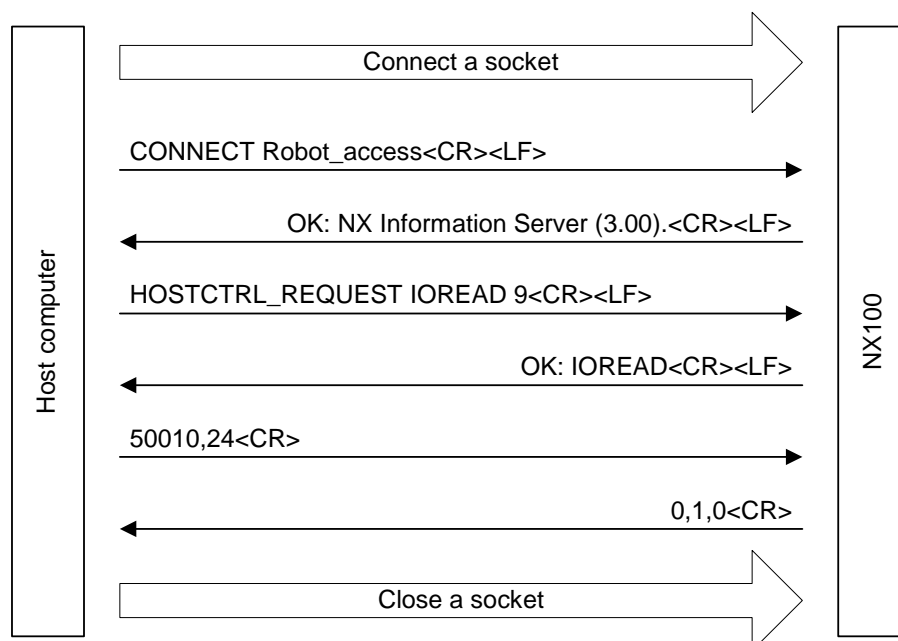Answer format: 0000 at normal answer

3-29

3.3 Transmission Examples

3.3.1 Read-Out of Status for Mode, Cycle, etc

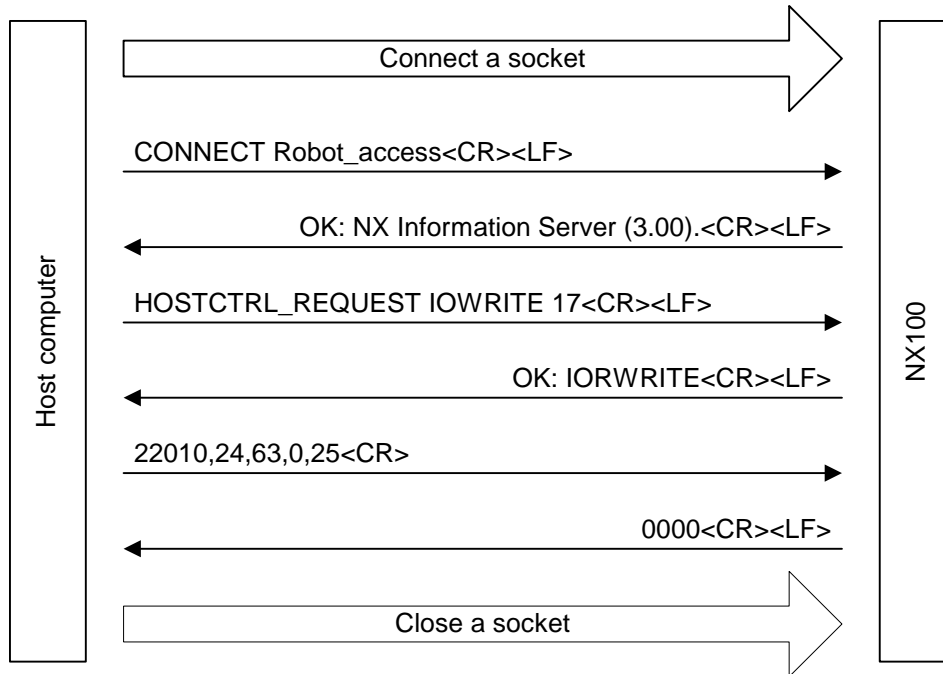See and follow the figure below to read out each status of mode, cycle, etc.

```
Host computer                                                    NX100

              ──────── Connect a socket ────────▷

              CONNECT Robot_access<CR><LF> ──────────▶

              ◀────── OK: NX Information Server (3.00).<CR><LF>

              HOSTCTRL_REQUEST RSTATS 0<CR><LF> ──────▶

              ◀────── OK: RSTATS<CR><LF>

              ◀────── 162,0<CR>

              ──────── Close a socket ────────▷
```

3.3.2 Read-Out of I/O Signals for the NX100

See and follow the figure below to read out three bytes (24 bits) from #50010.

```
Host computer                                                    NX100

              ──────── Connect a socket ────────▷

              CONNECT Robot_access<CR><LF> ──────────▶

              ◀────── OK: NX Information Server (3.00).<CR><LF>

              HOSTCTRL_REQUEST IOREAD 9<CR><LF> ──────▶

              ◀────── OK: IOREAD<CR><LF>

              50010,24<CR> ──────────────────▶

              ◀────── 0,1,0<CR>

              ──────── Close a socket ────────▷
```
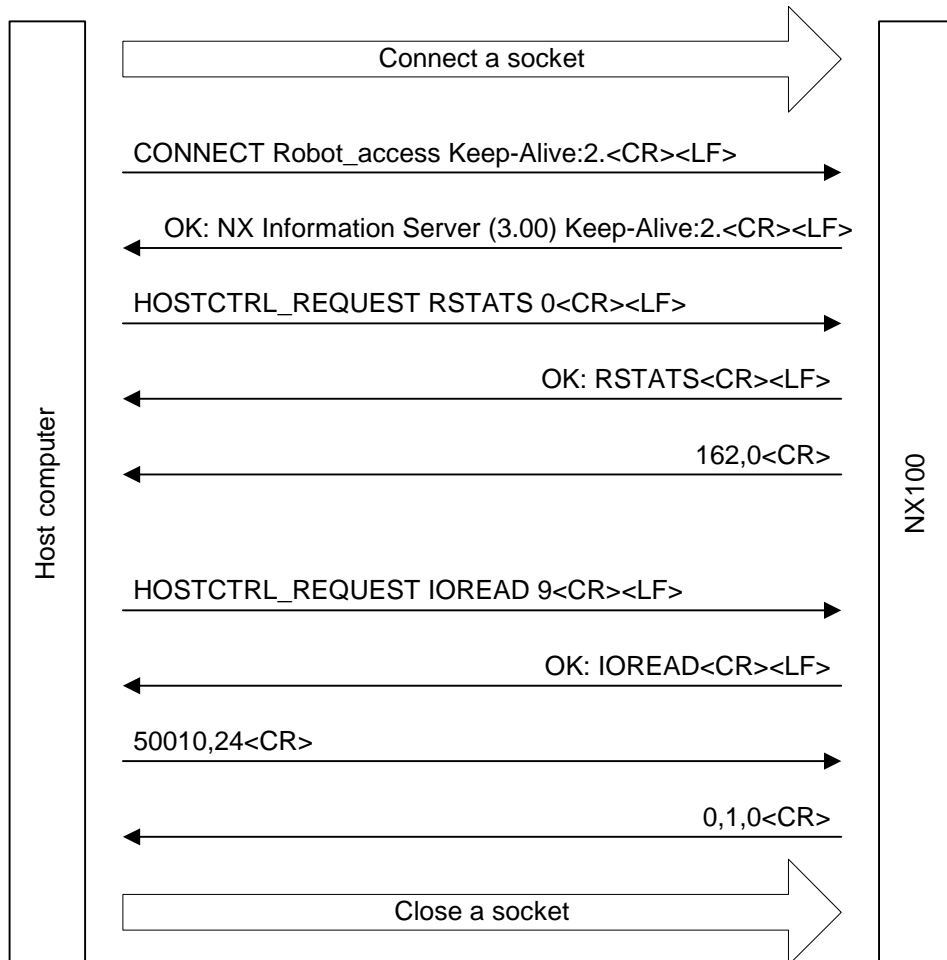
3-30

3.3.3   Write-In of I/O Signals for the NX100

See and follow the figure below to write in three bytes (24 bits) from #22010.

```
Host computer                                                    NX100

            ┌─────────────── Connect a socket ──────────────────▶

            CONNECT Robot_access<CR><LF>
            ───────────────────────────────────────────────────▶

                      OK: NX Information Server (3.00).<CR><LF>
            ◀───────────────────────────────────────────────────

            HOSTCTRL_REQUEST IOWRITE 17<CR><LF>
            ───────────────────────────────────────────────────▶

                                   OK: IORWRITE<CR><LF>
            ◀───────────────────────────────────────────────────

            22010,24,63,0,25<CR>
            ───────────────────────────────────────────────────▶

                                            0000<CR><LF>
            ◀───────────────────────────────────────────────────

            ┌─────────────────── Close a socket ────────────────▶
```

3-31

3.3.4 Continuous Execution of Multiple Commands

See and follow the figure below to read out three bytes (24 bits) from #50010, following the "Read-Out of Status for Mode, Cycle, etc".

```
Host computer                                                    NX100

            ┌────────────── Connect a socket ──────────────▶

            ── CONNECT Robot_access Keep-Alive:2.<CR><LF> ──▶

            ◀─ OK: NX Information Server (3.00) Keep-Alive:2.<CR><LF>

            ── HOSTCTRL_REQUEST RSTATS 0<CR><LF> ──────────▶

            ◀──────────────── OK: RSTATS<CR><LF>

            ◀──────────────── 162,0<CR>

            ── HOSTCTRL_REQUEST IOREAD 9<CR><LF> ──────────▶

            ◀──────────────── OK: IOREAD<CR><LF>

            ── 50010,24<CR> ──────────────────────────────▶

            ◀──────────────── 0,1,0<CR>

            ┌────────────── Close a socket ────────────────▶
```

3-32

4 Function Checkout Procedure

4.1 Cable Connection Check

- Check that the cable is securely connected to the LAN0 connector on the NCP01 board.
    - Check that the LED of the LAN0 (green LED on the left) is lit or blinks.

- Check that the cable connector of the hub side is properly connected.
    - Check that the power supply for the hub is ON.

- Check the cable type.
    - In case of connecting the cable to the hub which cannot auto-detect MDI/MDI-X connection, check if the cable used is a straight cable.
    - In case of directly connecting the NX100 and host computer, check if the cable used is a crossing cable.

- Check the baud rates of the hub and the host computer.
    - At start-up of the NX100, the NX100 auto-detects 10/100 Mbps for the NX100 baud rate. Communications, therefore, cannot be performed normally if the baud rate is changed after start-up of the NX100. Before start-up of the NX100, verify that the destination (HUB or host computer) of cable connection operates normally.

4.2 Connection Check with Lower Protocol

In the TCP/IP network, it is possible to check if the IP packets are transmitted to the destinations using the ping command from the host computer. Start the command prompt in the Windows 2000/XP, then input the IP address after entering "ping". If the communication is successfully done, the window displays as follows:

```
C:\>ping 192.168.255.1

Pinging 192.168.255.1 with 32 bytes of data:

Reply from 192.168.255.1: bytes=32 time<10ms TTL=255
Reply from 192.168.255.1: bytes=32 time<10ms TTL=255
Reply from 192.168.255.1: bytes=32 time<10ms TTL=255
Reply from 192.168.255.1: bytes=32 time<10ms TTL=255

Ping statistics for 192.168.255.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

In case of communication failure, the window displays as shown below. In this case, check the cable connections, network settings of the host computer, and the Ethernet communication settings of the NX100 once again.

```
C:\>ping 192.168.255.1

Pinging 192.168.255.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.255.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

4.3 Connection Check with telnet

Since the Ethernet server function uses the TCP port 80, it can be checked by the telnet command from the host computer whether the Ethernet server function operates normally.
In the Windows2000/XP, the command prompt can be started up to check the connection as follows:

1. Enter "telnet IP address 80".

   C:\>telnet 192.168.255.1 80

2. The display switches to the window for telnet.

3. Enter "CONNECT Robot_access" (At normal settings, an echo is not returned).

4. Enter "HOSTCTRL_REQUEST RSTATS 0".

   HOSTCTRL_REQUEST RSTATS 0

5. The check is completed when the response to command/answer is displayed.

   HOSTCTRL_REQUEST RSTATS 0
   OK: RSTATS
   194,0

4-3

5 Restrictions

5.1 Automatic Close of Socket

In the Ethernet server function, the socket is automatically closed in the following cases to prevent the socket connection from remaining unreasonably.
- When the command execution is completed at a single command execution
- When the command execution for the specified number of times is completed at multiple commands execution
- When the command is not input for a given amount of time at multiple commands execution
- When the processing abends (Including the case that the interpreter message is sent back.)

When the socket is closed, connect the socket again.

5.2 Unavailable Command/Function

The following commands/functions are not enabled in the Ethernet server function.
- JWAIT
- Load/Save of Files

Use the NX100 Ethernet function to enable these commands/files.

The FTP function (paid option) which is an advanced feature of the NX100 Ethernet function, can be used for load/save of files.

5.3 Restrictions on the Number of Sockets

The Ethernet server function, being different from the NX100 Ethernet function, adopts TCP, and arbitrary number can be used for the port on the host computer side. Furthermore, the Ethernet server function can perform faster data transmission compared to the NX100 Ethernet function. Therefore, if a socket is repeatedly created/closed in a small amount of time in the setting that allows the use of arbitrary unoccupied port for the port on the host computer side, many sockets remain waiting for time-out. Since the number of available sockets is limited, if sockets are used up to this limit, a new socket cannot be created until the sockets time out.

To avoid this, perform the following procedures if necessary.

- Perform multiple commands execution, not single command execution.
- Use a fixed number for the port number on the host computer side.
- (If sockets are insufficient on the personal computer side) Abortive connection release of a socket is forcibly executed.

5.4 Restrictions on Text Length

The NX100 data transmission function forms the basis of the Ethernet server function.
The text length of the Ethernet server function is therefore restricted to 256 bytes as is the NX100 data transmission function.
If command data are transmitted from the host computer to the NX100, the message exceeding 256 bytes cannot be sent.
If an answer is sent back to the host computer from the NX100, the message exceeding 256 bytes are split into several packets and sent back.

5.5 Restrictions on Access from Multiple Host Computers

Communication resources of the NX100 are occupied by the host computer during the period from when the host computer sends a START request (CONNECT Robot_access<CR><LF>) until when the NX100 sends back an answer. This situation makes it impossible to receive a START request from other host computers (or other applications of the foregoing host computer). Therefore the START requests from the other host computers are forced to wait until an earlier request is completed.
Especially if a START request is specified to infinite execution (CONNECT Robot_access Keep-Alive:-1<CR><LF>), START requests from the other host computers cannot be received. If access from several host computers is required, build a system so that a particular application does not occupy the communication resources of the NX100.

5-2

NX100 OPTIONS
INSTRUCTIONS

FOR ETHERNET SERVER FUNCTION:
ADVANCED FEATURE OF NX100 ETHERNET FUNCTION

YASKAWA ELECTRIC CORPORATION

YASKAWA