TS3000 series Robot Controller

TS3000   SCARA / LINEAR system
TS3100   SCARA / LINEAR system
TSL3000 SCARA system

INSTRUCTION MANUAL

COMMUNICATION MANUAL

Notice

- Make sure that this instruction manual is delivered to the final user of Toshiba Machine's industrial robot.

- Before operating the industrial robot, read through and completely understand this manual.

- After reading through this manual, keep it nearby for future reference.

TOSHIBA MACHINE CO., LTD.

– a –

Preface

This manual describes serial communication and ethernet communication between the robot controller and peripheral devices.   It covers such subjects as connecting communication channels and setting communication modes.   It also describes communication protocols, communication commands and data format, and presents information on how to operate the robot with data communication and how to handle communication dialogue with controller programs.

Before reading this manual, we ask that you first read and understand the contents of the following user manuals.

- Operator's Manual
- Robot Language Manual
- Interface Manual

This manual is divided into six (6) sections:

Section 1   Introduction
This section presents and introduction to the communication functions provided by the TS3000 Series robot system.

Section 2   COM1 Port and HOST Port Specifications and Settings
This section describes the RS-232C port hardware interfaces, communication modes, and other information.

Section 3   Ethernet Specifications and Settings
This section describes the Ethernet hardware interfaces, communication modes, and other information.

Section 4   Non-Protocol Communication
This section describes the protocol for communication between an external device and the robot program only.

Section 5   Simple Protocol Communication
This section describes the communication protocol, commands, and robot program files for communication with the host computer.

Section 6   Robot Operation Sequence Using External Communication
This section shows examples of methods for operating the robot.

Table of Contents

Section 1

Introduction

This manual describes serial communication and ethernet communication between the robot controller and an external computer. There are two procedures for data transmission: non-protocol communication and simple protocol communication, each of which can achieve the following functions.

Non-protocol communication
   (1) Input variables into the program from the external device.
   (2) Output variables or messages in the program.

Simple protocol communication
   (1) Start and stop operation of the robot
   (2) Program reset, step reset, cycle reset and output signal reset
   (3) Servo ON/OFF
   (4) Upload and download program files
   (5) Erase program files
   (6) Select programs
   (7) Monitor the status of the robot
   (8) Monitor for robot errors

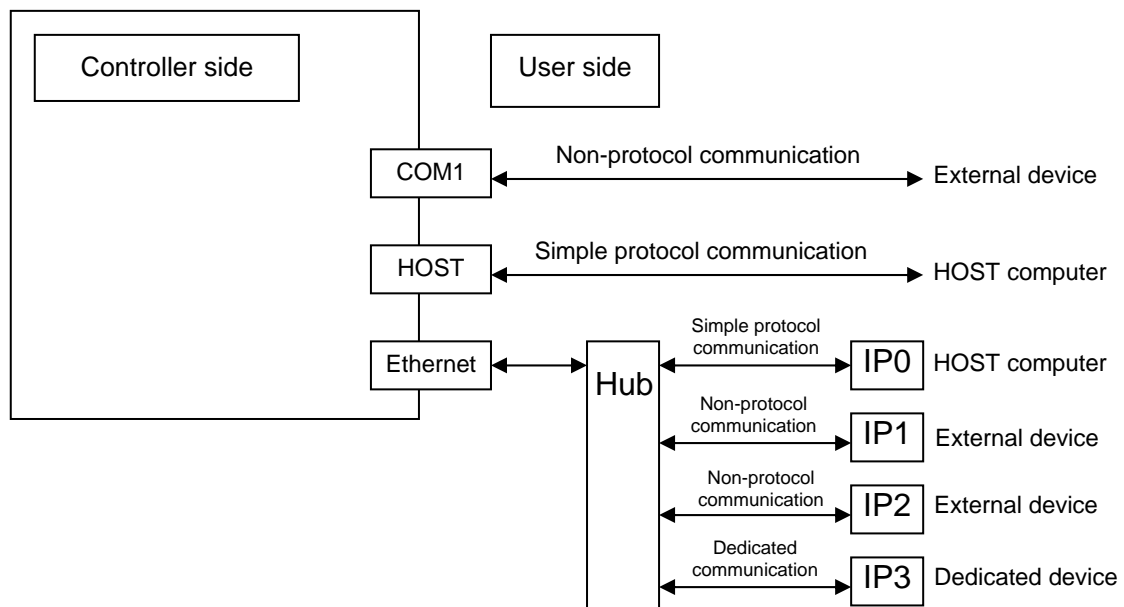The communication interface configuration is as shown below.

Fig. 1.1   Communication interface configuration

Section 2

COM1 Port and HOST Port Specifications and Settings

The TS3000/TSL3000 controller has two (2) serial communication ports. The COM1 port is used for non-protocol communication, being connected to external devices, and the HOST port is used for protocol communication and communicates with the HOST computer.

2.1 Connections

The COM1 port and HOST port should be connected, using an RS-232C cross connect cable (D-SUB: 9-pin).   (For details, see the Interface Manual.)
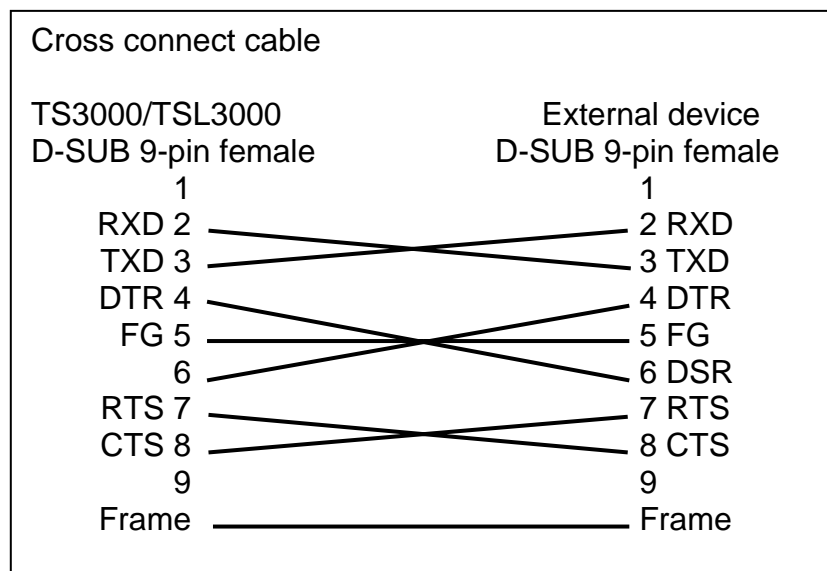
```
Cross connect cable

TS3000/TSL3000                    External device
D-SUB 9-pin female               D-SUB 9-pin female
        1                               1
RXD 2  ─────────╲  ╱──────────  2 RXD
TXD 3  ─────────╱  ╲──────────  3 TXD
DTR 4  ──────╲                   4 DTR
 FG 5  ────────────╳──────────  5 FG
    6  ──────╱       ╲────────  6 DSR
RTS 7  ─────────╲  ╱──────────  7 RTS
CTS 8  ─────────╱  ╲──────────  8 CTS
    9                           9
Frame  ──────────────────────  Frame
```

Fig. 2.1   Pin arrangement for cross connect cable

2.2 Communication Specifications

2.2.1 COM1 Port

Table 2.1 COM1 port communication specifications

Item	Specifications
Interface	RS-232C
Synchronous system	Start-stop synchronization system
Communication system	Full duplex system
Communication rate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 bps
Data format	ASCII code
No. of transfer bytes	Max. 256 bytes
Transfer data structure	Data length: 7 or 8 bits Parity: None, odd, even Stop bit: 1 or 2 bits
Protocol	Non-protocol communication
Timeout	No timeout
Fault recovery	No special protocols are used.

Note: The communication rate and transfer data structure are specified in the user parameter file.   (For details, see Para. 2.3.)

2.2.2　HOST Port

Table 2.2　HOST port communication specifications

Item	Specifications
Interface	RS-232C
Synchronous system	Start-stop synchronization system
Communication system	Half-duplex system
Communication rate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 115200 bps
Data format	ASCII code
No. of transfer bytes	Max. 255 bytes
Transfer data structure	Data length: 7 or 8 bits Parity: None, odd, even Stop bit: 1 or 2 bits
Protocol	Simple protocol communication
Timeout	10 sec. (character receiving intervals)
Fault recovery	Command is re-sent from the host computer. No special protocols are used.

Note:　The communication rate and transfer data structure are specified in the user parameter file.　(For details, see Para. 2.3.)

2.3   Communication Parameters

The communication rate and transfer data structure of the COM1 port and HOST port are defined in the user parameter file (USER. PAR).
Communication is performed according to the contents defined under [U06] of the user parameter file (USER.PAR) in the RAM drive.

[U06] Serial port setting

Specify the communication ports [COM1 port, HOST port].

TS3000

```
[U06] Serial port setting
{Speed }(38400, 19200, 9600, 4800, 2400, 1200)
{Character}(7, 8)
{Parity }(0:Without, 1:Odd, 2:Even)
{Stop bit }(1, 2)
{COM1}
= 9600   8   0   1
{HOST}
= 9600   8   0   1
```

{COM1}

=   (Speed) (Character length) (Parity) (Stop bit length)   ← COM1 port setting

{HOST}

=   (Speed) (Character length) (Parity) (Stop bit length)   ← HOST port setting

(Speed) : Select the speed for data communication from the following six (6) rates.

115200	:	115200 bps
38400	:	38400 bps
19200	:	19200 bps
9600	:	9600 bps
4800	:	4800 bps
2400	:	2400 bps
1200	:	1200 bps

(Character length) : Specify the length of characters to be transferred.

8 : 8 bits

7 : 7 bits

(Parity) : Specify the parity of characters to be transferred.

0 : Without parity

1 : Odd-number parity

2 : Even-number parity

(Stop bit length) : Specify the stop bit length of characters to be transferred.

1 : Stop bit 1

2 : Stop bit 2

Set value

Example: TS3000

= 9600 8 0 1

= 38400 8 1 1

"Speed 9,600 bps, character length 8 bits, without parity and stop bit 1" are specified for the COM1 port.

"Speed 38,400 bps, character length 8 bits, even-number parity and stop bit 1" are specified for the HOST port.

Section 3

Ethernet Specifications and Settings

The TS3000/TSL3000 controller has one Ethernet connector.

IP0 can communicate with the host computer using simple protocol, IP1 and IP2 can communicate with an external device using non-protocol communication, and IP3 can communicate with a dedicated device using a dedicated protocol.

This function uses TCP/IP protocol to enable communication on the same local area network with computers and visual equipment on the Ethernet. This allows program creation and editing, debugging start support, and maintenance using the computer programmer TSPC.

Data can be sent and received by the robot program for exchanging numerical and positional data.

INPUT and PRINT statements from SCOL language can be used to perform data input and output.

Example: INPUT IP1, INDATA, PRINT IP1, OUTDATA, CR, etc.

3.1   Connections

If a hub is used to connect to an external device, connect using a straight cable.   Use a cross connect cable when connecting a computer directly to the controller.   Some devices have the automatic recognition function so that either a straight cable or cross connect cable can be used, and so check the specifications of the device that is used.

3.2   Communication Specifications

Table 3.1   Communication specifications

Item	Specification
Interface	10BASE-T
Communication system	TCP
Communication rate	10 Mbps
Maximum packet size	512 bits/packet
Number of simultaneous connectable ports	4
Applications	IP0: Simple protocol communication IP1: Non-protocol communication IP2: Non-protocol communication IP3: Dedicated communication

3.3   Communication Parameters

This section describes the setting of the Ethernet parameters.   These settings are defined in the Ethernet Parameter File (ETHERNET.PAR).

[E00] Open mode   IP0 IP1 IP2 IP3

```
[E00] Open mode IP0 IP1 IP2 IP3
{ 0: non   1: Robot is TCP server   2: Robot is TCP client   }
=   1   0   0   0
```

IP0 is for simple protocol only, and so the server mode must be specified.
If the external device is a server, the controller uses client settings, and if the external device is a client, the controller uses server settings.

=(IP0) (IP1) (IP2) (IP3)

This enables or disables the Ethernet function.
0: Not used
1: Operates as a server
2: Operates as a client

[E01] Robot controller IP address

```
[E01] Robot controller IP address
=   192.168.0.124
```

This sets the IP address of the robot controller itself.
The IP address is an address for identifying the controller.   It is expressed as four (4) numbers from 0 to 255 with a period (.) in between.   If the controller is connected directly to a computer on the network, the IP address may remain at the initial value (any setting), but if it is connected to a LAN, the IP address must be set by following the instructions of the user's LAN system administrator.   This function will not work properly if any IP addresses are duplicated.   The robot controller and computer to communicate with it must be connected on the same network.
When sending data to devices on the network, the controller can be operated as a client or a server.

If the external device is a server, use the client setting for the controller, and if the external device is a client, use the server setting for the controller.

= (Robot controller IP address)

[E02] Robot controller name

(Planned for use in function expansion)

```
[E02] Robot controller name
= "NO1-TS3K"
```

This parameter specifies the name of the robot controller.
This is planned for use when the Ethernet function is expanded.

[E03] Subnet mask

```
[E03] Subnet mask
= 255.255.255.0
```

This parameter specifies the subnet mask of the connected network.
When connecting to a LAN, set the subnet mask by following the instructions of the user's LAN system administrator.

= (Subnet mask)

[E04] Default gateway

```
[E04] Default gateway
= 192.168.0.1
```

This parameter specifies the default gateway address.
When connecting to a LAN, set the subnet mask by following the instructions of the user's LAN system administrator.

= (Default gateway address)

[05] Own port no

```
[E05] Own port no
{ IP0 }
= 1000
{ IP1 }
= 1001
{ IP2 }
= 1002
{ IP3 }
= 1003
```

This parameter specifies each port number that is used by the controller. A port number is similar to a radio frequency. The numbers IP0 to IP3 indicate the port numbers.   Be careful that no port numbers are duplicated.

{IP0}
=(Port number used)
{IP1}
=(Port number used)
{IP2}
=(Port number used)
{IP3}
=(Port number used)

[06] Port number of destination

```
[E06] Port number of destination
{ IP0 }
= 1000
{ IP1 }
= 1001
{ IP2 }
= 1002
{ IP3 }
= 1003
```

This parameter specifies each port number that is used by the destination. This is used in client mode only. This must match the port number used by the server at the destination.

{IP0}
=(Port number of destination)
{IP1}
=(Port number of destination)
{IP2}
=(Port number of destination)
{IP3}
=(Port number of destination)

[E07] IP address of destination

```
[E07]   IP address of destination
{IP0}
= 192.168.0.150
{IP1}
= 192.168.0.151
{IP2}
= 192.168.0.152
{IP3}
= 192.168.0.153
```

STE 80722

This parameter specifies the IP address of the destination where each port is connected.
This is used in client mode only.

{IP0}
=(IP address of destination)
{IP1}
=(IP address of destination)
{IP2}
=(IP address of destination)
{IP3}
=(IP address of destination)

[E08] Robot community name

(Planned for use in function expansion)

```
[E08] Robot community name
= "NO1-TS3K"
```


This parameter specifies the community name of the network.
This is planned for usage when the Ethernet function is expanded.

[E09] DHCP

(Planned for use in function expansion)

```
[E09] DHCP
= 0
```


This parameter specifies if an address is obtained from the DHCP server.
This is planned for usage when the Ethernet function is expanded.

[E10] TCP status alarm IP0 IP1 IP2 IP3

> [E10] TCP status alarm IP0 IP1 IP2 IP3
> {0:NON 1:Lv_1 2:Lv_2 4:LV_4 8:Lv_8}
> = 0 0 0 0

For the port being used, the alarm level (1Lv, 2Lv, 4Lv, 8Lv) generated when the Ethernet connection is cut off during running of a program can be set.
When the value is 0 (zero), no alarm is generated.

*Parameter (ETHERNET.PAR) setting example

```
[E00] OPEN MODE IP0 IP1 IP2 IP3
{0:NON 1:ROBOT IS TCP SERVER 2:ROBOT IS CLIENT}
= 1   0   0   0
[E01] ROBOT CONTROLLER IP ADDRESS
= 192.168.0.124
[E02] ROBOT CONTROLLER NAME
= "NO1-TS3K"
[E03] SUBNET MASK
= 255.255.255.0
[E04] DEFAULT GATEWAY
= 192.168.0.1
[E05] OWN PORT NO
{IP0}
= 1000
{IP1}
= 1001
{IP2}
= 1002
{IP3}
= 1003
[E06] PORT NUMBER OF DESTINATION
{IP0}
= 1000
{IP1}
= 1001
{IP2}
= 1002
{IP3}
= 1003
[E07] IP ADDRESS OF DESTINATION
{IP0}
= 192.168.0.150
{IP1}
= 192.168.0.151
{IP2}
= 192.168.0.152
{IP3}
= 192.168.0.153
[E08] ROBOT COMMUNITY NAME
= "NO1-TS3K"
[E09] DHCP
= 0
[E10] TCP status alarm IP0 IP1 IP2 IP3
{0:NON   1:Lv_1   2:Lv_2   4:Lv_4   8:Lv_8}
= 0   0   0   0
[END]
```

Section 4

Non-Protocol Communication

Non-protocol communication is used for exchanging data with a robot language program and for printing out files.
The COM1 (RS-232C) port, and Ethernet IP1 and IP2 ports are used for non-protocol communication.

The data format for character strings to be transferred between the robot controller and the TCP/IP device is as shown below.

Character string + CR (0x0D)
Character string, character string + CR (0x0D)

These character strings are based on ASCII format.
ASCII 0x0D (carriage return) is added in the character string data.   When a plurality of data is transferred, the character string is separated by commas.

The functions as described in the simple protocol cannot be used for the non-protocol data communication.   Also, response to data received by the controller and timeout check are not processed at all.

4.1   Communication with a Robot Language Program

Data communication with a robot language program can be performed using the PRINT and INPUT commands.   For more information on the robot language, see "Robot Language Manual".

4.1.1   PRINT Command

The PRINT command allows you to send out a specified character string from the controller.   The controller transfers character strings or values of variable specified by the PRINT command.   The format of the PRINT command is shown below:

PRINT ⊔ [<channel>,] {<character string>|<expression>|}[,{<character string>|

<expression>}] .....[,CR]

<channel>:

Specifies the communication port over which the data is to be transmitted.   One of the following should be specified as the channel.

      COM1   : COM1 port
      IP1   : Ethernet IP1 port
      IP2   : Ethernet IP2 port
      TP   : Screen output to teach pendant

Unless <channel> is specified, data is transmitted to the teach pendant.

<character string>:

The character string to be transmitted is specified by enclosing that string in double quotation marks (").

<expression>:

Expressions made up of constants, variables, arithmetic operands and functions may be specified.

CR:

CR is used when the record end code (0x0D) is added to the last of sending data.

> Ex.:   PRINT COM1, "INPUT DATA =", -1000.0/3, CR
>
>   PRINT IP1, "INPUT DATA =", -1000.0/3, CR

Commas are used to separate any multiple character strings or expressions specified in the PRINT command.   Character strings are enclosed inside double quotation marks, and everything inside of those double quotation marks is transmitted in ASCII code.   Expressions are first solved, and the result is sent as a 12-character block of fixed length with the result pushed over to the right of that block.   Should the result of the expression be an integer, that result is sent as a Base 10 number having a maximum of ten places (digits).

Should the result of the expression be a real number, the result is sent as a number having an integral part with a maximum of four (4) digits and a decimal part with a maximum of three (3) digits (for a maximum of eight (8) places counting the decimal point).   One space is in front of the number is allocated for the sign (+ or -) of that number, although the sign is omitted if it is plus (+).   The number is sent in a 12-character block, with the number pushed over to the right.   The remaining spaces are filled with space codes (0x20).   The number itself is sent in ASCII code. The commas used to separate character strings or expressions in a PRINT command are themselves not transmitted.   When multiple character strings and/or expressions are specified with the PRINT command, the controller will send out these character strings and/or expressions as a single text.

> Ex.:   When the above example program is executed, the following data are
>   transmitted.
>
> INPUT DATA = -333.333 (CR)
>
> Note: CR signifies the record end code (0x0D).

4.1.2   INPUT Command

As opposed to the PRINT command, which is used for transmitting data, the INPUT command is used for receiving data.   The only data which may be received by the controller are integers and real numbers.   Data received by the controller is put into a variable (in a robot language program) specified by the INPUT command.   This data can be referred to later in the program to operate the robot.

The format of the INPUT command is shown below:

INPUT ⊔ [<channel>,] <variable>, [<variable>] ...

<channel>:
Specifies the communication channel over which the data is to be received.   One of the following should be specified as the channel.

COM1	:	COM1 port
IP1	:	Ethernet IP1 port
IP2	:	Ethernet IP2 port
TP	:	Key input from teach pendant

Unless <channel> is specified, the controller receives data from the teach pendant.

<variable>:
Specifies the variable in the robot program into which the data is to be entered.

> Ex.:   INPUT COM1, N1, N2
>
>   INPUT IP1, N1, N2

Commas are used to separate two (2) or more variables specified in the INPUT command.   The controller waits until data comes in over the communication channel specified by the INPUT command.   Add a record end code (0x0D) to the end of data which is sent to the controller.   When multiple units of data are to be sent to the controller, the individual data units should be separated with commas before being transmitted.   When more units of data are received by the controller than was specified with the INPUT command, the surplus data are ignored and used for the next INPUT command.   If fewer units of data are received by the controller

than was specified with the INPUT command, the controller waits until the short data reach.

Note 1)

The controller starts reading any data sent to it only after an INPUT command is executed.   Data are set in the ring buffer, and the input data are picked up according to the request of the INPUT statement.
Any data received before the INPUT command is executed may be ignored.   Take careful precautions when determining the data transmission timing.

Note 2)

Should program execution be suspended while the controller is waiting for data to come in (as directed by an INPUT command), the execution of the INPUT command will be cancelled.   Should the program be resumed, program execution will start from the step following the INPUT command.   Any variables for which data was not yet received when the program was suspended will be treated as 0.

You should keep this in mind when writing your robot language program, i.e., you should arrange your program in such a way that it will still function properly even should the controller (mistakenly) treat input values as 0.   One way to do this is to have the controller ask the host computer for confirmation (retransmission) whenever the controller receives a 0.   Another way to do this is to add on check-sum data (to the data to be transmitted) and check the validity of the received data. (For programming examples, see Para. 4.2, "Programming examples for communication with a robot language program.")

Note 3)

In the step operation mode, which is one of the test operation modes, when the INPUT command is executed, the program enters the wait state until data are received.   The same holds true when an INPUT command is executed directly.

Note 4)

When an INPUT command is to be used to receive multiple variables transmitted as one text from the host computer, individual numerals should be separated in the text with commas.

4.1.3   Clearing Communication Buffer

If the number of data the controller has received is larger than the number of data specified by the INPUT command, such data are stored in the communication buffer and used at request of the next INPUT command.   If data is left in the communication buffer, unexpected data will reach by the INPUT command.   To clear the data left in the communication buffer, output character string "BUFFRESET" to relevant communication port.

Example)
When clearing the communication buffer of the COM1 port:

  PRINT ⎵ COM1, "BUFFRESET"

Example)
When clearing the communication buffer of the IP1 port:

  PRINT ⎵ IP1,"BUFFRESET"

4.2   Programming Examples for Communication with a Robot Language Program

It is possible to utilize communication functions in a robot language program in order to do such things as specify program branches, correct the position of the robot to be performed, etc.   Listed below are several programming examples showing how to do this.

4.2.1   Program Branching

The INPUT command can be utilized to specify the number of times a certain action is to be repeated, to select a task for execution, etc.

Specifying the number of times an action is to be repeated (COM1)

PROGRAM REPEAT	In this program, the robot will repeat a certain series
N=0	of motions as many times as the external device
START:	tells it to.
PRINT COM1, "REQ"	First, the controller will send the character string
INPUT COM1, N	REQ to the external device. The external device
IF N == 0 THEN GOTO START	will reply by sending back the number of times the
FOR K = 1 TO N	operation is to be repeated.
MOVE A1	The controller will read in this number as the
MOVE A2	variable N, and will use this variable in the FOR
. . .	statement.
NEXT K	Should the program execution have been
PRINT COM1, "END"	suspended while the controller was waiting
END	receiving data in the INPUT command, the variable

N would be(mistakenly) taken as 0 when the program is resumed.   In such a case, branch is executed to the top address of the program by asking the external device for a retransmission. When the task is completed, the controller will send the character string END to the external device.

Data exchange

Controller		External device

PRINT COM1, "REQ"   Execute
                              REQ $\longrightarrow$

INPUT COM1, N   Execute
                           $\longleftarrow$ (Number of repetitions) Send

(Repeat operation.)
PRINT COM1, END   Execute
                           END $\longrightarrow$

Selecting a task to be performed (COM1)

```
PROGRAM SELECT
 K = 0
 START:
  PRINT COM1, "STR"
 SELECT:
  INPUT COM1, K
  GOTO(K) L1, L2, L3
  PRINT COM1, "NG"
  GOTO FIN
 L1:
   (Task 1)
  GOTO FIN
 L2:
   (Task 2)
  GOTO FIN
 L3:
   (Task 3)
  GOTO FIN
 FIN:
END
```

In this program, the external device tells the controller what task is to be performed. First, the controller will send the character string STR to the external device.   The external device will send back a number which specifying task the controller (robot) is to perform.   The controller reads in this number as variable K, which is then used in the GOTO statement to branch the program to the appropriate task.

Task 1 will be performed if the value of K is 1, Task 2 if the value is 2, and Task 3 if the value is 3.   If the value of K is anything else, the controller will send the character string NG (non-acknowledge) back to the external device.

Should program execution have been suspended while the controller was waiting for the number to come in, variable K will be taken as 0 when the program is resumed.   In this case also, the controller will send the character string NG back to the host computer.

Data exchange

Controller		External device
PRINT COM1, "STR"	Execute	
	STR ⟶	
INPUT COM1, K	Execute	
	⟵	(Task number) Send
(Execute specified task.)		
Should a task number of anything besides 1, 2 or 3 be received:		
PRINT COM1, "NG"	Execute	
	NG ⟶	
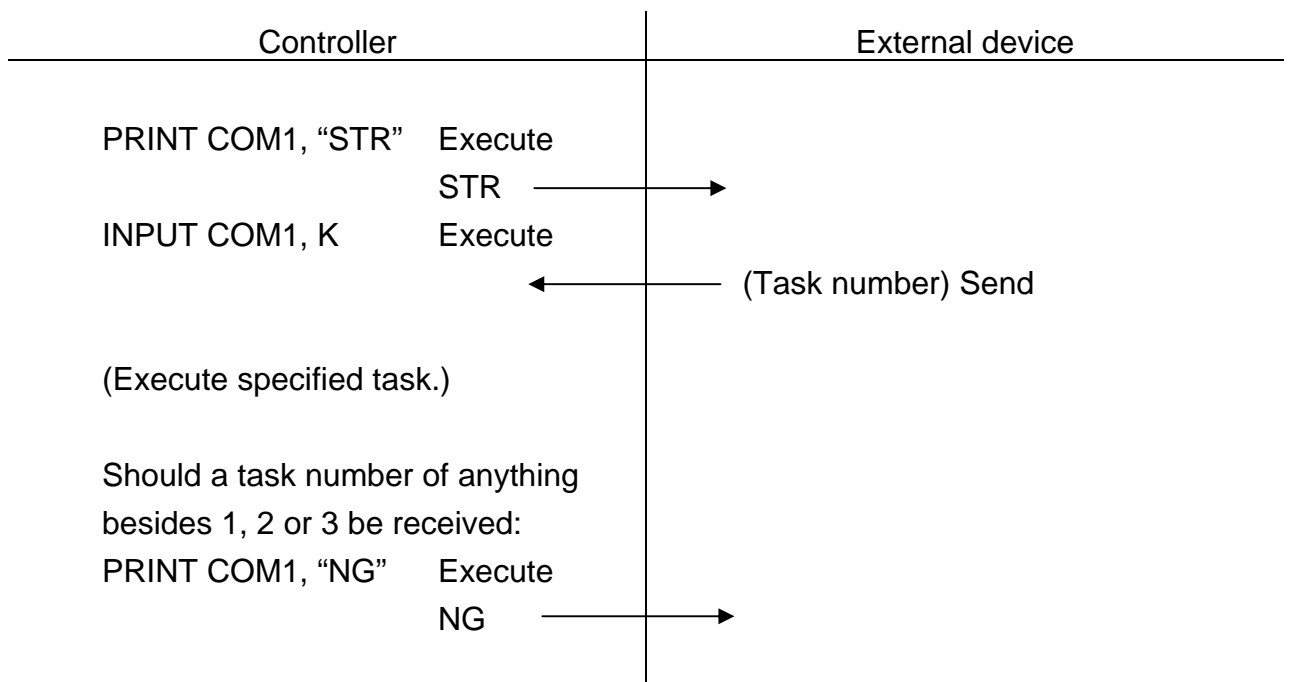
4.2.2   Correcting the Position of the Robot

Data received from the host computer can be used to correct the position of the robot.

Directly specifying position (COJM1)

```
PROGRAM DIRECT
  X=0.0
  Y=0.0
  Z=0.0
  C=0.0
  T=0.0
  PRINT COM1,"REQ"
  INPUT X,Y,Z,C,T,SUM
  IF SUM==X+Y+Z+C+T+1
              THEN GOTO ACTION
  PRINT COM1,"NG"
  GOTO FIN
  ACTION:
  PRINT COM1,"OK"
  P1=POINT(X,Y,Z,C,T)
  MOVE P1
  FIN:
END
```

In this program, the controller moves the robot to a position specified by the external device.

The controller sends the character string REQ to the external device.   The external device will respond by sending back the values for positional data X, Y, Z, C, and T, and also the sum of these values plus 1 (as check-sum data).   The controller will then see if the received data is correct by adding all the positional data, adding 1 to the result, and determining if check-sum value.   If it does agree, the controller will send the character string OK.

If the received data is wrong, the controller will send the character string NG back to the external device and quit the program. The controller will create positional data P1 using that data and then tell the robot to move to that point.

Data exchange

Controller	External device

PRINT COM1, "REQ"   Execute
REQ   ⟶

INPUT COM1, X, Y, Z, C, T, SUM

Execute   Positional data transmission

⟵   (<X>, <Y>, <Z>, <C>, <T>, <SUM>

(CR))

PRINT COM1, "OK"   Execute
OK   ⟶

Move to position (X, Y, Z, C, T)
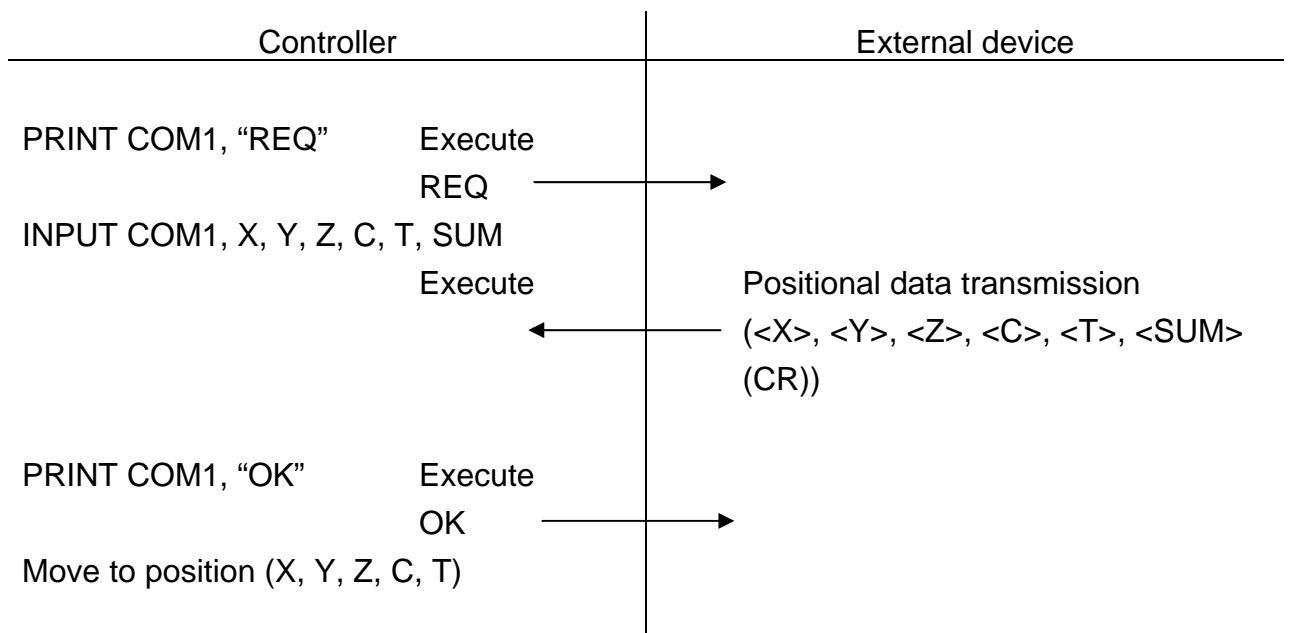
Specifying a relative position (COM1)

```
PROGRAM RELATIV
   X = 0.0
   Y = 0.0
   SUM = 0.0
   MOVE P1
   PRINT COM1, "REQ"
   INPUT COM1, X, Y, SUM
   IF SUM == X + Y + 1 THEN GOTO
   ACTION
   PRINT COM1, "NG"
   GOTO FIN
  ACTION:
   PRINT COM1, "OK"
   MOVE P1 + POINT (X, Y)
  FIN:
END
```

In this program, the controller moves the robot to a location specified by the external device.

The controller sends the character string REQ to the external device. The external device will respond by sending back the values for coordinate positions X and Y, and also the sum of these values plus 1 (as check-sum data).   The controller will then see if the received data is correct by adding X and Y, adding 1 to the result, value agrees with the check-sum value.   If it does agree, the controller will send back the character string OK to the external device.

If the received data is wrong, the controller will send the character string NG back to the external device and quit the program.

The controller will add the X and Y values of the transmission to the X and Y values of positional data P1 and make the robot move to the new position thus created.

Data exchange

Controller	External device

PRINT COM1, "REQ"   Execute

REQ ———————→

INPUT COM1, X, Y, SUM   Execute

←———————   Positional data transmission
(<X>, <Y>, <SUM> (CR))

PRINT COM1, "OK"   Execute

OK ———————→

Move to position P1 + (X, Y)
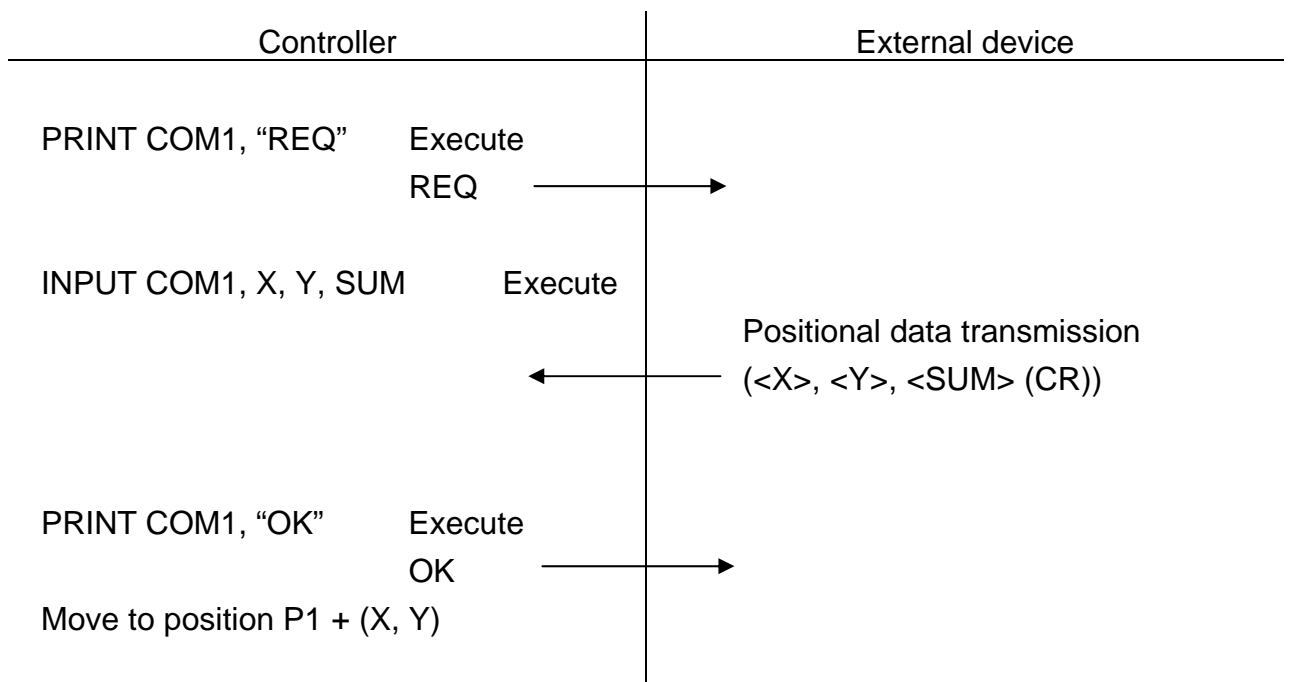
Section 5

Simple Protocol Communication

The simple protocol communication issues a command to the robot controller and performs start/stop operation of the robot, transfer of program files, and monitoring of the status.

5.1   Transmission Protocol

The RS-232C HOST port and Ethernet IP0 port wait for requests from the host computer after the power is supplied to the robot controller.
Basically, the host computer is a master station and the robot controller is a slave station.   The robot controller sends back necessary data to the host computer in reply to commands reached from the host computer.
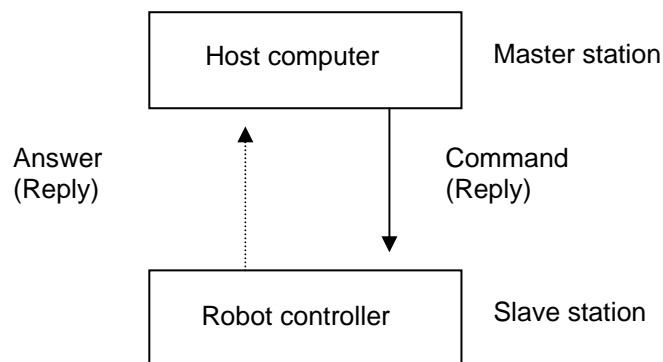
Fig. 5.1   Basic communication

Should the robot controller return an NG (Non-acknowledge) code in response to a command from the host computer, or should the robot controller give no response at all, resend the command from the host computer.   Even should an error have occurred while transmitting a file, resend the file upload (or download) command from the host computer.

Furthermore, after receiving text from the controller, insert a delay of about 50 msec before beginning to transmit subsequent text from the host computer.

5.2 Transmission Format

Transmissions are made with the text unit shown below. A maximum of 253 bytes of actual data can be transmitted as a single text. Actual data in amounts over 253 bytes will be transmitted in one of the following two ways.

(1) Files (robot programs, positional data, parameters) will be broken down into multiple texts and transmitted as described in Para. 5.3.3. The receiving station will send an answer signal back for each block of text transmitted. File upload (or download) commands are not necessary to transmit the second and following blocks of text.

(2) Messages to be transmitted
     Messages will be sent with multiple transmissions. In other words, the data will be broken down into multiple texts and each text will be transmitted independently. The station that received the data will reconstruct the multiple texts back into a single message.

5.2.1 Text

←	Max. 255 bytes	→
← 1 byte →	← Max. 253 bytes →	← 1 byte →
STX(0x02)	DATA	ETX(0x03)
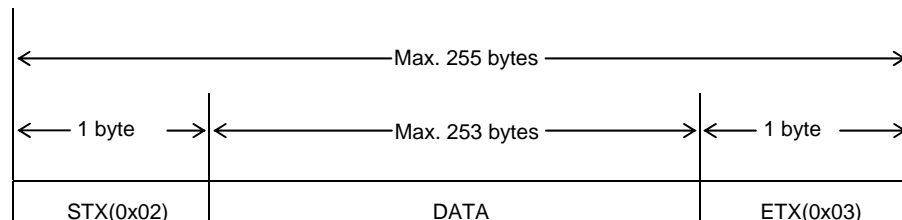
Text length: Max. 255 byte (from STX through ETX)

Text content:

• STX   Start of text code (0x02) – 1 byte

• DATA   Data section (Max. 253 bytes)

• ETX   End of text code (0x03) – 1 byte

5.2.2   Data Section Format

The basic format of data is shown below.

Command	Operand	CR(0x0D)
2 characters	Variable length	Delimiter
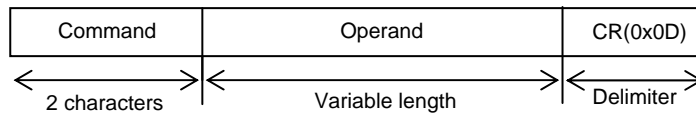
(1) Command

Command consists of two alphabetical letters which signify the type of command.
See Table 5.1 for command names and descriptions.

(2) Operand

The form of the operand varies depending on the type of command.   For more
information, read the description for the command in question.

(3) Delimiter

Add CR as delimiter at the end of the data.   In the description of commands in
Para 5.3.3, (CR) is used.

(4) Characters

The characters to be used should be ASCII code alphanumeric characters and
symbols.

Alphanumeric characters:

a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9

Special symbols:

" ' ( ) + - * / , . < > = ! [ ] { } % ^ & ? ;

5.2.3   Ethernet Status Display [ETHER]

(1) Function
Displays the Ethernet status.

(2) Procedures

1. Press the NEXT key repeatedly until [ETHER] is displayed in the menu, and then press [ETHER].
The screen shown below is displayed.

```
E T H E R N E T   S T A T U S                         [ 1 / 3 ]

  I P   A D D R E S S              1 9 2 . 1 6 8 .    0 . 1 2 4
  S U B N E T   M A S K            2 5 5 . 2 5 5 . 2 5 5 .    0
  M A C   A D D R E S S              0 0 0 0 4 8 : 1 4 0 2 0 6
  F I R M W A R E        S 1 S 6 0 0 0 0 / 0 0 0 1 : 0 0 : 2 0

 ████  ████  ████  ████  ████  ██
```

2. The display items can be changed by pressing ALT + [↓] and the NEXT key (or ALT + [↑] and the ALT + NEXT key).
The information displayed on each page is shown below.

(Page 1)
```
E T H E R N E T   S T A T U S                         [ 1 / 3 ]

  I P   A D D R E S S              1 9 2 . 1 6 8 .    0 . 1 2 4
  S U B N E T   M A S K            2 5 5 . 2 5 5 . 2 5 5 .    0
  M A C   A D D R E S S              0 0 0 0 4 8 : 1 4 0 2 0 6
  F I R M W A R E        S 1 S 6 0 0 0 0 / 0 0 0 1 : 0 0 : 2 0

 ████  ████  ████  ████  ████  ██
```

♦ IP ADDRESS: IP address of the robot controller itself that is written in the "ETHERNET.PAR" [E01] file
♦ SUBNET MASK: Subnet mask of the network that is written in the "ETHERNET.PAR" [E03] file
♦ MAC ADDRESS: MAC address of the robot controller itself
♦ FIRMWARE: Version of the Ethernet unit firmware

(Page 2)

```
E T H E R N E T   S T A T U S                           [ 2 / 3 ]
     O w n P o r t   C o n n e c t    I P             P o r t
I P 0      1 0 0 0      0 0 0 . 0 0 0 . 0 0 0 . 0 0 0   0 0 0 0 0
I P 1      0 0 0 0      0 0 0 . 0 0 0 . 0 0 0 . 0 0 0   0 0 0 0 0
I P 2      0 0 0 0      0 0 0 . 0 0 0 . 0 0 0 . 0 0 0   0 0 0 0 0
I P 3      0 0 0 0      0 0 0 . 0 0 0 . 0 0 0 . 0 0 0   0 0 0 0 0
```

- ♦ IP0 to IP3:   Ethernet channel number
- ♦ OwnPort: Robot port number
- ♦ Port: Port number of the destination
- ♦ Connect IP: IP address of the destination

(Page 3)

```
E T H E R N E T   S T A T U S                           [ 3 / 3 ]
       A p p .        M o d e          S t a t u s
I P 0   H O S T        T C P / S        L I S E N
I P 1   S C O L        T C P / C        C L O S E D
I P 2   S C O L        T C P / S        C L O S E D
I P 3   S Y S T E M    T C P / S        C L O S E D
```

- ♦ IP0 to IP3:   Ethernet channel number
- ♦ App.:   Name of application using the channel

  The connection in the control status from robot language uses simple protocol when HOST is shown and uses non-protocol communication when SCOL is shown.

- ♦ Mode: Protocol type, open status

  TCP/S indicates a server connection using TCP protocol, and TCP/C indicates a client connection using TCP protocol.

- ♦ Status: Ethernet port status

  The statuses below are displayed.   For the specific meaning of each status, refer to any commercially-available reference that provides explanations of TCP/IP.

  Closed

  Listen

  Syn-Sent

  Syn-Received

> Established
>
> Fin-Wait1
>
> Fin-Wait2
>
> Close wait
>
> Closing
>
> Last ACK
>
> Time Wait

(3) Cautions
- Check that the Ethernet cable between the controller and external device is connected correctly.
- Use a cross connect cable if connecting the robot controller directly to an external device, and use a straight cable if the connection passes through a hub.
- After making any parameter settings, turn the power off and then on again to enable the new settings.
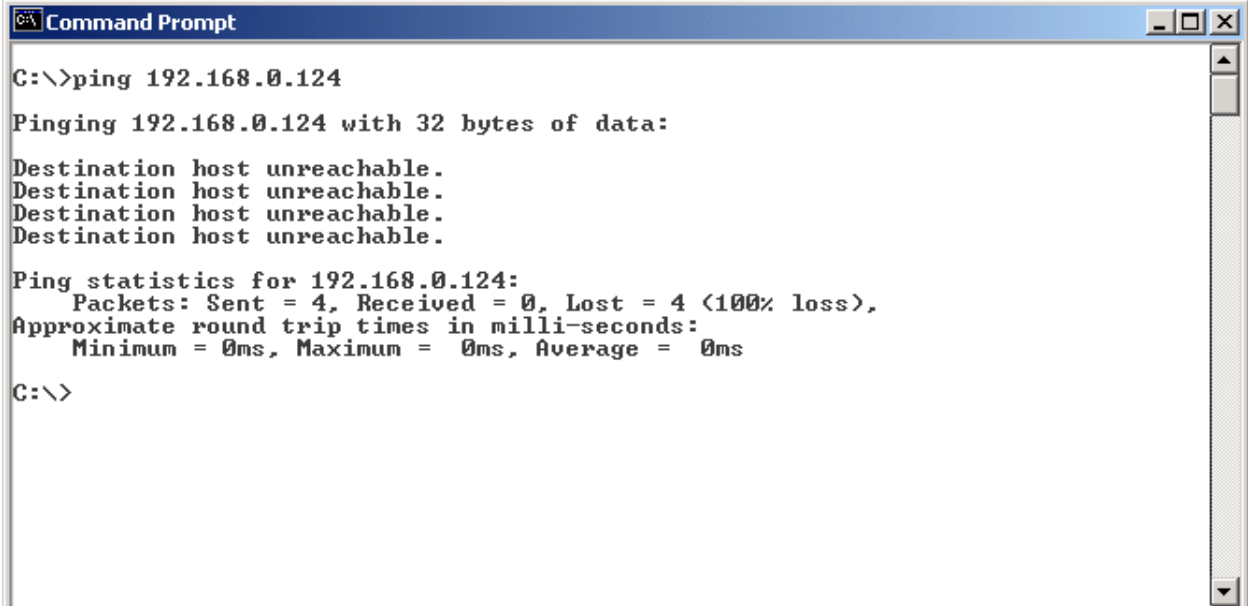
5.2.4   Confirmation with a Windows Computer

This describes how to confirm the connection using the DOS command "ping".

From the Start menu in Windows, go to the Command Prompt, and specify the IP address of the robot controller as shown below.

C:¥> ping   Robot controller IP address

If there is a transmission problem, the result "unreachable" or "Request Time out" is displayed as shown in Figure 5.2.

If transmission is performed normally, the result "Reply from …" is displayed as shown in Figure 5.3.
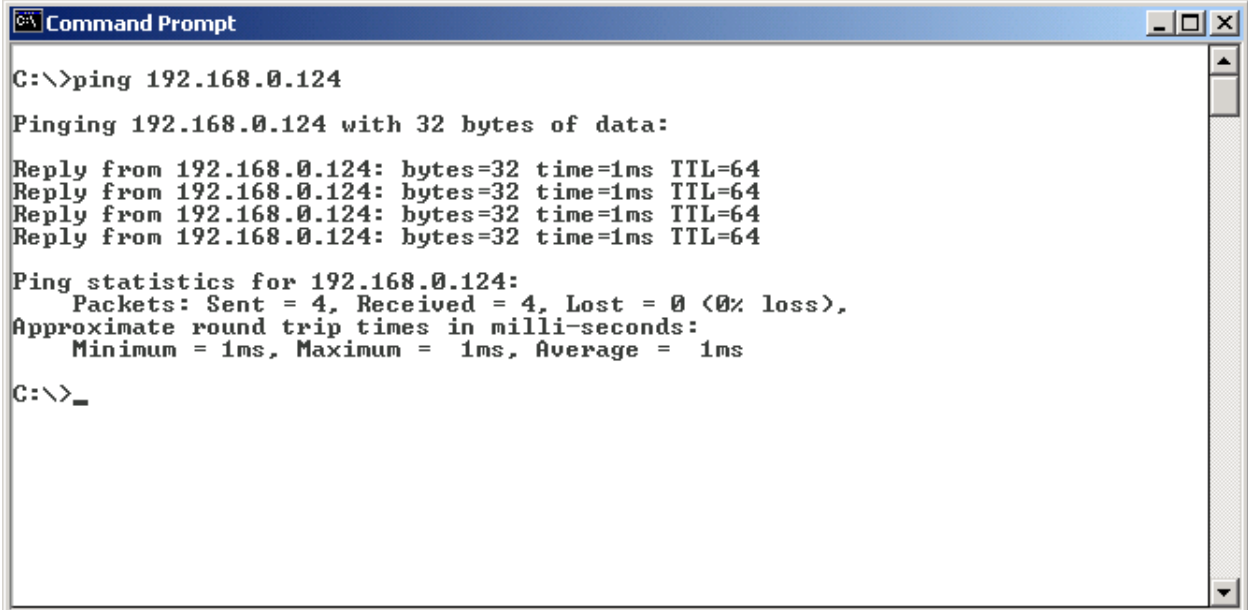
Fig. 5.2   Ping execution screen (Problem result)

Fig. 5.3   Ping execution screen (Normal result)

5.3   Commands

5.3.1   List of Commands

Commands which may be used with this communication protocol are shown in Tables 5.1 and 5.2.

Table 5.1   List of commands (Host computer → Controller)

No.	Command	Descriptions	Details	Text
1	RN	Start	Automatic operation	RN
2	SP	Stop		SP
3	BR	Servo OFF		BR
4	SO	Servo ON		SO
5	RS	Reset	Program reset	RS, PRG
			Step reset	RS, STP
			Cycle reset	RS, CYC
			Output signal reset	RS, SIG
			Select reset	RS, SEL
6	SL	Program selection		SL, <u>File name</u>
7	UL	File name	RAM file upload	UL, <u>File name</u>
8	DL	File name	RAM file download	DL, <u>File name</u>
9	CA	Directory	Directory request	CA
10	SU	Status	Status request	SU
11	EU	Error information	Error history upload	EU
12	ER	File erase		ER, <u>File name</u>
13	FL	File		FL, <u>File contents</u>
14	OK	Acknowledge		OK
15	NG	Non-acknowledge		NG
16	EC*	Internal command	Internal command transmit	EC, <u>Internal command</u>
17	MW	Variable write	Global-defined variable write	MW, <u>Flag Name, Type Data</u>
18	FD	Feed hold	Feed hold	FD
19	SF	System total status	Request for fast-speed status information	SF
20	MR	Variable read	Global-defined variable read	MR, <u>Name, Type</u>
21	IW	I/O write	I/O forced-writing	IW, <u>Line No., Status</u>
22	VR	Version read	Robot and software version	VR
23	DO	Execution of DO statement	Execution of DO statement	DO, <u>Statement</u>

　　　*　The internal commands executable by the EC command are as follows:

　　　(1)　MODE

　　　(2)　OVRD

　　　(3)　BREAK

Table 5.2　List of commands (Controller → Host computer)

No.	Command	Descriptions	Details	Text
1	FL*	File		FL, <u>File contents</u>
2	OK	Acknowledge		OK
3	NG	Non-acknowledge		NG

　　　*　The following files can be sent with the FL command.

　　　(1)　RAM files

　　　　　•　User file (program and positional data)

　　　　　•　Parameter file

　　　(2)　File directories

　　　(3)　Status files

　　　(4)　Error history files

　　　(5)　System total status files

　　　(6)　Version information files

　　　(7)　Variable read data files

5.3.2   Commands and Operation Modes

Tables 5.3 and 5.4 show the operation modes in which each command is operative.
The communications through the HOST port are effective irrespective of master modes, but their function is limited by a master mode.   For details, see the following table.

Table 5.3   HOST port (RS232C) commands and operation modes

• HOST port commands and operation modes

Master mode (Key switch)		TEACHING	INTERNAL (TS3000)	EXT		
Parameter (EXTERNAL mode control right)				SIG	RS-232C	ETHER
Maximum speed of robot		250mm/sec	100%	100%	100%	100%
Command	Description					
RN	Automatic operation start	X	X	X	△	○
SP	Stop	○	○	○	○	○
BR	Servo OFF	○	○	○	○	○
SO	Servo ON	X	X	X	○	X
RS	Program reset	X	X	X	○	X
SL	Program selection	X	X	X	○	X
UL	RAM file upload	△	△	△	△	△
DL	RAM file download	△	△	△	△	△
CA	Directory request	○	○	○	○	○
SU	Status request	○	○	○	○	○
ER	File erase	△	△	△	△	△
EU	Error history upload	○	○	○	○	○
EC	Internal command	X	X	X	○	×
MW	Variable write	○	○	○	○	○
FD	Feed hold	○	○	○	○	○
SF	System total status	○	○	○	○	○
MR	Variable read	○	○	○	○	○
IW	I/O write	X	X	X	○	X
VR	Version read	○	○	○	○	○
DO	Execution of DO statement	X	X	X	○	X

○: Valid, X: Invalid, △: Valid when the robot is stopped

The communications through the IP0 port are effective irrespective of master modes, but their function is limited by a master mode.   For details, see the following table.

Table 5.4   IPO port (Ethernet) commands and operation modes

Master mode (Key switch)		TEACHING	INTERNAL (TS3000)	EXT		
Parameter (EXTERNAL mode control right)				SIG	RS-232C	ETHER
Maximum speed of robot		250mm/sec	100%	100%	100%	100%
Command	Description					
RN	Automatic operation start	X	X	X	X	△
SP	Stop	○	○	○	○	○
BR	Servo OFF	○	○	○	○	○
SO	Servo ON	X	X	X	X	○
RS	Program reset	X	X	X	X	○
SL	Program selection	X	X	X	X	○
UL	RAM file upload	△	△	△	△	△
DL	RAM file download	△	△	△	△	△
CA	Directory request	○	○	○	○	○
SU	Status request	○	○	○	○	○
ER	File erase	△	△	△	△	△
EU	Error history upload	○	○	○	○	○
EC	Internal command	X	X	X	X	○
MW	Variable write	○	○	○	○	○
FD	Feed hold	○	○	○	○	○
SF	System total status	○	○	○	○	○
MR	Variable read	○	○	○	○	○
IW	I/O write	X	X	X	X	○
VR	Version read	○	○	○	○	○
DO	Execution of DO statement	X	X	X	X	○

○: Valid, X: Invalid, △: Valid when the robot is stopped

Commands transmitted from the host computer to the controller can be received by the controller in the modes marked "○".

Commands transmitted from the controller to the host computer can be transmitted by the controller in the mode marked "○".

Note: All external control input signals are also limited by a mater mode.
        For details, see the Interface Manual.

5.3.3   Details of Commands

: Automatic Operation Start   (Host computer → Controller)   RN

Format

RN   (CR)

Description

The RN (Automatic Operation Start) command is a command given by the host computer to the controller telling the controller to start up the program.   If, after stopping the program with the SP (Automatic Operation Stop) command, one sends the RN command again, the robot will start up from the step immediately following the step at which it was stopped.

Protocol

	Controller	Host computer
		RN (CR) (Automatic operation start-up directive)
(1)	Normal operation	
	OK (CR)	Normal response (Automatic operation normal commencement)
	In automatic operation	
	SP (CR)	Automatic operation stop
(2)	Error (Cannot receive request)	
	NG (CR)	Request not accepted (Servo OFF, etc.) Controller cannot be started up
(3)	Abnormal completion	Abnormal completion
	FA, <u>error code</u> (CR)	(Error occurred midway through)

Note

If the power is turned on while the controller is set for the host mode, the operation mode of the system will be the cycle operation mode.   If the controller is changed over to the host mode from the internal automatic mode, the operation mode of the system will be that in effect beforehand.

: Automatic Operation Stop   (Host computer → Controller)   SP

Format

SP   (CR)

Description

The SP (Automatic Operation Stop) command is given by the host computer to the controller telling the controller to stop automatic operation.   Also, this command informs the host computer when the controller is stopped.   Furthermore, should the robot be stopped in accordance with any request other than that made by the host computer, this command will automatically be sent from the controller to the host computer.

Protocol

Controller	Host

1. Automatic operation stop directive

          ← SP (CR)   (Automatic operation
                              stop directive)

  (1)   Normal operation
          OK (CR)   →   Normal response, program stop

  (2)   Error
          NG (CR)   →   Request not accepted

  (3)   Abnormal completion   →   Abnormal completion
          FA, <u>error code</u> (CR)

2.   Automatic operation stop
          Program stop
          SP (CR)   →
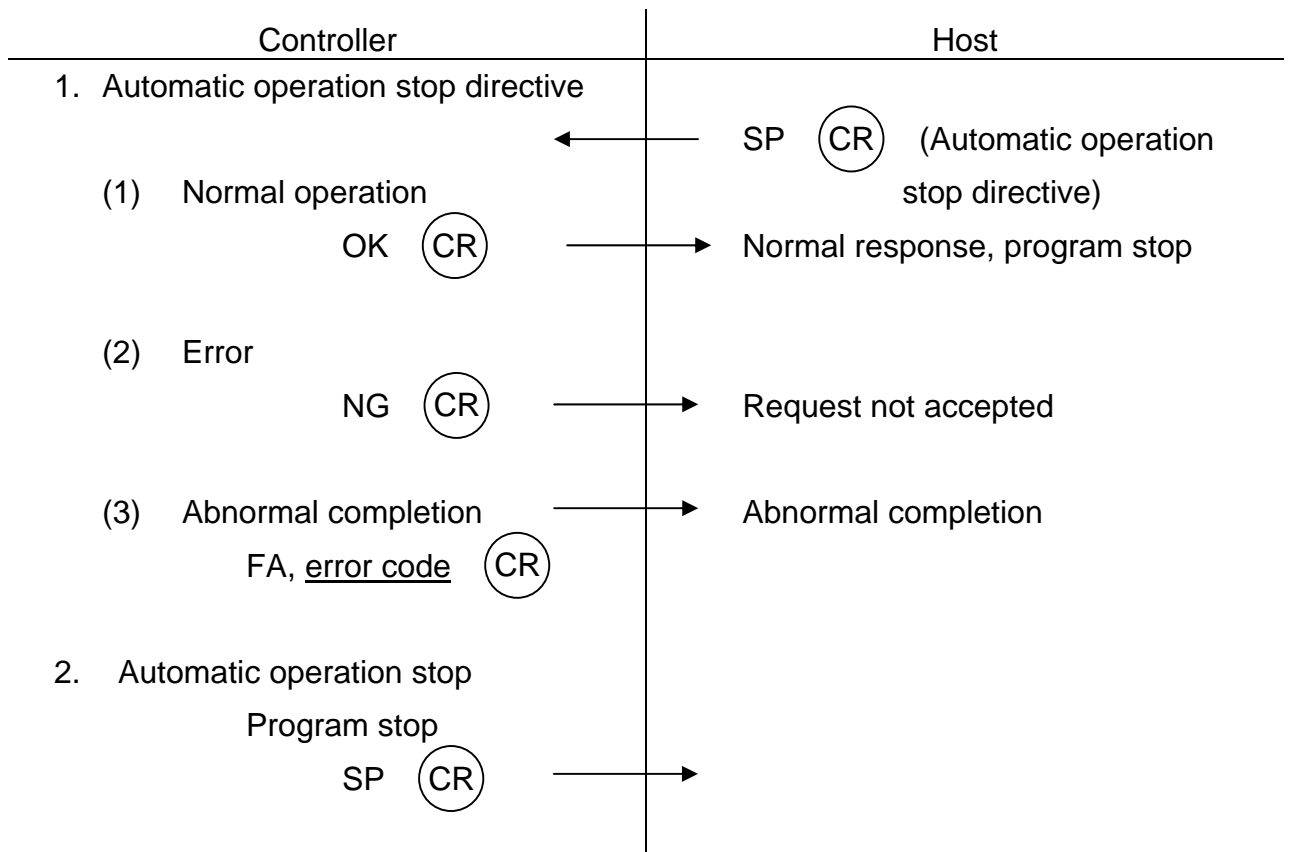
: Servo OFF   (Host computer → Controller)   BR

Format

BR   (CR)

Description

The BR (Servo OFF) command is given by the host computer to the controller telling the controller to turn off the servo power.

Protocol

Controller	Host computer

BR (CR) (Servo power OFF)

(1)   Normal response

OK (CR) ──────────→ Normal response

(2)   Error (Cannot receive request)

NG (CR) ──────────→ Request not accepted
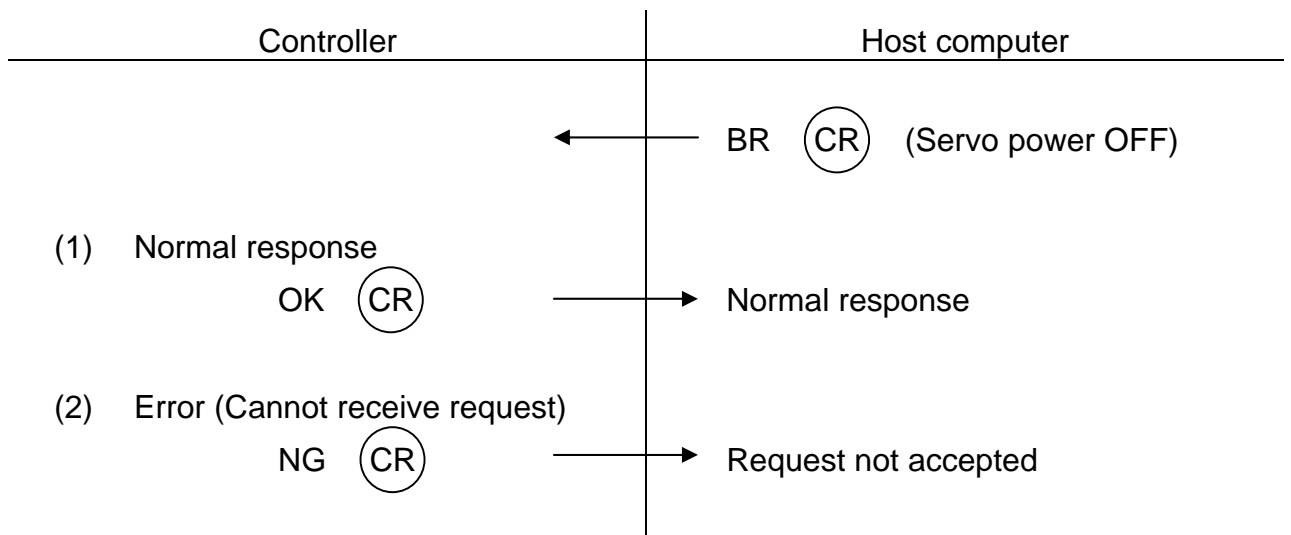
: Servo ON   (Host computer → Controller)   SO

Format

SO   (CR)

Description

The servo ON command is given from the host computer to the controller to turn the power on.

Protocol

Controller	Host computer

←———————— SO (CR)   (Servo power ON)

(1)   Normal response

OK (CR) ————————→ Normal response

(2)   Error (Cannot receive request)

NG (CR) ————————→ Request not accepted
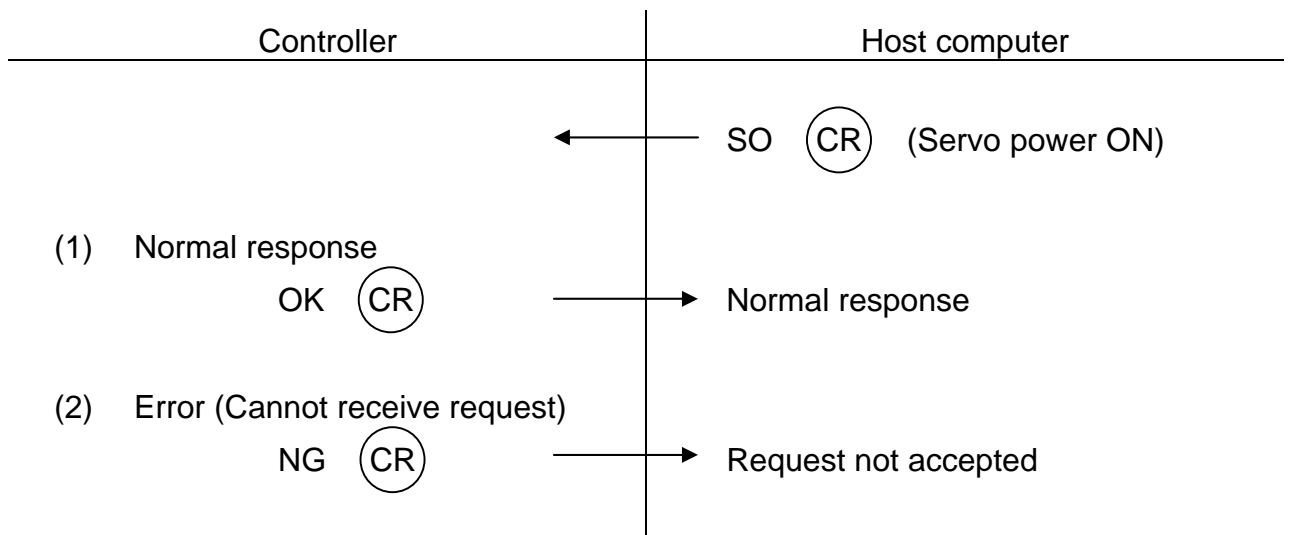
: Reset   (Host computer → Controller)   RS

Format

RS, XXX   (CR)

XXX : PRG   Reset all program data (internal data)
      STP   Reset the program back to Step 1.   All program data
            remains unchanged.
      CYC   Reset the program back to the step marked by the label
            RCYCLE.
      SIG   Turns off the digital output signals (DO1 to DO24) available
            to the user.
      SEL   Resets the execution file.

Description

The RS command will not be accepted unless the system is in a stop mode.

(1) PRG

   PRG will reset the program to Step 1.   All program data will be initialized.

(2) STP

   STP will reset the program to Step 1.   All program data (such as variables) will
   remain unchanged.

(3) CYC

   CYC will specify step number to the specified label (RCYCLE).   All program
   data will remain unchanged.

(4) SIG

   SIG will put the digital outputs in the non-active state.

(5) SEL

   SEL will reset the execution file.

Protocol

Controller	Host computer

←—————— RS, PRG (CR)

(1)   Normal completion

OK (CR) ——————→ Normal completion

(2)   Abnormal completion

NG (CR) ——————→ Request not accepted
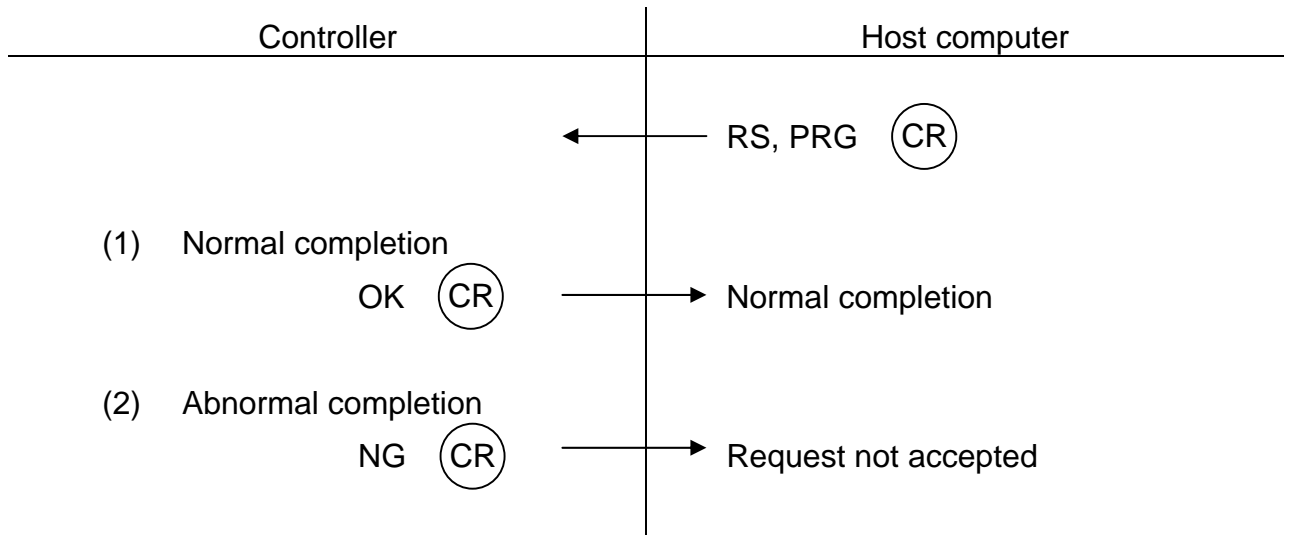
: Program Selection   (Host computer → Controller)   SL

Format

SL, <u>file name</u> (CR)

Description

The SL command selects a program to be executed under automatic operation.

Protocol

Controller	Host computer
	← SL, <u>file name</u> (CR)
(1) Normal completion	
OK (CR) →	Normal completion
(2) Abnormal completion	
NG (CR) →	Request not accepted
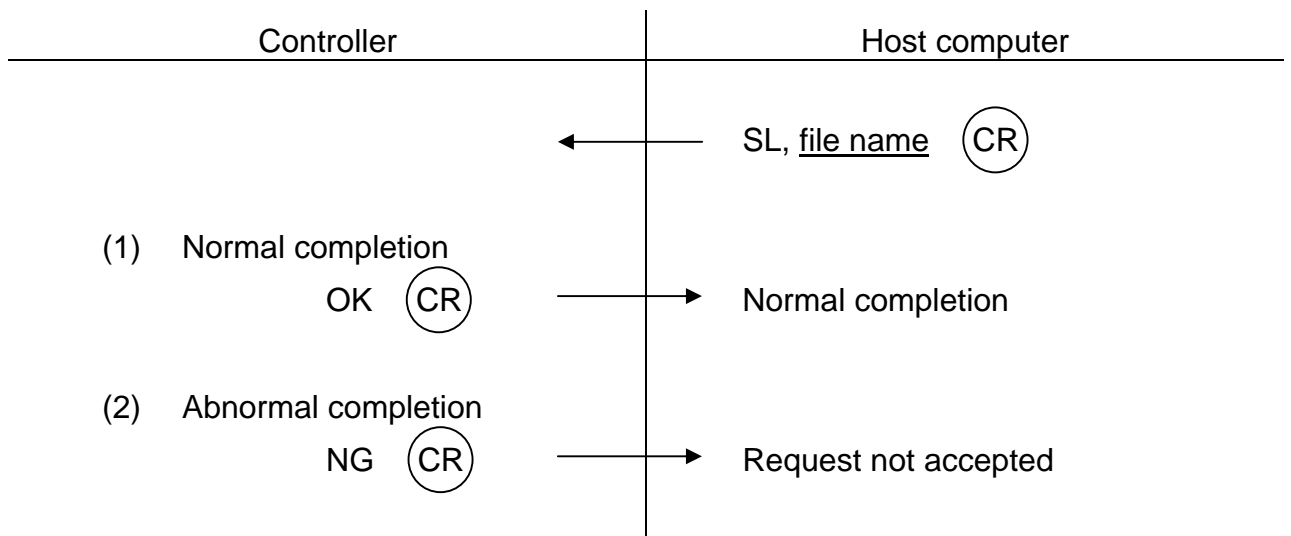
: File Upload Request   (Host computer → Controller)   UL

Format

UL, <u>file name</u> (CR)

Function

The UL command is used to upload (transmit) the specified RAM file in the controller to the host computer.   For information on the contents of files to be uploaded, see the FL (File) command.
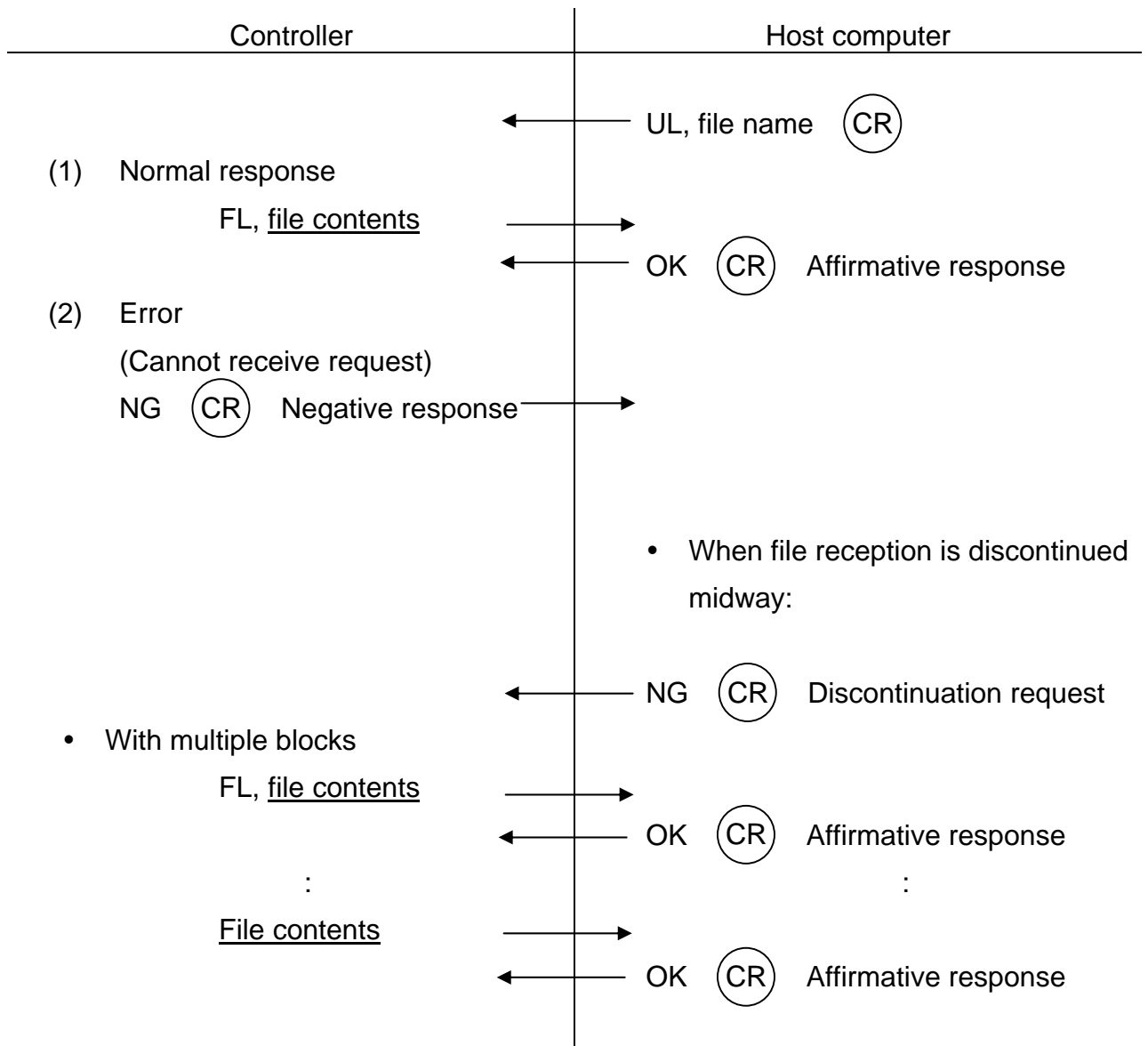
Protocol

Controller	Host computer
	← UL, file name (CR)

(1)   Normal response

    FL, <u>file contents</u> →

    ← OK (CR)   Affirmative response

(2)   Error

(Cannot receive request)

NG (CR)   Negative response →

- When file reception is discontinued midway:

← NG (CR)   Discontinuation request

- With multiple blocks

    FL, <u>file contents</u> →

    ← OK (CR)   Affirmative response

    :   :

<u>File contents</u> →

← OK (CR)   Affirmative response

Note

Put in a delay of about 50 msec until the host computer sends an affirmative response OK signal to the controller after the host computer has received the file contents.

: File Download Request   (Host computer → Controller)   DL

<div style="border:1px solid #000; display:inline-block;">Format</div>

DL, <u>file name</u> (CR)

<div style="border:1px solid #000; display:inline-block;">Function</div>

The DL command is used to download (transmit) a specified file from the host computer to the controller RAM drive.   For information on the contents of files to be downloaded, see the FL (File) command.

<div style="border:1px solid #000; display:inline-block;">Protocol</div>

Controller	Host computer
	← DL, <u>file name</u> (CR)
OK (CR) →	
	← FL, <u>file contents</u>
(1) Normal response	
OK (CR) →	
(2) Error (Cannot receive file)	
NG (CR) →	
Discontinuation/negative response	
• With multiple blocks	
	← FL, <u>file contents</u>
OK (CR) →	
:	:
	← <u>File contents</u>
OK (CR) →	
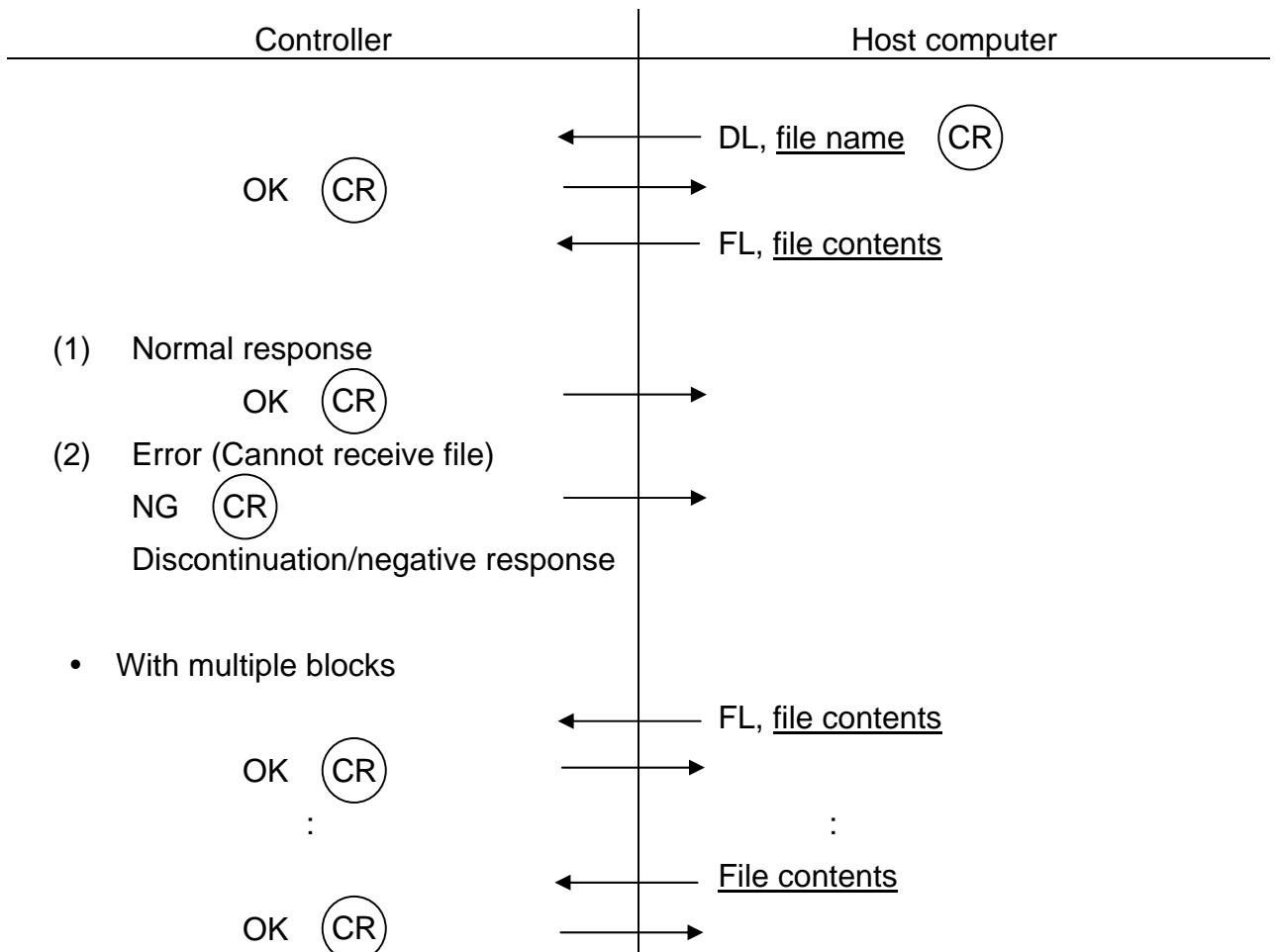
<div style="border:1px solid #000; display:inline-block;">Note</div>

Put in a delay of about 50 msec until the host computer sends the file contents to the controller after the host computer has received OK.

: File Directory Request   (Host computer → Controller)   CA

Format

CA   (CR)

Function

The CA command is used to transmit the directory of files in the RAM drive from the controller to the host computer.   For information on the contents of directories to be transmitted, see the FL (File) command.
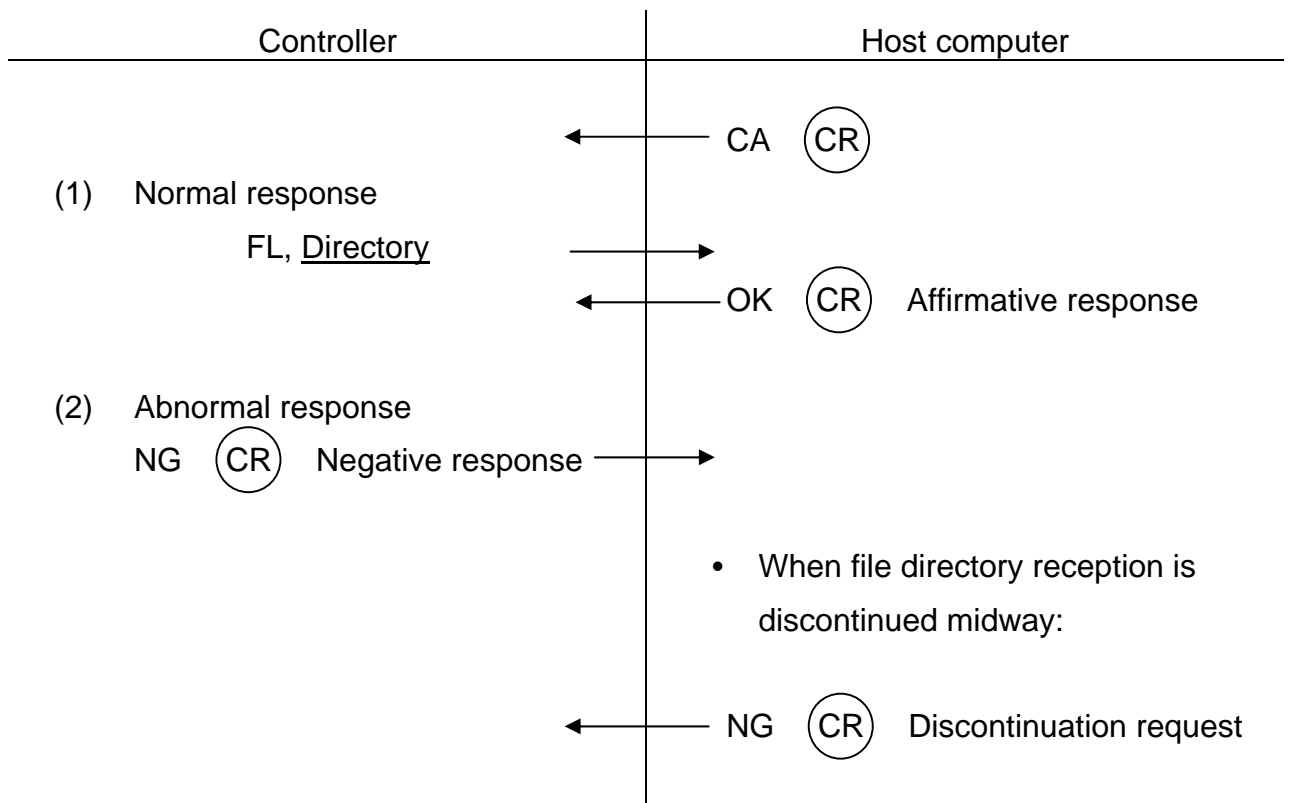
Protocol

Controller	Host computer

CA   (CR)

(1)   Normal response

　　　FL, <u>Directory</u>

OK   (CR)   Affirmative response

(2)   Abnormal response

NG   (CR)   Negative response

- When file directory reception is
  discontinued midway:

NG   (CR)   Discontinuation request

Note

Put in a delay of about 50 msec until the host computer sends an affirmative response OK to the controller after the host computer has received the directory.

: Status Request	(Host computer → Controller)	SU
: System Total Status Request	(Host computer → Controller)	SF

Format

SU (CR)   Status information

SF (CR)   System total status information

Function

The SU command is used to send the internal status (state) of the controller to the host computer.   For information on the contents of status file to be transmitted, see the FL (File) command.
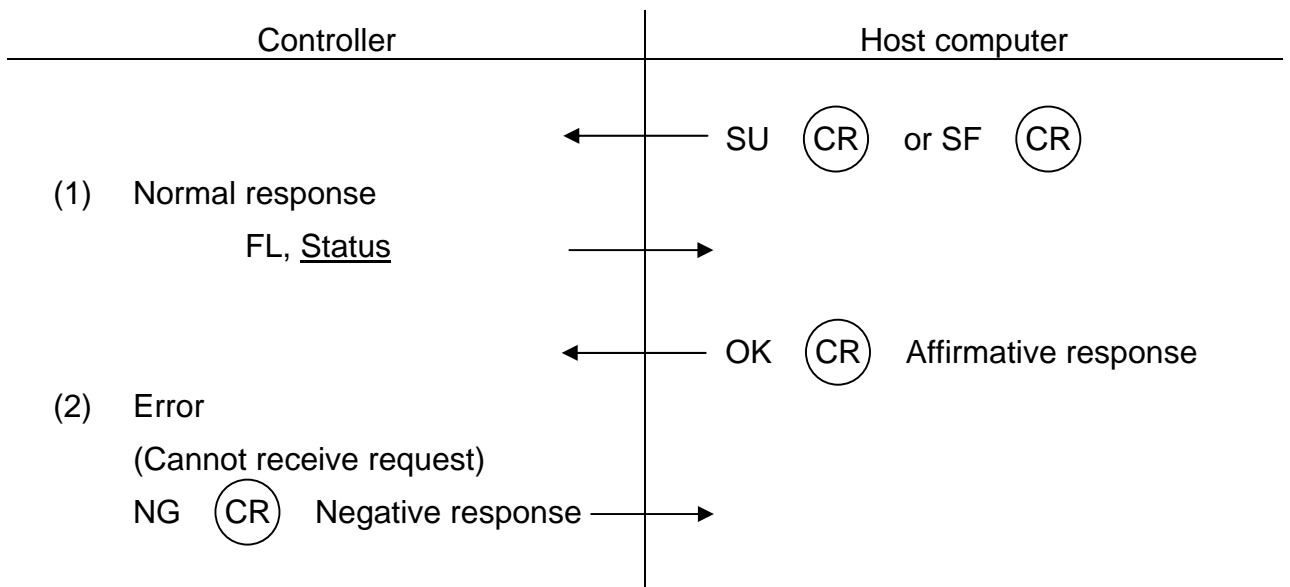
Protocol

Controller	Host computer

SU (CR) or SF (CR)

(1)   Normal response

FL, <u>Status</u>

OK (CR)   Affirmative response

(2)   Error

(Cannot receive request)

NG (CR)   Negative response

Note

Put in a delay of about 50 msec until the host computer sends an affirmative response OK to the controller after the host computer has received the status command.

: Error History Request   (Host computer → Controller)   EU

| Format |

EU   (CR)

| Function |

The EU command is used to send the controller error history data to the host computer.   For information on the contents of error history files to be transmitted, see the FL (File) command.
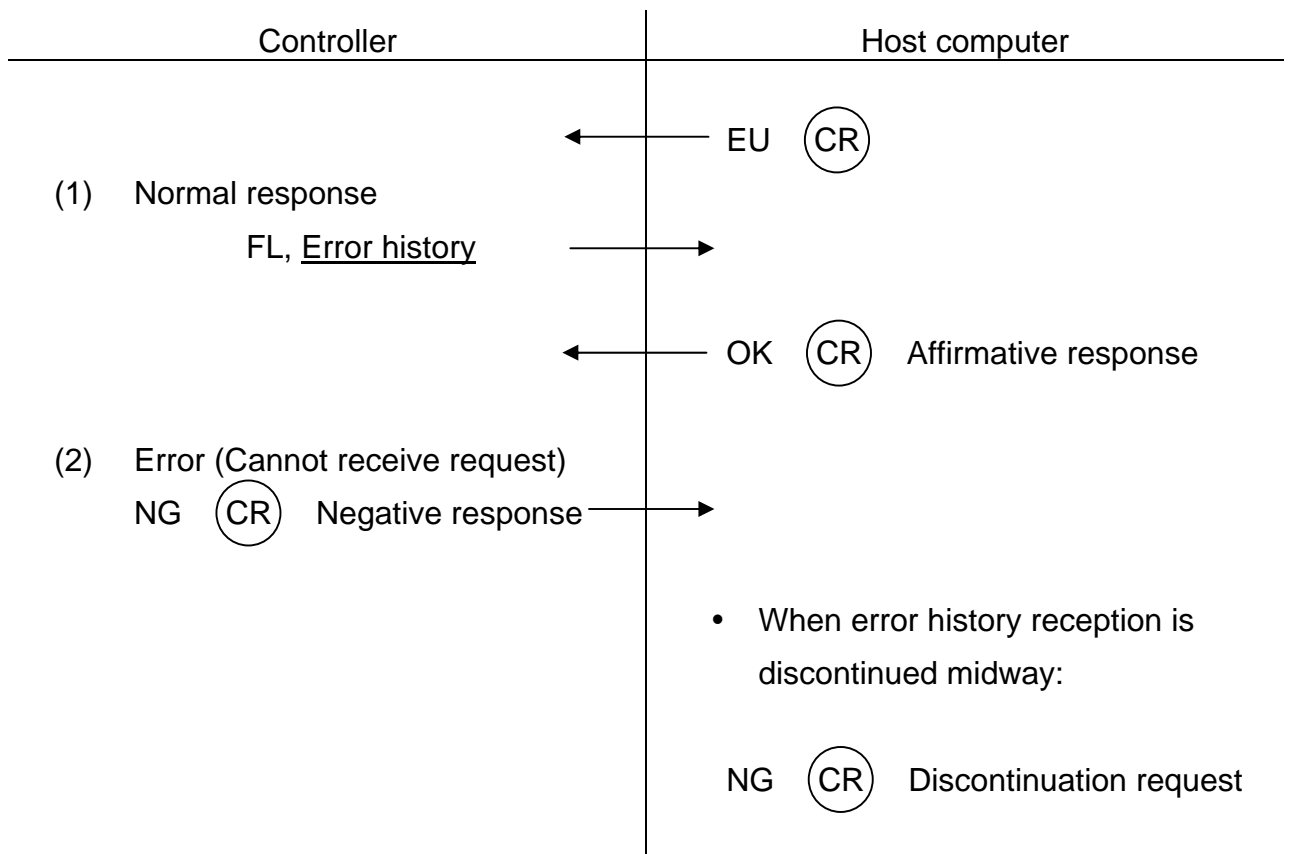
| Protocol |

Controller	Host computer

EU   (CR)

(1)   Normal response

FL, Error history

OK   (CR)   Affirmative response

(2)   Error (Cannot receive request)

NG   (CR)   Negative response

• When error history reception is discontinued midway:

NG   (CR)   Discontinuation request

| Note |

Put in a delay of about 50 msec until the host computer sends an affirmative response OK to the controller after the host computer has received the error history.

: File Erase　　　　　　　　　　(Host computer → Controller)　　ER

Format

ER, <u>file name</u> (CR)

Description

The ER command is used to erase a specified file from the controller RAM drive.

Protocol

Controller	Host computer

←─────── ER, <u>file name</u> (CR)

(1)　Normal completion

OK (CR) ──────► Affirmative response

(2)　Error

(Cannot receive request)

NG (CR) ──────► Negative response (when the specified file does not exist)
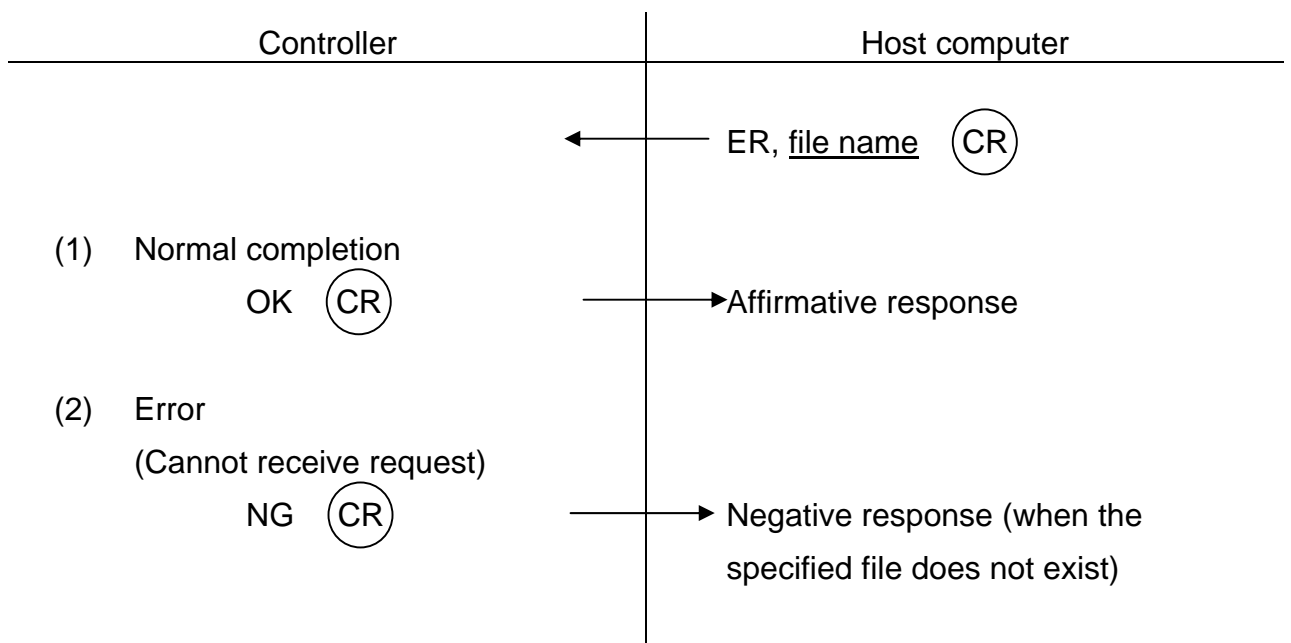
: I/O Write   (Host computer $\rightarrow$ Controller)   IW

Format

IW, Line No. ⎵ ON/OFF command   (CR)

Function

The IW command is used to instruct forced writing of up to 32 I/Os from the host computer to the controller.   For the ON/OFF command, specify "0" for OFF and "1" for ON.

Protocol

Controller	Host computer

| IW, Line No. ⎵ ON/OFF command
                                                        (CR)

(1)   Normal response
      OK   (CR)   ⟶   Normal response

(2)   Error
      (Cannot receive request)
      NG   (CR)   ⟶   Request not accepted
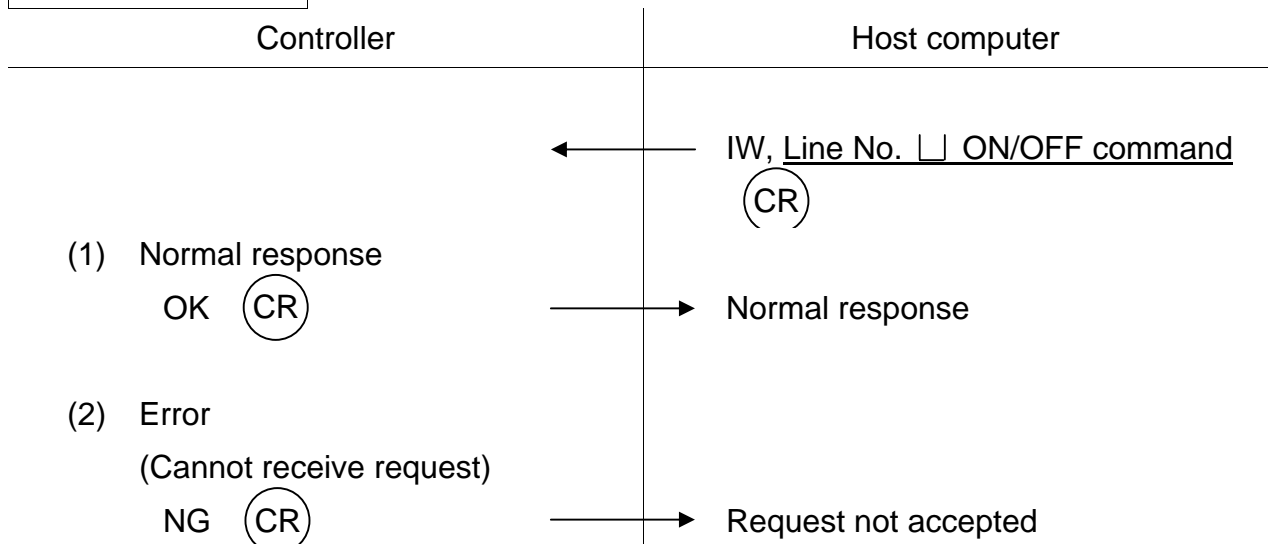
: Version read   (Host computer $\rightarrow$ Controller)   VR

| Format |

VR   (CR)

| Function |

The VR command is used to send the system version information to the host computer.

For details on the version information, see the FL (File) command.
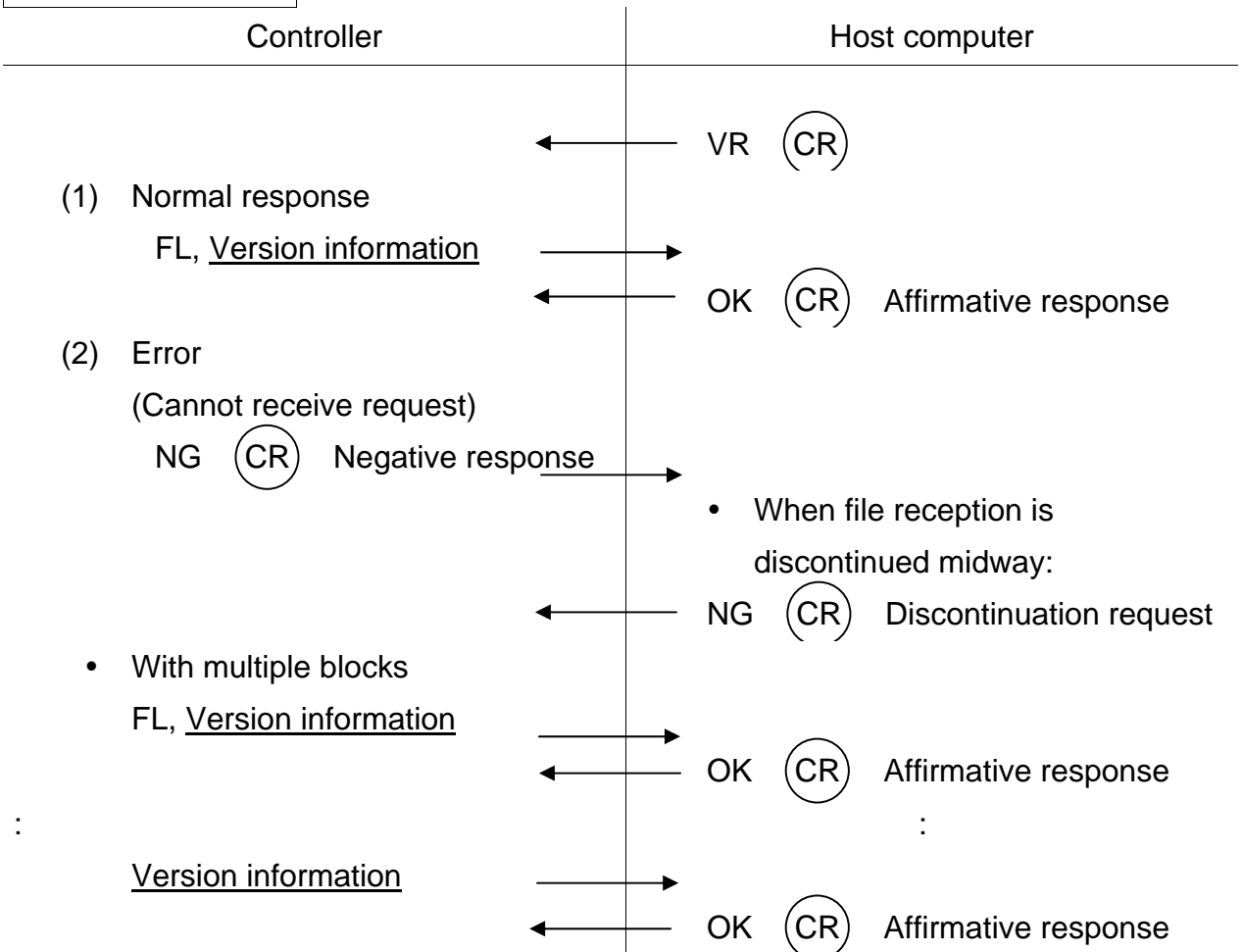
| Protocol |

Controller	Host computer

VR   (CR)

(1)   Normal response

    FL, <u>Version information</u>

                    OK   (CR)   Affirmative response

(2)   Error

    (Cannot receive request)

    NG   (CR)   Negative response

              •   When file reception is
                  discontinued midway:

               NG   (CR)   Discontinuation request

•   With multiple blocks

    FL, <u>Version information</u>

                    OK   (CR)   Affirmative response

:   :

    <u>Version information</u>

                    OK   (CR)   Affirmative response

| Note |

Put in a delay of about 50 msec until the host computer sends an affirmative response OK to the controller after the host computer has received the file contents.

: Execution of DO statement   (Host computer → Controller)   DO

Format

DO, <u>Command statement</u>   (CR)

Function

The DO command is used to directly execute a command from the host computer.
For details on the command, see the Robot Language Manual.
In the feed hold status or servo power OFF status, negative response NG is sent
back from the controller.

Protocol

Controller	Host computer
	← DO, <u>Command statement</u> (CR)
(1) Normal completion	
OK (CR)	→ Normal completion
(2) Abnormal completion	
NG (CR)	→ Request not accepted
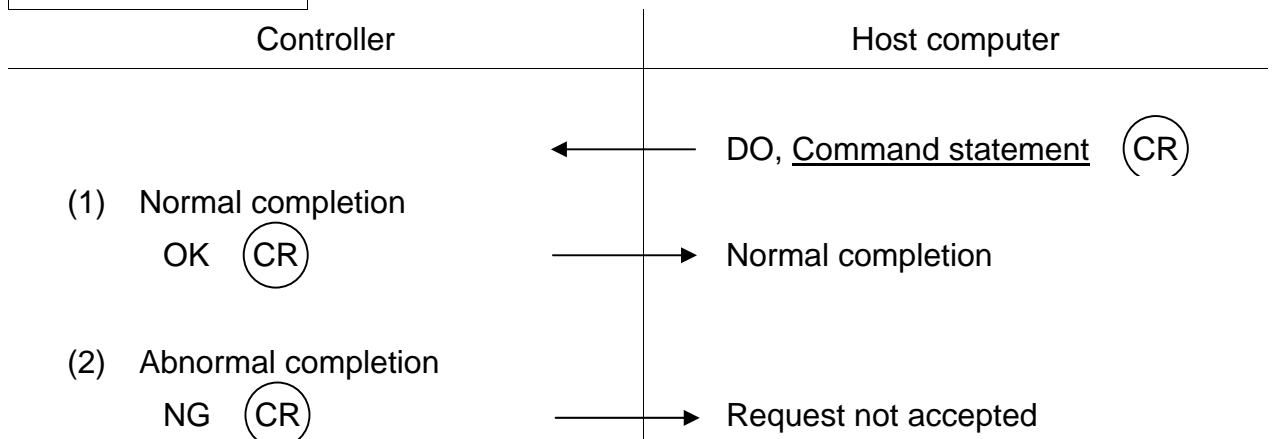
: Variable read   (Host computer → Controller)   MR

| Format |

MR, <u>Variable name ⊔ Variable type</u> (CR)

| Function |

The MR command is used to transfer the read data of global-defined variable to the host computer.   Each variable name should consist of up to ten (10) characters.
For the array variable, specify the array elements ("Variable name (*, *, ⋯)").
Specify one of the following numbers for the variable type data.

0	:	Integer type	5	:	Array integer type
1	:	Real number type	6	:	Array real number type
2	:	Load type	7	:	Array load type
3	:	Coordinate type	8	:	Array coordinate type
4	:	Position type	9	:	Array position type
-1	:	Unclarified			

For details of the read data, see the FL (File) command.
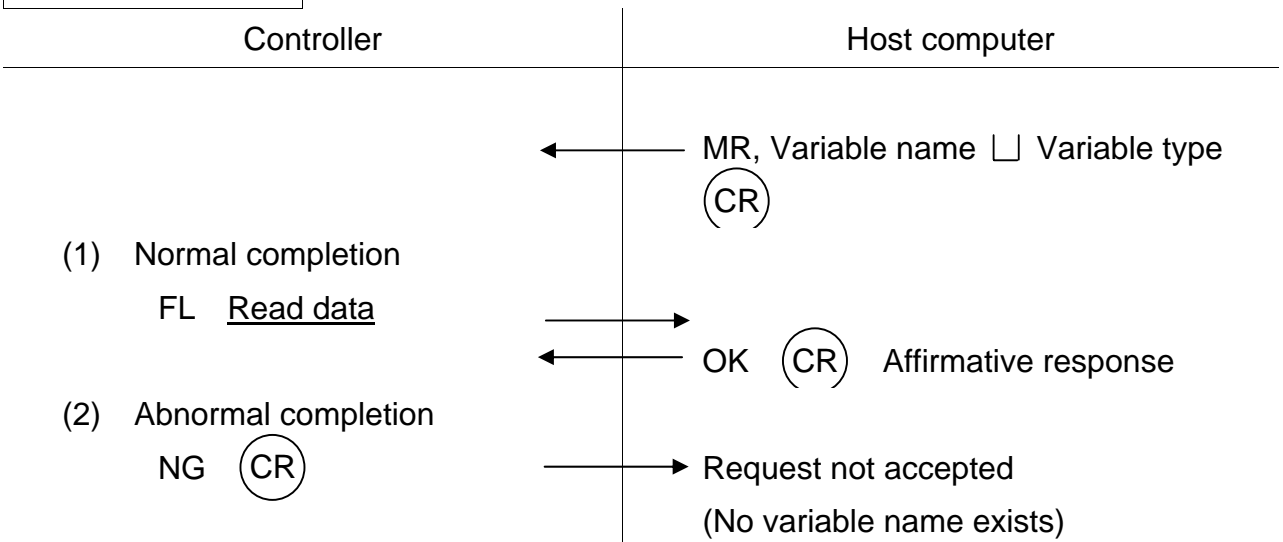
| Protocol |

Controller	Host computer

MR, Variable name ⊔ Variable type (CR)

(1) Normal completion

   FL <u>Read data</u>

   OK (CR)   Affirmative response

(2) Abnormal completion

   NG (CR)   Request not accepted

   (No variable name exists)

| Note |

Put in a delay of about 50 msec until the host computer sends an affirmative response OK to the controller after the host computer has received the file contents.

: Write global variable   (HOST → CNTL)   MW

| Format |

MW, Flag ⊔ Variable name ⊔ Variable type ⊔ Write data ⋯ (CR)

| Descriptions |

This function requests writing of data into the user's defined variable.

The flag is a control flag for restoring or non-restoring of data in the program file, as shown below.

Flag	Description
0	Restoring of data in the program file is not executed.
1	Restoring of data in the program file is executed.

The variable name should consist of up to ten (10) characters.   Specify the array element ("Variable name (*, *, ⋯)") for the array variable.

Specify the variable type data and write data in the following manner.

Variable type	Variable type data	Write data
Integer type	0	1 pc. (long)
Real number type	1	1 pc. (float)
Load type	2	2 pcs. (float⊔float)
Coordinate type	3	4 pcs. (float⊔float⊔float⊔float)
Position type	4	6 pcs. (float⊔float⊔float⊔float⊔float⊔float)
Array integer type	5	1 pc. (long)
Array real number type	6	1 pc. (float)
Array load type	7	2 pcs. (float⊔float)
Array coordinate type	8	4 pcs. (float⊔float⊔float⊔float)
Array position type	9	6 pcs. (float⊔float⊔float⊔float⊔float⊔float)

Protocol

Controller	Host computer

MW, Flag ⊔ Variable name ⊔
Variable type ⊔ Write data (CR)

(1)　Normal completion
　　OK　(CR)

(2)　Abnormal completion
　　NG　(CR)

Request not accepted
(Absence of variable name, abnormal
data)
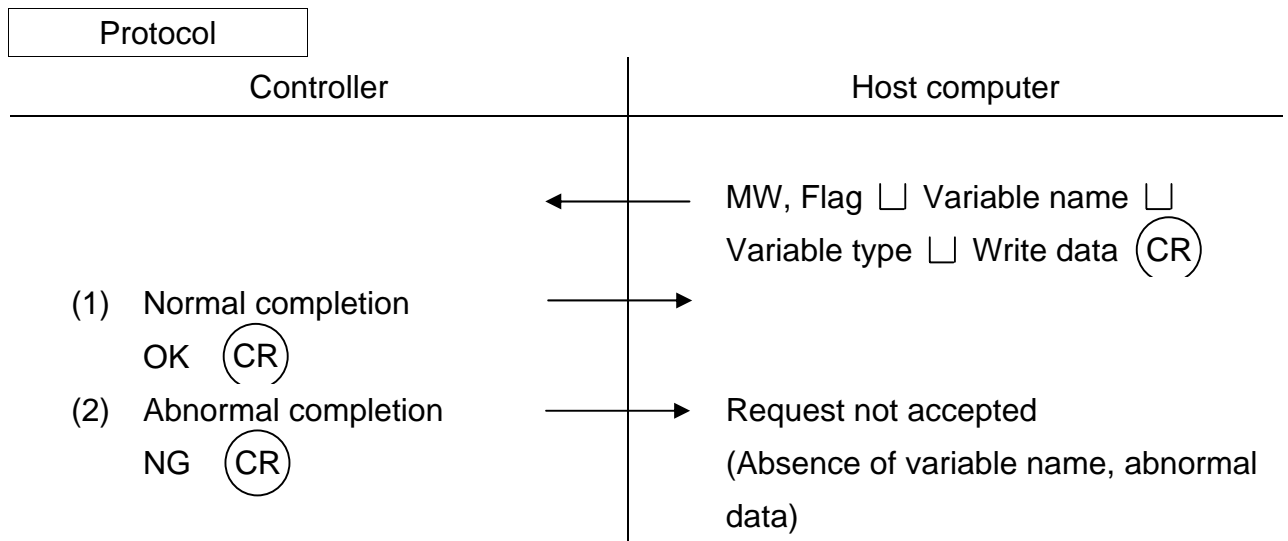
: Feed hold   (Host computer → Controller)   FD

Format

FD   (CR)

Function

The FD command is used to inform from the host computer to the controller that the FEED HOLD pushbutton switch has been pressed.

Protocol

Controller	Host computer

←────── FD   (CR)

(1)   Normal response

  OK   (CR)   ──────→ Normal response

(2)   Error

  (Cannot receive request)

  NG   (CR)   ──────→ Request not accepted
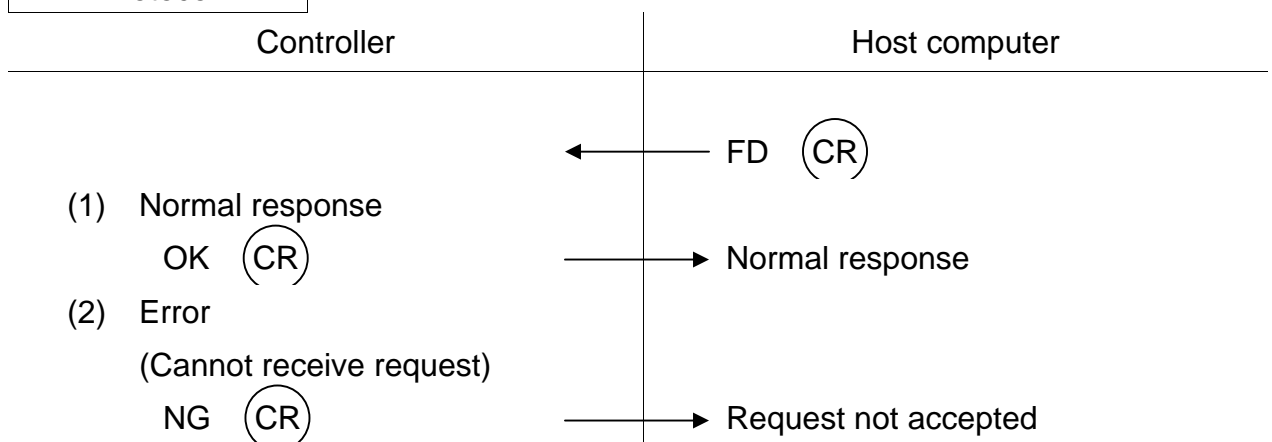
: Acknowledge　　　　　　　　　　　(Host computer ↔ Controller)　　　OK

Format

OK　(CR)

Description

The OK (Acknowledge) command indicates an affirmative response.

: Non-acknowledge   (Host computer ↔ Controller)   NG

Format

NG (CR)

Description

The NG (Non-acknowledge) command indicates a negative response.

: Internal Command   (Host computer → Controller)   EC

Format

EC, <u>Command</u> (CR)

Description

The internal command can be executed from the host computer.

The executable internal commands are as follows.

    MODE□CONT
    MODE□CYCLE
    MODE□STEP
    MODE□SEG
    OVRD□Set value (1 - 100)
    BREAK

Note: "□" signifies a space.

Protocol

Controller	Host computer

                    EC, <u>Command</u> (CR)   ←

(1)   Normal response

        OK (CR)   →

(2)   Error (Cannot receive request)

        NG (CR)   →
        Negative response
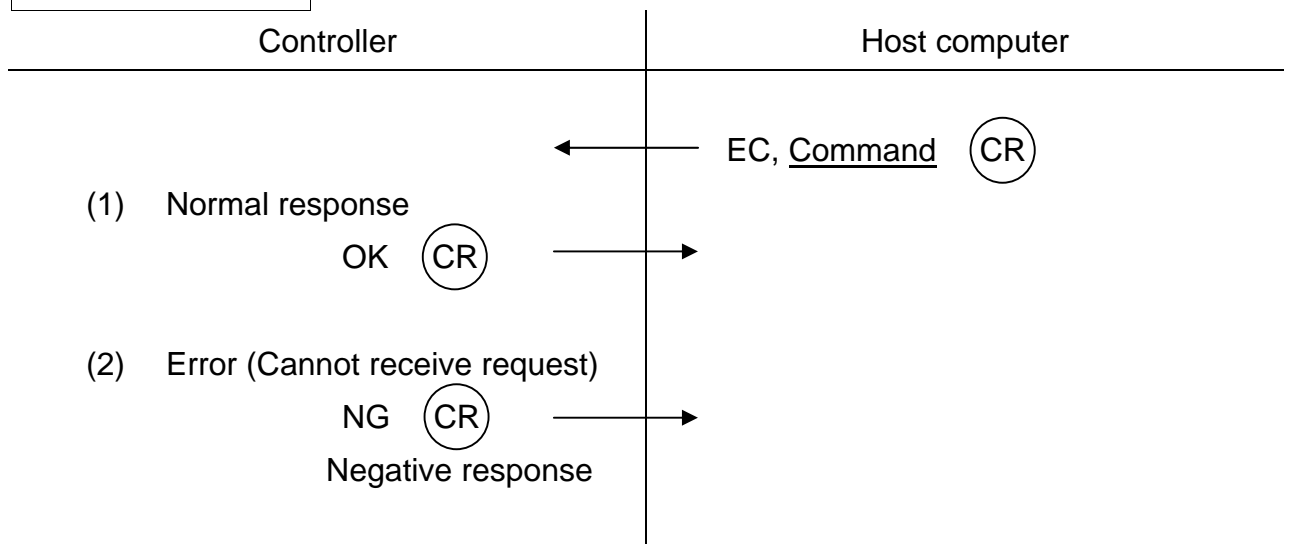
: File   (Host computer ↔ Controller)   FL

Format

FL, file contents EOF

- If the data exceeds one text field or 253 bytes, the data is transmitted in two or more texts, as shown below.

FL,	file contents 1		File contents 2		File contents 3	EOF

- EOF (End of File): File end code (0x1A) ⎯ 1 byte
  The STX, ETX, and SUM (if checksum exists) are attached to each text.

Descriptions

The FL command is used as a response command to the command given beforehand.

1.   File types and contents

   1.1   Types

Description	Corresponding command
RAM file	UL, DL
File directory	CA
Status	SU
Error history	EU
System total status	SF
Version information	VR
Memory read data	MR

1.2　File contents

A file is composed of one or more records.　Different kinds of files are made up of different kinds of records, each of which is shown below.

Furthermore, in the tables below, the value in the Size column is the maximum size of the data section.　"Fixed" means that the length of the data section is fixed; "Variable" means that the length is variable.

(1)　RAM file

1 record (line)
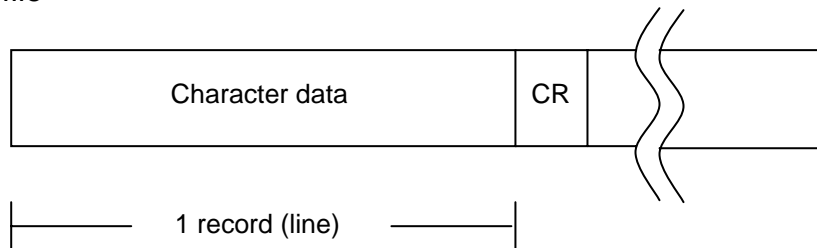
Record structure

No.	Name	Size(byte)	Description
1	Character data	Max 252 (Variable)	ASCII code alphanumeric characters and symbols
2	CR	1 (Fixed)	Record end code (0x0D)

This record corresponds to one line of a program or one line of positional data.

For information on file structure, see Para. 5.4, "RAM Files."

(2)　User file directory
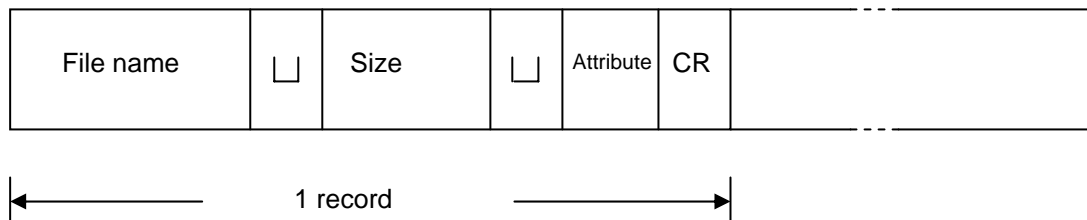
1 record

Record structure

No.	Name	Size(byte)	Description
1	File name	12(Variable)	File name in alphanumeric characters beginning with an alphabetic character.
2	Size	5(Variable)	Size of file in bytes.
3	Attribute	-(Variable)	Shows the attribution. The attribute is omitted if the attribution is not specified.
4	CR	1(Fixed)	Record end code (0x0D)

(3) Status file

Mode	⊔	File name	⊔	Speed override	⊔	Speed limit	⊔

Machine status	⊔	Execution status

No.	Name	Size(byte)	Description
1	Mode	30 (Variable)	The controller mode is shown in the following format: "MODE: (mode)/(operation mode)" (1) MODE : 5 byte fixed length (2) (mode) : Controller modes "external (sig)" : External Automatic Mode (External operation signal mode) "external (rs232c)" : External Automatic Mode (RS-232C mode) "external (ethernet)" : External Automatic Mode (Ethernet mode) "internal" : Internal Operation Mode "teaching" : Test run mode (3) / : 1 byte fixed length (4) (operation mode) : Automatic Operation Modes "step" : Step Operation Mode "continuous" : Continuous Operation Mode "cycle" : Cycle Operation Mode "segment" : Segment operation mode
2	File name	17 (Variable)	File name presently selected is shown in the following format: "FILE : (file name)" (1) FILE : 5 byte fixed length (2) (file name) : Name of the file that has been selected.
3	Speed override	11 (Variable)	The speed override is shown in the following format: "OVRD : (override) %" (1) OVRD : 5-byte fixed length (2) (override) : Percentage for the speed override. (3) % : 1 byte fixed length

No.	Name	Size (byte)	Description
4	Re-served	11 (Variable)	Reserved
5	Machine status	12 (Variable)	The machine status is shown in the following format: "MACHINE : (status)" (1) MACHINE : 8 byte fixed length (2) (status) : Robot status "free" : Status in which machine lock has been released : (Mode in which the robot can move) "lock" : Status in which machine lock is in effect. (Status in which the robot cannot move)
6	Execu-tion status	21 (Variable)	The operation status is shown in the following format: "STATUS : (execution status)" (1) STATUS : 7 byte fixed length (2) (execution status) : Operating state of the robot "running" : In Automatic Operation "stop (reset)" : In Stop (Initialization Mode) (Same status as that for program Reset) "stop (retry)" : In Stop (Retry Mode) (Restart operation from the interrupted movement) "stop (continue)" : In Stop (Continuous Mode) (Continue program from the present step)

(4)   Error history file

Size	,	Error code	⊔	Date	⊔	Time	CR	

|← ——————————— 1 record ——————————— →|
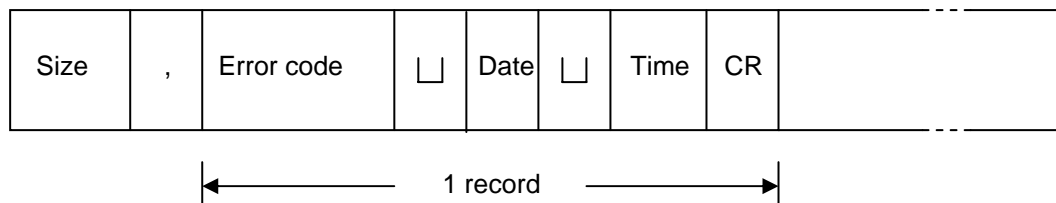
No.	Name	Size(byte)	Description
1	Size	2 (Variable)	Number (in base 10) of error histories. Placed only at the beginning of the file.
2	Error code	7 (Fixed)	The error code of an error which occurred is shown in the following format. "XXX-YYY" XXX : Main code YYY : Sub code For more information on errors, see the "Operator's Manual."
3	Date	8 (Fixed)	The date on which the error occurred is shown in the following format. "YY-MM-DD" YY : Year (Last two numbers only) MM : Month DD : Day
4	Time	8 (Fixed)	The date on which the error occurred is shown in the following format. "HH:SS:MM" HH : Hours (In 24-hour"military time.") MM : Minutes SS : Seconds
5	CR	1 (Fixed)	Record end code (0x0D)

(5) System total status

The following information is transmitted as the system total status information in the binary notation (250 bytes).

Motion status	I/O information	Current value data
52 bytes	64 bytes	134 bytes
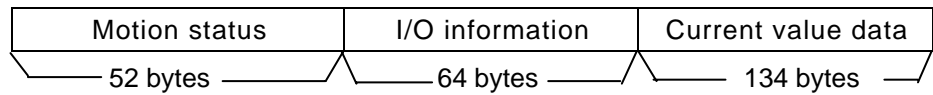
Detailed data of each group are tabled below.

I)   Motion status

No.	Name	Size (byte)	Description
1	Servo power status	1	0: OFF, 1: ON
2	EMERGENCY stop switch status	1	0: OFF, 1: ON
3	Motion status	1	0: STOP (RESET) 1: RUN 2: STOP (RETRY) 3: STOP (CONT)
4	SU command request	1	0: Without request, 1: With request
5	Current alarm information	2 × 10 pcs.	Error of level 8: 1 ~ 367 Error of level 4: 368 ~ 511 Error of level 2: 512 ~ 735 Error of level 1: 736 ~ 895 Other than above: No error
6	Program execution line	2	Line number during program execution
7	Program analysis line	2	Line number during program analysis
8	Program execution task	2	Task number during program execution
9	Program analysis task	2	Task number during program analysis
10	Feed hold status	2	0: OFF, 1: ON
11	Guidance coordinate system status	2	0: Joint, 1: Tool, 2: Work, 3: World
12	Guide rate status	2	0: Slow, 1: Mid, 2: Fast
13	Guide mode status	2	0: Jog, 1: Inching, 2: Free
14	Master mode status	2	0: TEACHING mode, 1: INTERNAL mode 2: EXT (SIG) mode, 3: EXT (HOST) mode
15	Dummy	2	Reserved
16	Power ON time	4	Unit: Min.
17	Program run time	4	Unit: Min.
Total		52	

II)   I/O information

No.	Name	Size (byte)	Description
1	General input 1	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Din1</td><td>General input</td></tr></table>

No.	Name	Size (byte)	Description		
1	General input 1	2	**Bit**	**Line No.**	**Signal name**
			0	Din1	General input
			1	Din2	General input
			2	Din3	General input
			3	Din4	General input
			4	Din5	General input
			5	Din6	General input
			6	Din7	General input
			7	Din8	General input
			8	Din9	General input
			9	Din10	General input
			10	Din11	General input
			11	Din12	General input
			12	Din13	General input
			13	Din14	General input
			14	Din15	General input
			15	Din16	General input
2	General input 2	2	**Bit**	**Line No.**	**Signal name**
			0	Din17	General input
			1	Din18	General input
			2	Din19	General input
			3	Din20	General input
			4	Din21	General input
			5	Din22	General input
			6	Din23	General input
			7	Din24	General input
			8	Din25	General input
			9	Din26	General input
			10	Din27	General input
			11	Din28	General input
			12	Din29	General input
			13	Din30	General input
			14	Din31	General input
			15	Din32	General input

No.	Name	Size (byte)	Description
3	General input 3	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Din33</td><td>General input</td></tr><tr><td>1</td><td>Din34</td><td>General input</td></tr><tr><td>2</td><td>Din35</td><td>General input</td></tr><tr><td>3</td><td>Din36</td><td>General input</td></tr><tr><td>4</td><td>Din37</td><td>General input</td></tr><tr><td>5</td><td>Din38</td><td>General input</td></tr><tr><td>6</td><td>Din39</td><td>General input</td></tr><tr><td>7</td><td>Din40</td><td>General input</td></tr><tr><td>8</td><td>Din41</td><td>General input</td></tr><tr><td>9</td><td>Din42</td><td>General input</td></tr><tr><td>10</td><td>Din43</td><td>General input</td></tr><tr><td>11</td><td>Din44</td><td>General input</td></tr><tr><td>12</td><td>Din45</td><td>General input</td></tr><tr><td>13</td><td>Din46</td><td>General input</td></tr><tr><td>14</td><td>Din47</td><td>General input</td></tr><tr><td>15</td><td>Din48</td><td>General input</td></tr></table>
4	General input 4	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Din49</td><td>General input</td></tr><tr><td>1</td><td>Din50</td><td>General input</td></tr><tr><td>2</td><td>Din51</td><td>General input</td></tr><tr><td>3</td><td>Din52</td><td>General input</td></tr><tr><td>4</td><td>Din53</td><td>General input</td></tr><tr><td>5</td><td>Din54</td><td>General input</td></tr><tr><td>6</td><td>Din55</td><td>General input</td></tr><tr><td>7</td><td>Din56</td><td>General input</td></tr><tr><td>8</td><td>Din57</td><td>General input</td></tr><tr><td>9</td><td>Din58</td><td>General input</td></tr><tr><td>10</td><td>Din59</td><td>General input</td></tr><tr><td>11</td><td>Din60</td><td>General input</td></tr><tr><td>12</td><td>Din61</td><td>General input</td></tr><tr><td>13</td><td>Din62</td><td>General input</td></tr><tr><td>14</td><td>Din63</td><td>General input</td></tr><tr><td>15</td><td>Din64</td><td>General input</td></tr></table>

No.	Name	Size (byte)	Description
5	Extension input 1	2	

Bit	Line No.	Signal name
0	Din101	Extension input
1	Din102	Extension input
2	Din103	Extension input
3	Din104	Extension input
4	Din105	Extension input
5	Din106	Extension input
6	Din107	Extension input
7	Din108	Extension input
8	Din110	Extension input
9	Din110	Extension input
10	Din111	Extension input
11	Din112	Extension input
12	Din113	Extension input
13	Din114	Extension input
14	Din115	Extension input
15	Din116	Extension input

No.	Name	Size (byte)	Description
6	Extension input 2	2	

Bit	Line No.	Signal name
0	Din117	Extension input
1	Din118	Extension input
2	Din119	Extension input
3	Din120	Extension input
4	Din121	Extension input
5	Din122	Extension input
6	Din123	Extension input
7	Din124	Extension input
8	Din125	Extension input
9	Din126	Extension input
10	Din127	Extension input
11	Din128	Extension input
12	Din129	Extension input
13	Din130	Extension input
14	Din131	Extension input
15	Din132	Extension input

No.	Name	Size (byte)	Description
7	Extension input 3	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din133</td><td>Extension input</td></tr><tr><td>1</td><td>Din134</td><td>Extension input</td></tr><tr><td>2</td><td>Din135</td><td>Extension input</td></tr><tr><td>3</td><td>Din136</td><td>Extension input</td></tr><tr><td>4</td><td>Din137</td><td>Extension input</td></tr><tr><td>5</td><td>Din138</td><td>Extension input</td></tr><tr><td>6</td><td>Din139</td><td>Extension input</td></tr><tr><td>7</td><td>Din140</td><td>Extension input</td></tr><tr><td>8</td><td>Din141</td><td>Extension input</td></tr><tr><td>9</td><td>Din142</td><td>Extension input</td></tr><tr><td>10</td><td>Din143</td><td>Extension input</td></tr><tr><td>11</td><td>Din144</td><td>Extension input</td></tr><tr><td>12</td><td>Din145</td><td>Extension input</td></tr><tr><td>13</td><td>Din146</td><td>Extension input</td></tr><tr><td>14</td><td>Din147</td><td>Extension input</td></tr><tr><td>15</td><td>Din148</td><td>Extension input</td></tr></table>
8	Extension input 4	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din149</td><td>Extension input</td></tr><tr><td>1</td><td>Din150</td><td>Extension input</td></tr><tr><td>2</td><td>Din151</td><td>Extension input</td></tr><tr><td>3</td><td>Din152</td><td>Extension input</td></tr><tr><td>4</td><td>Din153</td><td>Extension input</td></tr><tr><td>5</td><td>Din154</td><td>Extension input</td></tr><tr><td>6</td><td>Din155</td><td>Extension input</td></tr><tr><td>7</td><td>Din156</td><td>Extension input</td></tr><tr><td>8</td><td>Din157</td><td>Extension input</td></tr><tr><td>9</td><td>Din158</td><td>Extension input</td></tr><tr><td>10</td><td>Din159</td><td>Extension input</td></tr><tr><td>11</td><td>Din160</td><td>Extension input</td></tr><tr><td>12</td><td>Din161</td><td>Extension input</td></tr><tr><td>13</td><td>Din162</td><td>Extension input</td></tr><tr><td>14</td><td>Din163</td><td>Extension input</td></tr><tr><td>15</td><td>Din164</td><td>Extension input</td></tr></table>

No.	Name	Size (byte)	Description
9	System input 1	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din201</td><td>Hand input</td></tr><tr><td>1</td><td>Din202</td><td>Hand input</td></tr><tr><td>2</td><td>Din203</td><td>Hand input</td></tr><tr><td>3</td><td>Din204</td><td>Hand input</td></tr><tr><td>4</td><td>Din205</td><td>Hand input</td></tr><tr><td>5</td><td>Din206</td><td>Hand input</td></tr><tr><td>6</td><td>Din207</td><td>Hand input</td></tr><tr><td>7</td><td>Din208</td><td>Hand input</td></tr><tr><td>8</td><td>Din210</td><td></td></tr><tr><td>9</td><td>Din210</td><td></td></tr><tr><td>10</td><td>Din211</td><td></td></tr><tr><td>11</td><td>Din212</td><td></td></tr><tr><td>12</td><td>Din213</td><td></td></tr><tr><td>13</td><td>Din214</td><td></td></tr><tr><td>14</td><td>Din215</td><td></td></tr><tr><td>15</td><td>Din216</td><td></td></tr></table>
10	System input 2	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din217</td><td>Alarm of level 8</td></tr><tr><td>1</td><td>Din218</td><td>Alarm of level 8</td></tr><tr><td>2</td><td>Din219</td><td>Alarm of level 8</td></tr><tr><td>3</td><td>Din220</td><td>Alarm of level 8</td></tr><tr><td>4</td><td>Din221</td><td>Alarm of level 4</td></tr><tr><td>5</td><td>Din222</td><td>Alarm of level 4</td></tr><tr><td>6</td><td>Din223</td><td>Alarm of level 4</td></tr><tr><td>7</td><td>Din224</td><td>Alarm of level 4</td></tr><tr><td>8</td><td>Din225</td><td>Alarm of level 2</td></tr><tr><td>9</td><td>Din226</td><td>Alarm of level 2</td></tr><tr><td>10</td><td>Din227</td><td>Alarm of level 2</td></tr><tr><td>11</td><td>Din228</td><td>Alarm of level 2</td></tr><tr><td>12</td><td>Din229</td><td>Alarm of level 1</td></tr><tr><td>13</td><td>Din230</td><td>Alarm of level 1</td></tr><tr><td>14</td><td>Din231</td><td>Alarm of level 1</td></tr><tr><td>15</td><td>Din232</td><td>Alarm of level 1</td></tr></table>

No.	Name	Size (byte)	Description
11	System input 3	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din233</td><td></td></tr><tr><td>1</td><td>Din234</td><td></td></tr><tr><td>2</td><td>Din235</td><td></td></tr><tr><td>3</td><td>Din236</td><td></td></tr><tr><td>4</td><td>Din237</td><td></td></tr><tr><td>5</td><td>Din238</td><td></td></tr><tr><td>6</td><td>Din239</td><td></td></tr><tr><td>7</td><td>Din240</td><td></td></tr><tr><td>8</td><td>Din241</td><td></td></tr><tr><td>9</td><td>Din242</td><td></td></tr><tr><td>10</td><td>Din243</td><td></td></tr><tr><td>11</td><td>Din244</td><td></td></tr><tr><td>12</td><td>Din245</td><td></td></tr><tr><td>13</td><td>Din246</td><td></td></tr><tr><td>14</td><td>Din247</td><td></td></tr><tr><td>15</td><td>Din248</td><td></td></tr></table>
12	System input 4	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Din249</td><td>STROBE</td></tr><tr><td>1</td><td>Din250</td><td>PRG_RST</td></tr><tr><td>2</td><td>Din251</td><td>STEP_RST</td></tr><tr><td>3</td><td>Din252</td><td>CYC_RST</td></tr><tr><td>4</td><td>Din253</td><td>DO_RST</td></tr><tr><td>5</td><td>Din254</td><td>ALM_RST</td></tr><tr><td>6</td><td>Din255</td><td>RUN</td></tr><tr><td>7</td><td>Din256</td><td>EX_SVON</td></tr><tr><td>8</td><td>Din257</td><td>STOP</td></tr><tr><td>9</td><td>Din258</td><td>CYCLE</td></tr><tr><td>10</td><td>Din259</td><td>LOW_SPD</td></tr><tr><td>11</td><td>Din260</td><td>BREAK</td></tr><tr><td>12</td><td>Din261</td><td>SVOFF</td></tr><tr><td>13</td><td>Din262</td><td></td></tr><tr><td>14</td><td>Din263</td><td></td></tr><tr><td>15</td><td>Din264</td><td></td></tr></table>

No.	Name	Size (byte)	Description		
13	System reservation	2	Bit	Line No.	Signal name
			0		
			1		
			2		
			3		
			4		
			5		
			6		
			7		
			8		
			9		
			10		
			11		
			12		
			13		
			14		
			15		
14	System reservation	2	Bit	Line No.	Signal name
			0		
			1		
			2		
			3		
			4		
			5		
			6		
			7		
			8		
			9		
			10		
			11		
			12		
			13		
			14		
			15		

No.	Name	Size (byte)	Description
15	System reservation	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td></tr><tr><td>2</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td></tr><tr><td>4</td><td></td><td></td></tr><tr><td>5</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td></tr><tr><td>8</td><td></td><td></td></tr><tr><td>9</td><td></td><td></td></tr><tr><td>10</td><td></td><td></td></tr><tr><td>11</td><td></td><td></td></tr><tr><td>12</td><td></td><td></td></tr><tr><td>13</td><td></td><td></td></tr><tr><td>14</td><td></td><td></td></tr><tr><td>15</td><td></td><td></td></tr></table>
16	System reservation	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td></td><td></td></tr><tr><td>1</td><td></td><td></td></tr><tr><td>2</td><td></td><td></td></tr><tr><td>3</td><td></td><td></td></tr><tr><td>4</td><td></td><td></td></tr><tr><td>5</td><td></td><td></td></tr><tr><td>6</td><td></td><td></td></tr><tr><td>7</td><td></td><td></td></tr><tr><td>8</td><td></td><td></td></tr><tr><td>9</td><td></td><td></td></tr><tr><td>10</td><td></td><td></td></tr><tr><td>11</td><td></td><td></td></tr><tr><td>12</td><td></td><td></td></tr><tr><td>13</td><td></td><td></td></tr><tr><td>14</td><td></td><td></td></tr><tr><td>15</td><td></td><td></td></tr></table>

No.	Name	Size (byte)	Description
17	General output 1	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Dout1</td><td>General output</td></tr><tr><td>1</td><td>Dout2</td><td>General output</td></tr><tr><td>2</td><td>Dout3</td><td>General output</td></tr><tr><td>3</td><td>Dout4</td><td>General output</td></tr><tr><td>4</td><td>Dout5</td><td>General output</td></tr><tr><td>5</td><td>Dout6</td><td>General output</td></tr><tr><td>6</td><td>Dout7</td><td>General output</td></tr><tr><td>7</td><td>Dout8</td><td>General output</td></tr><tr><td>8</td><td>Dout9</td><td>General output</td></tr><tr><td>9</td><td>Dout10</td><td>General output</td></tr><tr><td>10</td><td>Dout11</td><td>General output</td></tr><tr><td>11</td><td>Dout12</td><td>General output</td></tr><tr><td>12</td><td>Dout13</td><td>General output</td></tr><tr><td>13</td><td>Dout14</td><td>General output</td></tr><tr><td>14</td><td>Dout15</td><td>General output</td></tr><tr><td>15</td><td>Dout16</td><td>General output</td></tr></table>
18	General output 2	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Dout17</td><td>General output</td></tr><tr><td>1</td><td>Dout18</td><td>General output</td></tr><tr><td>2</td><td>Dout19</td><td>General output</td></tr><tr><td>3</td><td>Dout20</td><td>General output</td></tr><tr><td>4</td><td>Dout21</td><td>General output</td></tr><tr><td>5</td><td>Dout22</td><td>General output</td></tr><tr><td>6</td><td>Dout23</td><td>General output</td></tr><tr><td>7</td><td>Dout24</td><td>General output</td></tr><tr><td>8</td><td>Dout25</td><td>General output</td></tr><tr><td>9</td><td>Dout26</td><td>General output</td></tr><tr><td>10</td><td>Dout27</td><td>General output</td></tr><tr><td>11</td><td>Dout28</td><td>General output</td></tr><tr><td>12</td><td>Dout29</td><td>General output</td></tr><tr><td>13</td><td>Dout30</td><td>General output</td></tr><tr><td>14</td><td>Dout31</td><td>General output</td></tr><tr><td>15</td><td>Dout32</td><td>General output</td></tr></table>

No.	Name	Size (byte)	Description
19	General output 3	2	

Bit	Line No.	Signal name
0	Dout33	General output
1	Dout34	General output
2	Dout35	General output
3	Dout36	General output
4	Dout37	General output
5	Dout38	General output
6	Dout39	General output
7	Dout40	General output
8	Dout41	General output
9	Dout42	General output
10	Dout43	General output
11	Dout44	General output
12	Dout45	General output
13	Dout46	General output
14	Dout47	General output
15	Dout48	General output

No.	Name	Size (byte)	Description
20	General output 4	2	

Bit	Line No.	Signal name
0	Dout49	General output
1	Dout50	General output
2	Dout51	General output
3	Dout52	General output
4	Dout53	General output
5	Dout54	General output
6	Dout55	General output
7	Dout56	General output
8	Dout57	General output
9	Dout58	General output
10	Dout59	General output
11	Dout60	General output
12	Dout61	General output
13	Dout62	General output
14	Dout63	General output
15	Dout64	General output

No.	Name	Size (byte)	Description
21	Extension output 1	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Dout101</td><td>Extension output</td></tr><tr><td>1</td><td>Dout102</td><td>Extension output</td></tr><tr><td>2</td><td>Dout103</td><td>Extension output</td></tr><tr><td>3</td><td>Dout104</td><td>Extension output</td></tr><tr><td>4</td><td>Dout105</td><td>Extension output</td></tr><tr><td>5</td><td>Dout106</td><td>Extension output</td></tr><tr><td>6</td><td>Dout107</td><td>Extension output</td></tr><tr><td>7</td><td>Dout108</td><td>Extension output</td></tr><tr><td>8</td><td>Dout110</td><td>Extension output</td></tr><tr><td>9</td><td>Dout110</td><td>Extension output</td></tr><tr><td>10</td><td>Dout111</td><td>Extension output</td></tr><tr><td>11</td><td>Dout112</td><td>Extension output</td></tr><tr><td>12</td><td>Dout113</td><td>Extension output</td></tr><tr><td>13</td><td>Dout114</td><td>Extension output</td></tr><tr><td>14</td><td>Dout115</td><td>Extension output</td></tr><tr><td>15</td><td>Dout116</td><td>Extension output</td></tr></table>
22	Extension output 2	2	<table><tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr><tr><td>0</td><td>Dout117</td><td>Extension output</td></tr><tr><td>1</td><td>Dout118</td><td>Extension output</td></tr><tr><td>2</td><td>Dout119</td><td>Extension output</td></tr><tr><td>3</td><td>Dout120</td><td>Extension output</td></tr><tr><td>4</td><td>Dout121</td><td>Extension output</td></tr><tr><td>5</td><td>Dout122</td><td>Extension output</td></tr><tr><td>6</td><td>Dout123</td><td>Extension output</td></tr><tr><td>7</td><td>Dout124</td><td>Extension output</td></tr><tr><td>8</td><td>Dout125</td><td>Extension output</td></tr><tr><td>9</td><td>Dout126</td><td>Extension output</td></tr><tr><td>10</td><td>Dout127</td><td>Extension output</td></tr><tr><td>11</td><td>Dout128</td><td>Extension output</td></tr><tr><td>12</td><td>Dout129</td><td>Extension output</td></tr><tr><td>13</td><td>Dout130</td><td>Extension output</td></tr><tr><td>14</td><td>Dout131</td><td>Extension output</td></tr><tr><td>15</td><td>Dout132</td><td>Extension output</td></tr></table>

No.	Name	Size (byte)	Description
23	Extension output 3	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Dout133</td><td>Extension output</td></tr><tr><td>1</td><td>Dout134</td><td>Extension output</td></tr><tr><td>2</td><td>Dout135</td><td>Extension output</td></tr><tr><td>3</td><td>Dout136</td><td>Extension output</td></tr><tr><td>4</td><td>Dout137</td><td>Extension output</td></tr><tr><td>5</td><td>Dout138</td><td>Extension output</td></tr><tr><td>6</td><td>Dout139</td><td>Extension output</td></tr><tr><td>7</td><td>Dout140</td><td>Extension output</td></tr><tr><td>8</td><td>Dout141</td><td>Extension output</td></tr><tr><td>9</td><td>Dout142</td><td>Extension output</td></tr><tr><td>10</td><td>Dout143</td><td>Extension output</td></tr><tr><td>11</td><td>Dout144</td><td>Extension output</td></tr><tr><td>12</td><td>Dout145</td><td>Extension output</td></tr><tr><td>13</td><td>Dout146</td><td>Extension output</td></tr><tr><td>14</td><td>Dout147</td><td>Extension output</td></tr><tr><td>15</td><td>Dout148</td><td>Extension output</td></tr></table>
24	Extension output 4	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Dout149</td><td>Extension output</td></tr><tr><td>1</td><td>Dout150</td><td>Extension output</td></tr><tr><td>2</td><td>Dout151</td><td>Extension output</td></tr><tr><td>3</td><td>Dout152</td><td>Extension output</td></tr><tr><td>4</td><td>Dout153</td><td>Extension output</td></tr><tr><td>5</td><td>Dout154</td><td>Extension output</td></tr><tr><td>6</td><td>Dout155</td><td>Extension output</td></tr><tr><td>7</td><td>Dout156</td><td>Extension output</td></tr><tr><td>8</td><td>Dout157</td><td>Extension output</td></tr><tr><td>9</td><td>Dout158</td><td>Extension output</td></tr><tr><td>10</td><td>Dout159</td><td>Extension output</td></tr><tr><td>11</td><td>Dout160</td><td>Extension output</td></tr><tr><td>12</td><td>Dout161</td><td>Extension output</td></tr><tr><td>13</td><td>Dout162</td><td>Extension output</td></tr><tr><td>14</td><td>Dout163</td><td>Extension output</td></tr><tr><td>15</td><td>Dout164</td><td>Extension output</td></tr></table>

No.	Name	Size (byte)	Description
25	System output 1	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Dout201</td><td>Hand output</td></tr><tr><td>1</td><td>Dout202</td><td>Hand output</td></tr><tr><td>2</td><td>Dout203</td><td>Hand output</td></tr><tr><td>3</td><td>Dout204</td><td>Hand output</td></tr><tr><td>4</td><td>Dout205</td><td>Hand output</td></tr><tr><td>5</td><td>Dout206</td><td>Hand output</td></tr><tr><td>6</td><td>Dout207</td><td>Hand output</td></tr><tr><td>7</td><td>Dout208</td><td>Hand output</td></tr><tr><td>8</td><td>Dout210</td><td></td></tr><tr><td>9</td><td>Dout210</td><td></td></tr><tr><td>10</td><td>Dout211</td><td></td></tr><tr><td>11</td><td>Dout212</td><td></td></tr><tr><td>12</td><td>Dout213</td><td></td></tr><tr><td>13</td><td>Dout214</td><td></td></tr><tr><td>14</td><td>Dout215</td><td></td></tr><tr><td>15</td><td>Dout216</td><td></td></tr></table>
26	System output 2	2	<table><tr><td>Bit</td><td>Line No.</td><td>Signal name</td></tr><tr><td>0</td><td>Dout217</td><td>Seq. parameter</td></tr><tr><td>1</td><td>Dout218</td><td>Seq. parameter</td></tr><tr><td>2</td><td>Dout219</td><td>Seq. parameter</td></tr><tr><td>3</td><td>Dout220</td><td>Seq. parameter</td></tr><tr><td>4</td><td>Dout221</td><td>Seq. parameter</td></tr><tr><td>5</td><td>Dout222</td><td>Seq. parameter</td></tr><tr><td>6</td><td>Dout223</td><td>Seq. parameter</td></tr><tr><td>7</td><td>Dout224</td><td>Seq. parameter</td></tr><tr><td>8</td><td>Dout225</td><td></td></tr><tr><td>9</td><td>Dout226</td><td></td></tr><tr><td>10</td><td>Dout227</td><td></td></tr><tr><td>11</td><td>Dout228</td><td></td></tr><tr><td>12</td><td>Dout229</td><td></td></tr><tr><td>13</td><td>Dout230</td><td></td></tr><tr><td>14</td><td>Dout231</td><td></td></tr><tr><td>15</td><td>Dout232</td><td></td></tr></table>

No.	Name	Size (byte)	Description
27	System output 3	2	

Bit	Line No.	Signal name
0	Dout233	
1	Dout234	
2	Dout235	
3	Dout236	
4	Dout237	
5	Dout238	
6	Dout239	
7	Dout240	
8	Dout241	
9	Dout242	
10	Dout243	
11	Dout244	
12	Dout245	
13	Dout246	
14	Dout247	
15	Dout248	

No.	Name	Size (byte)	Description
28	System output 4	2	

Bit	Line No.	Signal name
0	Dout249	EMG_ST
1	Dout250	SV_RDY
2	Dout251	ACK
3	Dout252	TEACH
4	Dout253	INT
5	Dout254	EXTSIG
6	Dout255	EXTHOST
7	Dout256	SYS_RDY
8	Dout257	AUTORUN
9	Dout258	CYC_END
10	Dout259	LOW_ST
11	Dout260	CYC_ST
12	Dout261	BT_ALM
13	Dout262	ALARM
14	Dout263	
15	Dout264	

No.	Name	Size (byte)	Description
29	System reservation	2	

Bit	Line No.	Signal name
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

No.	Name	Size (byte)	Description
30	System reservation	2	

Bit	Line No.	Signal name
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

No.	Name	Size (byte)	Description		
31	System reservation	2	Bit	Line No.	Signal name
			0		
			1		
			2		
			3		
			4		
			5		
			6		
			7		
			8		
			9		
			10		
			11		
			12		
			13		
			14		
			15		
32	System reservation	2	Bit	Line No.	Signal name
			0		
			1		
			2		
			3		
			4		
			5		
			6		
			7		
			8		
			9		
			10		
			11		
			12		
			13		
			14		
			15		

III)   Current value data

No.	Name	Size (byte)	Description
1	Joint coordinate value	4 × 8 axes	As shown below, values are set in the order of axis 1 to axis 6.
2	World coordinate value	4 × 8 axes	
3	Work coordinate value	4 × 8 axes	
4	Work coordinate name	20	Name of work coordinate system
5	Tool coordinate name	20	Name of tool coordinate system
6	Base coordinate name	20	Name of base coordinate system
7	Reserved	2	
Total		158	

Axis 1	(float)
Axis 2	(float)
Axis 3	(float)
Axis 4	(float)
Axis 5	(float)
Axis 6	(float)

(6)   Version information

System name	␣	Date of creation (Year/Month/Day)	␣	Time of creation (Hour/Min)	␣	Checksum	CR	

|←———————————— 1 record ————————————→|

Record structure

No.	Name	Size (byte)	Description
1	System name	10	
2	Date of creation (Year/Month/Day)	10	20**/**/**
3	Time of creation (Hour/Min)	5	**.**
4	Checksum	4	
5	CR	1	Record end code (0x0D)

(7)　Variable read data

Read data 1	⊔	Read data 2	⊔	Read data 3	⊔	...

The variable read data differs as shown below with the type of variable.

Variable type data	Type	Read data
0	Integer type	1 pc.　(long)
1	Real number type	1 pc.　(float)
2	Load type	2 pcs. (float⊔float)
3	Coordinate type	4 pcs. (float⊔float⊔float⊔float)
4	Position type	6 pcs. (float⊔float⊔float⊔float⊔float⊔float)
5	Array integer type	1 pc.　(long)
6	Array real number type	1 pc.　(float)
7	Array load type	2 pcs. (float⊔float)
8	Array coordinate type	4 pcs. (float⊔float⊔float⊔float)
9	Array position type	6 pcs. (float⊔float⊔float⊔float⊔float⊔float)

The float data is expressed by a value calculated to three decimal places.

5.4 Ram Files

5.4.1 User Files (Work files)

The user files store some programs and positional data.

PROGRAM MAIN ... END	Main program or global variable (The program described at the beginning of the file becomes the main program.)
PROGRAM SUB1 ... END ... PROGRAM SUBn ... END	Subprogram (Programs other than the main program)

The portion from the PROGRAM (program name) to END becomes one program.

DATA END	Positional data (The portion from DATA to END is called a data block in which the positional data are described. The data block is placed at the end of the file.)

As shown in the above figure, the main program or global variable data, subprograms, and data block, in which the positional data are described, are placed in that order. The delimiting codes for file names and programs and the check sum are not placed. At the end of each line, a carriage return code is placed. At the end of the file, an EOF code is placed.

The characters to be used should be ASCII code alphanumeric characters and symbols.

Alphanumeric characters:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
Special symbols:
" ' ( ) + - * / , . < > = ! [ ] { } % ^ & ?

The robot controller does not distinguish between upper case and lower case characters.   Do not use Chinese characters or double width characters.   In particular, make sure that blanks are not double width characters.   Moreover, do not place a tab code in the file.

(1) Global variable

The global variable area which can be referred from all programs are placed at the head of the program.   If the global variable is not used, this area can be omitted in the program.

(2) Program

One program starts with the PROGRAM statement of the robot language and ends with the END statement.   The program described at the beginning of the file becomes the main program.   In the automatic operation mode, the program is executed from the beginning.

Each statement of the robot language is described on each line.

In one file, two or more programs can be described.

For details of the robot language, see the Robot Language Manual.

(3) Positional data

The positional data are described in the data block, following the program, at the end of the file.

The data block starts with a DATA statement and ends with an END statement.

The data block stores coordinate data and load data in addition to the positional data.

Each data is designated on each line.

The format of each data is as follows:

[1]   Positional data

POINT <Position name>=<X>,<Y>,<Z>,C>,<T>/<configuration>

<Position name>   :   Name of positional data

<X>

<Y>   :   Coordinate values of X, Y, and Z (unit: mm)

<Z>

<C>   :   Rotating angle of tool end (unit: deg)

<T>   :   Data of axis 5

<Physical orientation>   :   Configuration of the robot arm that rotates in
                             horizontal direction

                             Omitted ....Undefined
                             LEFTY .... Left shoulder
                             RIGHTY .... Right shoulder

When the values of X, Y, Z, C, and T are omitted, they are treated as 0.

[2]   Coordinate data

TRANS <Coordinate name>=<X>,<Y>,<Z>, <C>

<Coordinate name>   :   Name of coordinate data

<X>

<Y>   :   Coordinate values in directions of
                             axes X, Y, and Z (unit: mm)
<Z>

<C>   :   Rotating angle about Z axis (unit: deg)

When the values of X, Y, Z, and C are omitted, they are treated as 0.

[3] Load data
PAYLOAD <Load name>=<mass>, <center of gravity offset>

<Load name>:	Name of load data
<mass>:	Mass of the load on the end of the robot hand (unit: kg)
<center of gravity offset>:	Offset of the center of gravity of the load on the end of the robot hand from the center of the tool axis (unit: mm)

When the values of the mass and center-of-gravity offset are ignored, they are taken as 0.

[4] Designation of work coordinate system
The positional data represent the position in the work coordinate system.   The positional data are designated by the work coordinate system.
When the work coordinate system is designated, the positional data described in the next line are treated as the position in the work coordinate system.
When the work coordinate system is not designated, it is assumed that the work coordinate system accords with the world coordinate system.
The work coordinate system is described in the following format.

WORK <Coordinate name>

<Coordinate name>:	Name of coordinate data used as work coordinate system

(3) Example of user file

Examples of the user file is as follows:

A block end code (0x0A) is attached to the end of each line (or block).

File name: TEST

```
GLOBAL
A = 10                    Global variable block
END
PROGRAM SAMPLE
SPEED = 20
MOVE A1
MOVE A2
MOVE A3
SUB1
MOVE A3
MOVE A2                   Main program
MOVE A4
MOVE A5
SUB2
MOVE A5
MOVE A4
MOVE A1
END
PROGRAM SUB1
PAYLOAD = HAND 1
MOVE A11
MOVE A12                  Subprogram [SUB1]
MOVE A13
FREELOAD
END
PROGRAM SUB2
PAYLOAD = HAND 2
MOVE A21
MOVE A22
MOVE A23                  Subprogram [SUB2]
FREELOAD
END
```

Program

```
DATA
POINT A1=400,0,200,0,0/RIGHTY
POINT A2-400,200,200,0,0/RIGHTY
POINT A3=400,200,50,0,0/RIGHTY
POINT A4=400,-200,200,0,0/LEFTY
POINT A5=400,-200,50,0,0/LEFTY
TRANS W1=400,200,50,90
TRANS W2=400,-200,50,0
PAYLOAD HAND1=3,100
PAYLOAD HAND2=5,0
WORK W1
POINT A11=50,0,60,-45,0/RIGHTY
WORK W1
POINT A12=0,50,70,0,0/RIGHTY
WORK W1
POINT A13=-50,0,60,45,0/RIGHTY
WORK W2
POINT A21=-80,0,60,30,0/LEFTY
WORK W2
POINT A22=0,30,70,0,0/LEFTY
WORK W2
POINT A23=-80,0,60,-30,0/LEFTY
END
```

Position in work coordinate system = world coordinate system

Coordinate data W1
Coordinate data W2
Load data HAND1
Load data HAND2

Position in work coordinate system = W1

Position in work coordinate system = W2

Data block

5.5 TSPC Settings

TSPC corresponding to the ethernet interface is Version 1.06 and subsequent versions.
Select Communication Settings from among the options.
Select TCP/IP Connection.
To connect to the controller through TCP/IP, enter IP address and port number ([E06]
Own port no {IP0}) specified for the controller.
For the operation, see TSCP Manual.
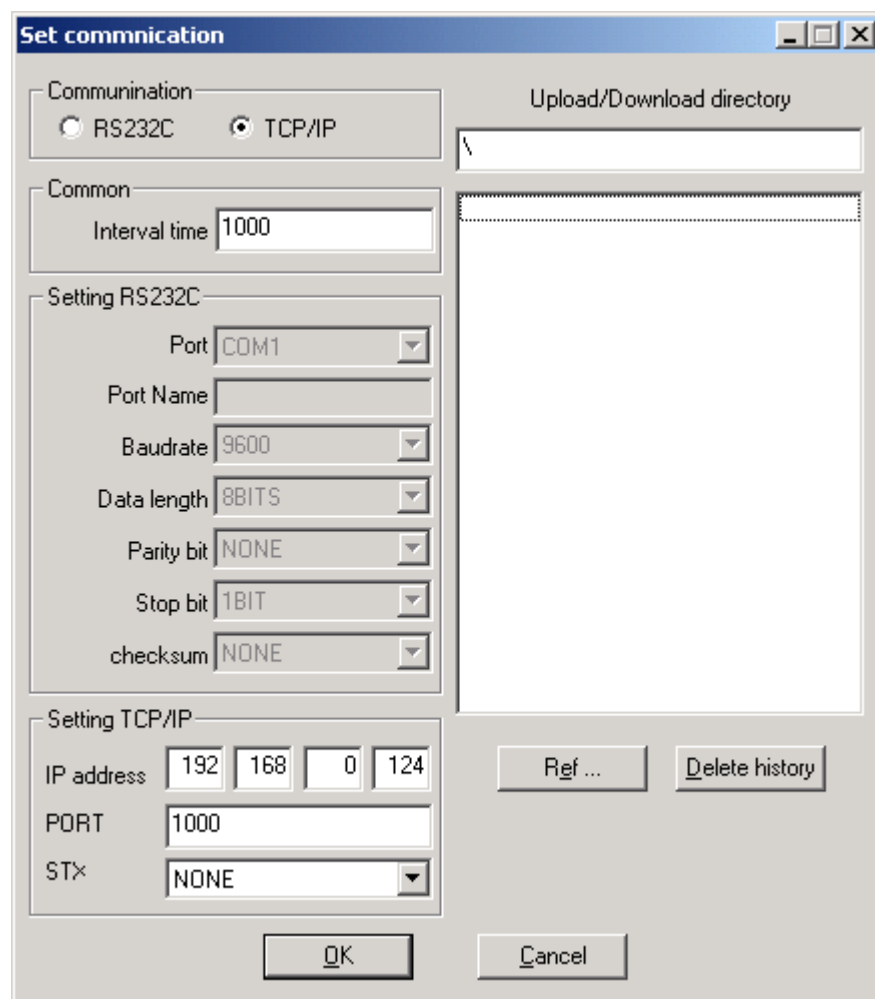
Section 6

Robot Operation Sequence Using External Communication

Operating procedures for the robot in the External Control Mode are basically the same as those in the Internal Control Mode.

6.1   Basic Operation

Figure 6.1 presents a flow sequence in which a certain block is selected and executed repeatedly.

6.2   Program Download

Figure 6.2 presents a flow sequence in which a program is downloaded at the completion of each cycle and it is executed.

6.3   Reinitializing and Starting a Stopped Program

Figure 6.3 presents a flow sequence which reinitializes and starts a program which has been stopped.
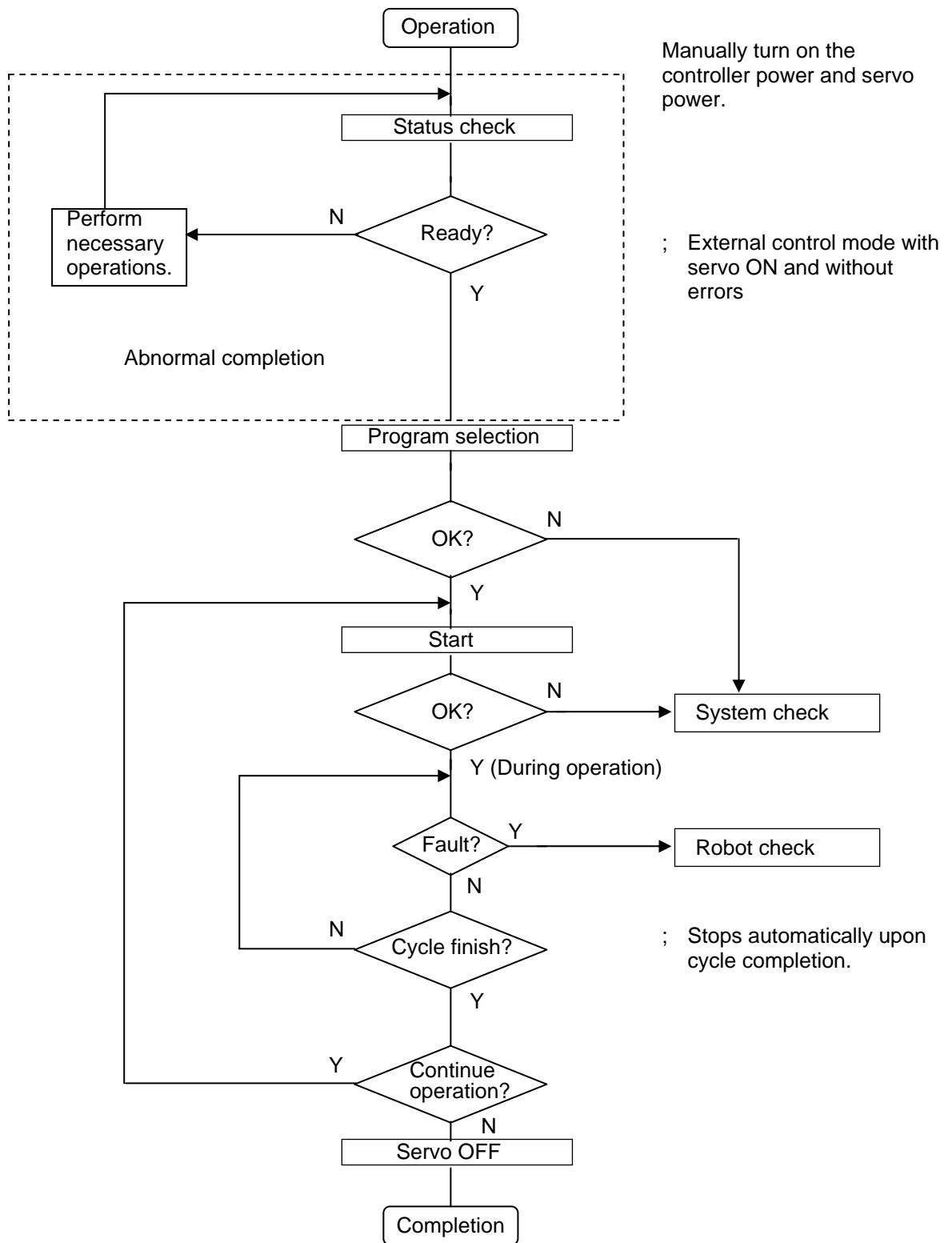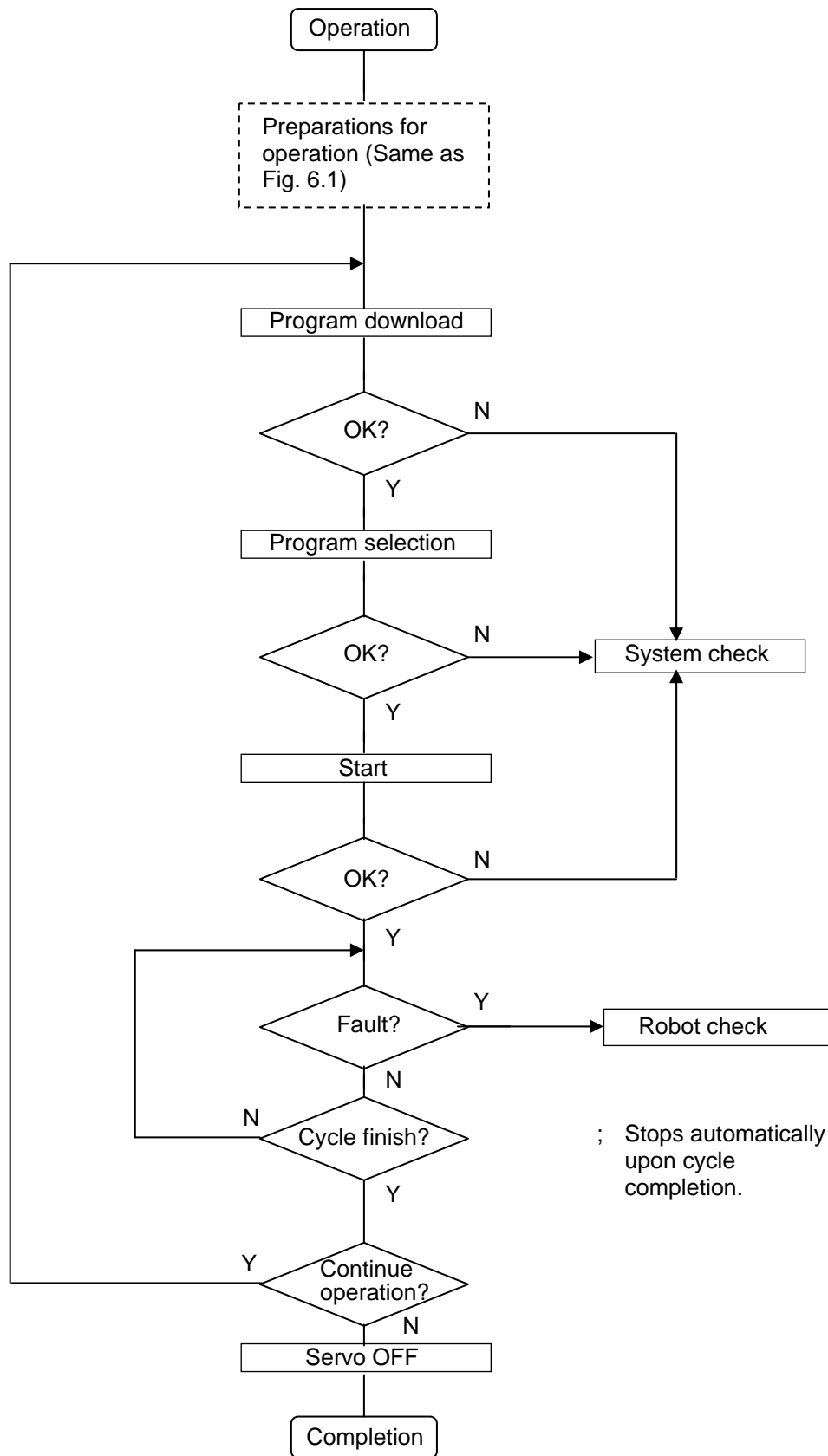
Fig. 6.1   Basic operation sequence

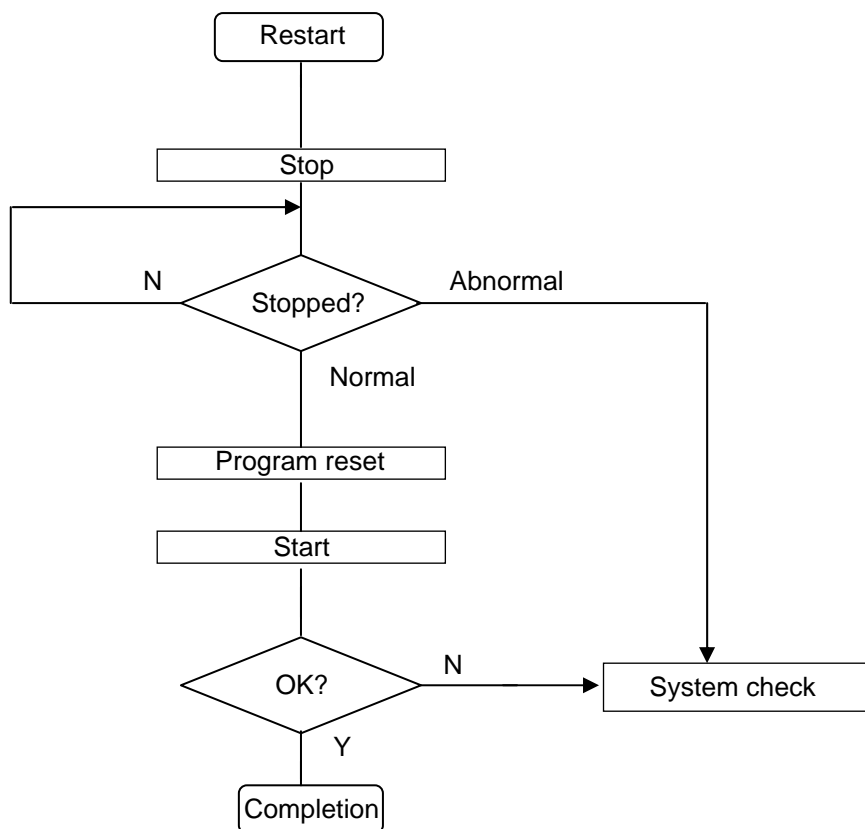Fig. 6.2   Operation sequence including program download

Fig. 6.3   Restart after program interruption

Section 7

Appendix

7.1 ASCII Code

High-order 4 bits →   Hexadecimal number

Low-order 4 bits ↓

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DE		0	@	P	`	p								
1	SH	D1	!	1	A	Q	a	q								
2	SX	D2	"	2	B	R	b	r								
3	EX	D3	#	3	C	S	c	s								
4	ET	D4	$	4	D	T	d	t								
5	EQ	NK	%	5	E	U	e	u								
6	AK	SN	&	6	F	V	f	v								
7	BL	EB	'	7	G	W	g	w								
8	BS	CN	(	8	H	X	h	x								
9	HT	EM	)	9	I	Y	i	y								
A	LF	SB	*	:	J	Z	j	z								
B	HM	EC	+	;	K	[	k	(								
C	CL	→	,	<	L	¥	l									
D	CR	←	-	=	M	]	m	)								
E	SO	↑	.	>	N	^	n	~								
F	SI	↓	/	?	O	-	o									

↑   *

Hexadecimal number

* Codes 00 to 1F correspond to control characters.   These characters will either be interpreted as a space or as a code having a specific meaning.   The meaning of these codes is shown in the above table.

– z. –