TS1000
TS2000
TS2100
Robot Controller

INSTRUCTION MANUAL

TS1000/TS2000 ROBOT CONTROLLER

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.

TOKYO, JAPAN

The information contained in this manual is subject to change without prior notice to effect improvements.

Preface

This manual describes serial 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.

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

This Manual is divided into five sections:

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

Section 2 Communication Port Specifications
This section presents information on port configurations, hardware interfaces, communication modes, etc.

Section 3 Communication via COM port
This section discusses the protocol for communication between the external device and robot program.

Section 4 Communication via HOST port
This section describes the communication protocol, commands, and robot program files as related to communication with the host computer.

Section 5 Robot Operation Sequences
This section presents examples of how to use communication commands to control robot operation.

Table of Contents

Page

Section 1

Introduction

This manual describes hardware and software related to serial interfaces between the controller and host computer. It is possible to perform the following with such serial communications:

(1) Upload and download programs and positional data;

(2) Erase programs and positional data;

(3) Select programs;

(4) Externally control the robot with the following signals:

a) Start, Stop
b) Program Reset, Step Reset, Cycle Reset and Output Signal Reset
c) Servo OFF;

(5) Monitor the status of the robot;

(6) Communicate and control in robot language including:

a) Inputting values from an external device into variables in a robot language program;
b) Outputting variables and messages from the robot language program.

(7) Monitor for robot errors:

The basic arrangement of the communication interface is shown in Figure 1.1. In this Manual, we treat only the external device and host computer.

There are two types of communication protocols. One, simple protocol, is used for file transmission and robot operation. The other, non-protocol, is used for (and only for) exchanging simple messages back and forth between the robot program and the external device.

TS1000/2000 robot controller

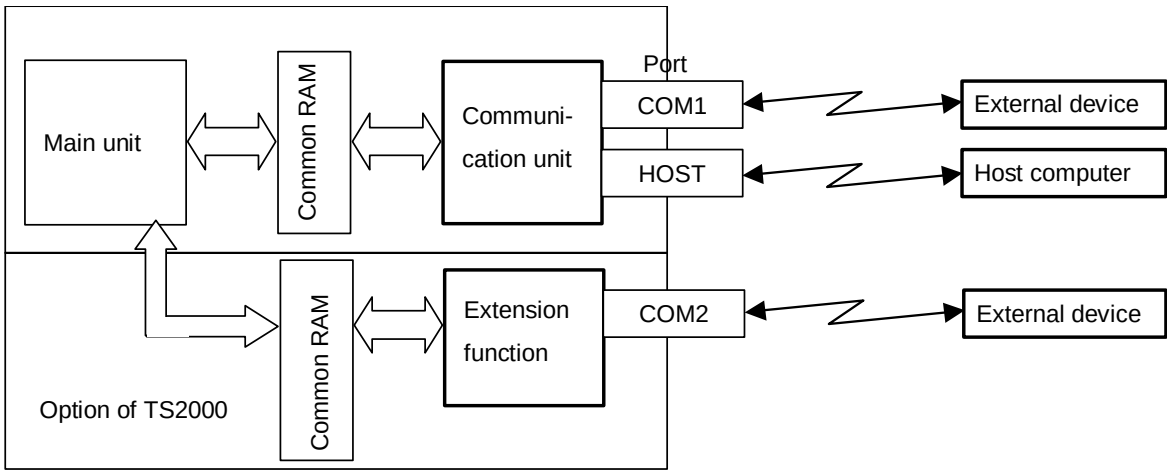


Fig. 1.1 Communication interface configuration

Section 2

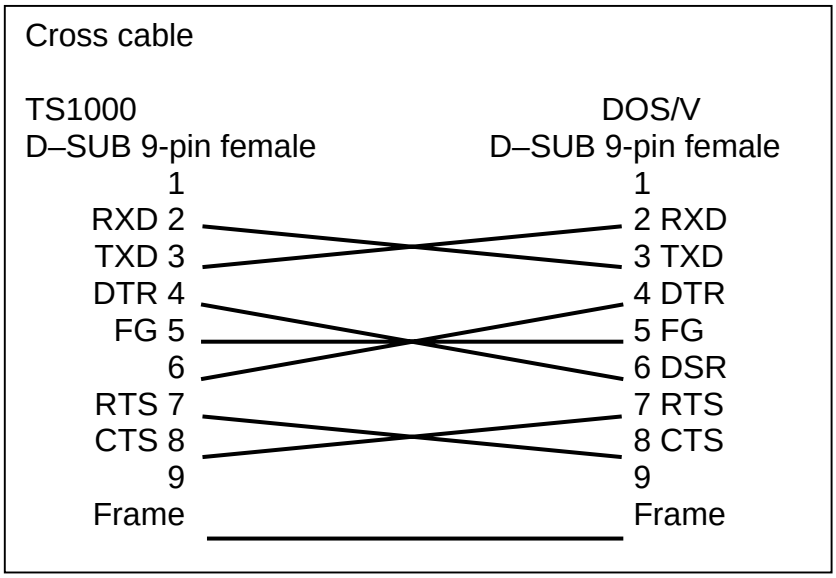
Communication Port Specifications

2.1 Port Configuration

The TS1000 controller has two (2) serial communication ports which are used for the external device and host computer. The TS2000 controller allows addition of one (1) optional serial port, in addition to the above communication port.
The COM1 and COM2 ports correspond to the external devices, and the HOST port to the host computer.

2.2 Connections

The port 1 and port 2 should be connected, using an RS232C cross cable (D-SUB; 9-pin). (For details, see the Interface Manual.)



2.3 Communication Specifications

2.3.1 COM Port

Table 2.1 COM1 port communication specifications

Item	Specification
Interface	RS232C
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 bytes	Max. 256 bytes
Data structure	Data length: 7 or 8 bits Parity: None, odd, even Stop bit: 1 or 2 bits
Protocol	Non-protocol
Timeout	No timeout
Fault recovery	No special protocols are used.

Note: The communication rate and data structure are specified in the user parameter file. (For details, see Para. 2.4.)
The COM2 port is an option of TS2000.

2.3.2 HOST Port

Table 2.2 HOST port communication specifications

Item	Specification
Interface	RS232C
Synchronous system	Start-stop synchronization system
Communication system	Half-duplex system
Communication rate	300, 600, 1200, 2400, 4800, 9600, 19200, 38400 bps
Data format	ASCII code
No. of bytes	Max. 255 bytes
Data structure	Data length: 7 or 8 bits Parity: None, odd, even Stop bit: 1 or 2 bits
Protocol	Without checksum, simple protocol
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 data structure are specified in the user parameter file. (For details, see Para. 2.4.)

2.4 Communication Parameters

The communication rate and data structure of port 2 are defined in the user parameter file.
Communication is performed according to the contents defined under [U06] of the user parameter file (USER.PAR) in the RAM drive.

TS1000

```
[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)
=    9600   8   0   1
=    38400   8   0   1
```

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

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

TS2000

```
[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
{COM2}
= 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

[COM2]

= (Speed) (Character length) (Parity) (Stop bit length) ← COM2 port setting

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

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

Example: TS1000

Set value = 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

Communication Via COM Port

Non-protocol communication is used for exchanging messages between the external device and a robot language program and for printing out files.

The SCOL robot language provides two (2) commands for data communication; PRINT and INPUT.
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.

3.1 Communication with a Robot Language Program

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

3.1.1 PRINT Command

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

PRINT␣<channel>.]{<character string>|<expression>}[,{character string>
<expression>}] ...[,CR]

<channel>:

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

COM1	: COM1 port
COM2	: COM2 port (option of TS2000)
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>:
Constants, variables, and expressions (made up of constants, variables, arithmetic operands and functions) may be specified.

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

Ex.: PRINT COM1, ”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 -digits and a decimal part with a maximum of three digits (for a maximum of eight places counting the decimal point). (For example, the number 2315.753 has a four digit integral part (2315) a three digit decimal part (753), and a decimal point for a total of eight places. 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 (20H). 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 (0DH).

3.1.2 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>] , [<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
COM2 : COM2 port (option of TS2000)
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

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 (0DH) 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 less units of data are received by the controller than was specified with the INPUT command, the controller waits until the surplus 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. 3.2, "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.

3.1.3 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"
```

3.2 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.

3.2.1 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.

Example 1)

Specifying the number of times an action is to be repeated

```
PROGRAM REPEAT
 N=0
START:
 PRINT COM1, "REQ"
 INPUT COM1, N
 IF N == 0 THEN GOTO START
 FOR K = 1 TO N
   MOVE A1
   MOVE A2
   . . .
 NEXT K
 PRINT COM1, "END"
END
```

In this program, the robot will repeat a certain series of motions as many times as the external device tells it to.

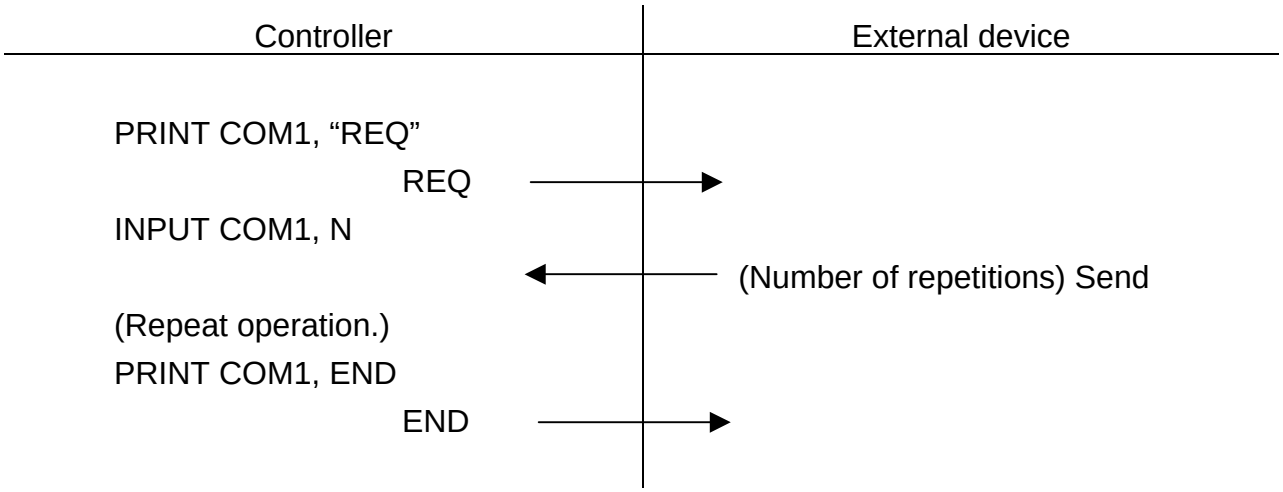
First, the controller will send the character string REQ to the external device. The external device will reply by sending back the number of times the operation is to be repeated.

The controller will read in this number as the variable N, and will use this variable in the FOR statement.

Without the IF statement, should the program execution have been suspended while the controller was waiting for data to come in, the variable N would be(mistakenly) taken as 0 when the program is resumed. However, the IF construction prevents this value from being used in the FOR loop 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



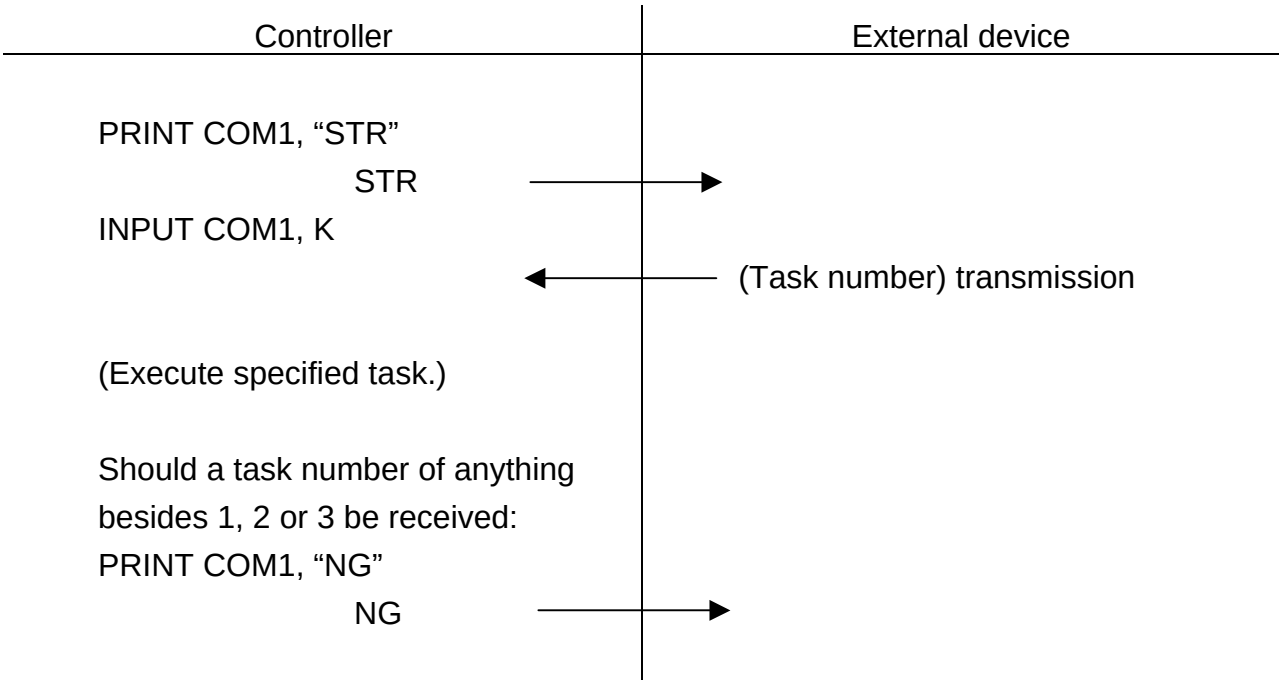
Example 2)

Selecting a task to be performed

```
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 external device.

Data exchange



3.2.2 Correcting the Position of the Robot

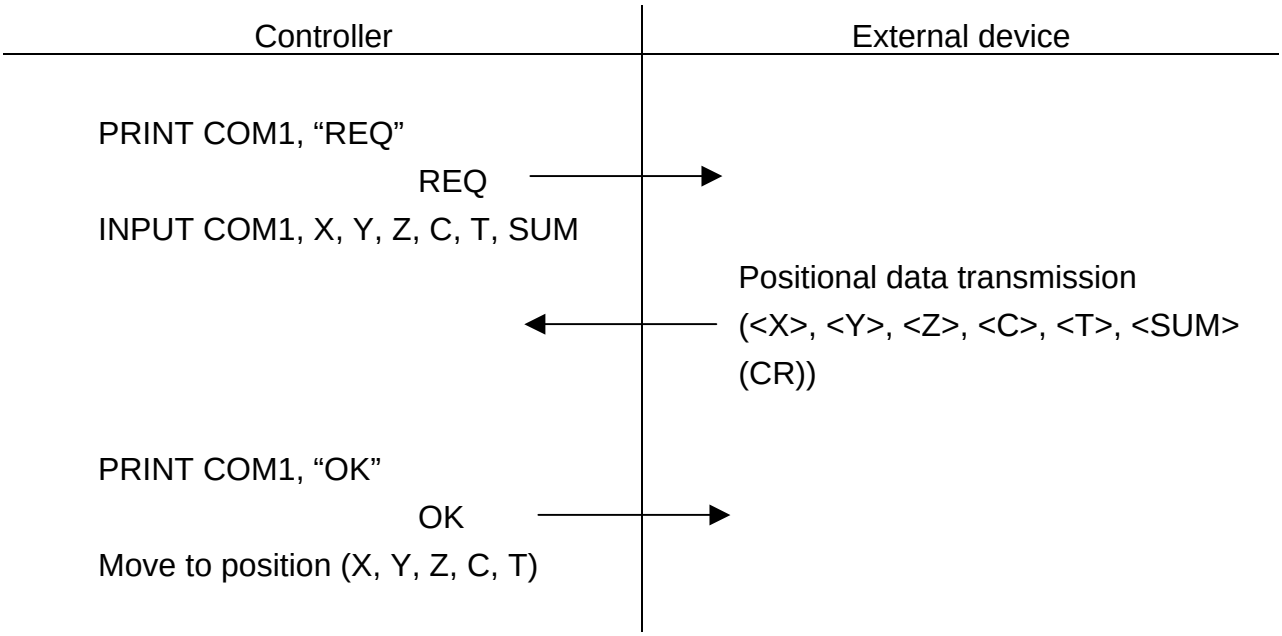
Data received from the host computer (upper level computer) can be used to correct the position of the robot.

Example 1)

```
PROGRAM SELECT
  X = 0.0
  Y = 0.0
  Z = 0.0
  C = 0.0
  T = 0.0
  PRINT COM1, "REQ"
  INPUT COM1, 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 judges that the transmission was correctly received and sends back the character string NG.
Assuming that the transmission was correctly received, the controller will create point P1 using that data and then tell the robot to move to that point.

Data exchange



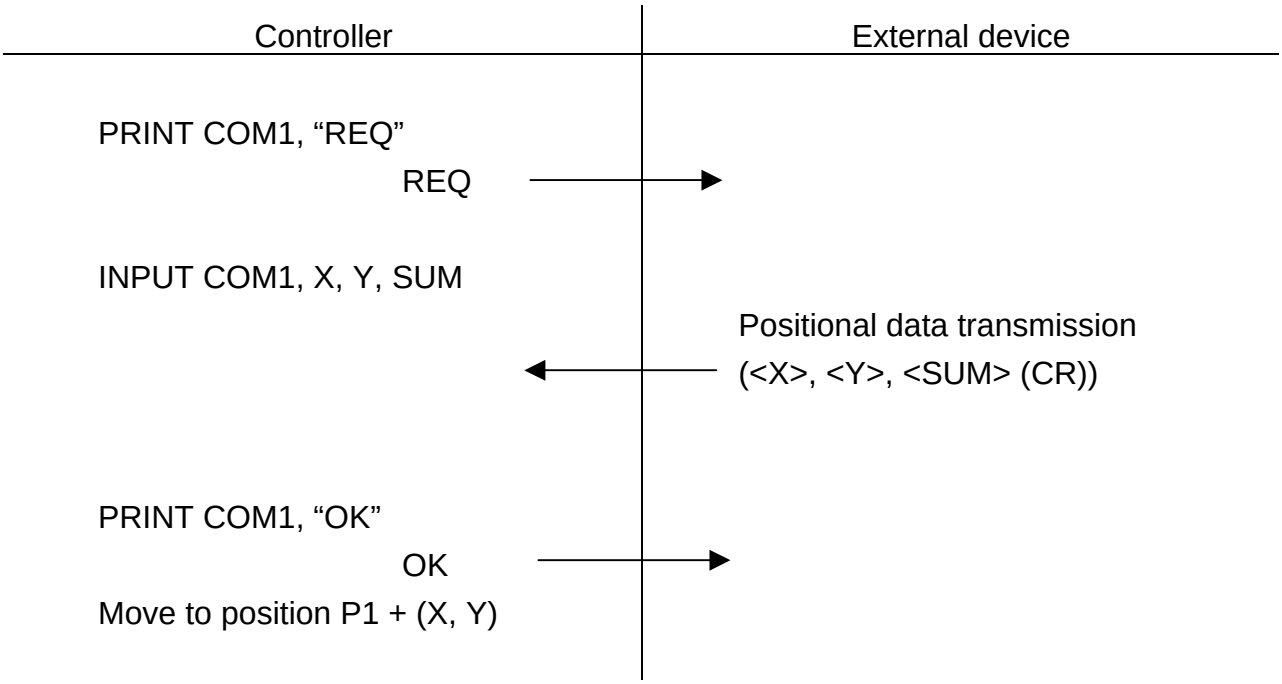
Example 2)

Specifying a relative position

```
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 judges that the transmission was correctly received and sends back the character string OK to the external device. If it does not agree, the controller sends back the character string NG.
Assuming that the transmission was correctly received, the controller will add the X and Y values of the transmission to the X and Y values of Point P1 and make the robot move to the new position thus created.

Data exchange



Section 4

Host Port Communication

4.1 Transmission Protocol

The HOST port is positioned between the host computer and robot controller. After the controller power is turned on, the port waits for requests from the host computer. 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.

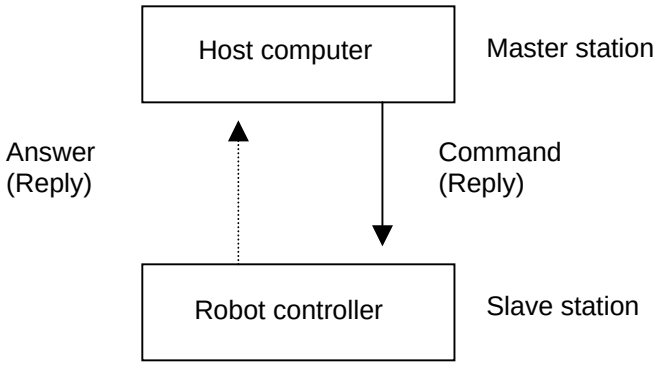


Fig. 4.1 Basic communication

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.

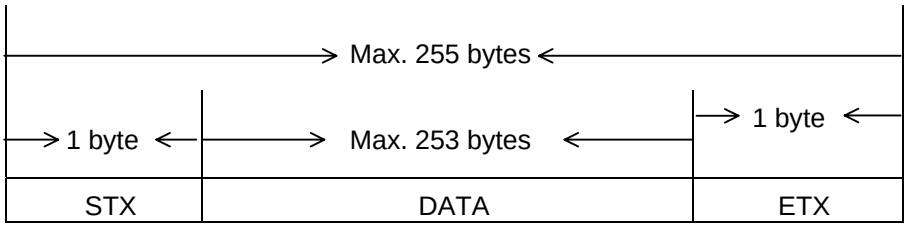
4.2 Transmission Format

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. 4.3.3. 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 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.

4.2.1 Text



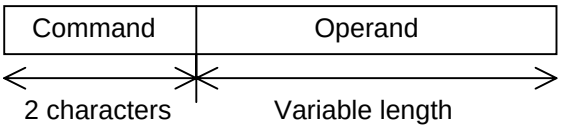
Text length: Max. 255 byte from STX through ETX

Text content:

- STX Start of text code (02H) – 1 byte
- DATA Data section (Max. 253 bytes)
- ETX End of text code (03H) – 1 byte

4.2.2 Data Section Format

The basic format of a text block is shown below.



(1) Command

Command consists of two alphabetical letters which signify the type of command. See Table 3.1 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) 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:

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

4.3 Commands

4.3.1 List of Commands

Commands which may be used with this communication protocol are shown in Tables 4.1 and 4.2.

Table 4.1 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, File name
7	UL	File name	RAM file upload	UL, File name
8	DL	File name	RAM file download	DL, File name
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, File name
13	FL	File		FL, File contents
14	OK	Acknowledge		OK
15	NG	Non-acknowledge		NG
16	EC*	Internal command	Internal command transmit	EC, Internal command
17	MD	Guide mode setting	Guide mode setting	MD, Guide mode
18	RT	Guide rate setting	Guide rate setting	RT, Guide rate
19	SC	Guidance coordinate setting	Manual guidance coordinate system setting	SC, Guidance coordinate
20	MW	Variable write	Global-defined variable write	MR, Flag Name, Type Data
21	FD	Feed hold	Feed hold	FD
22	SF	System total status	Request for fast-speed status information	SF
23	MR	Variable read	Global-defined variable read	MR, Name, Type
24	IW	I/O write	I/O forced-writing	IW, Line No., Status

No.	Command	Descriptions	Details	Text
25	VR	Version read	Robot and software version	VR
26	DO	Execution of DO statement	Execution of DO statement	DO, Statement

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

(1) MODE

(2) OVRD

(3) BREAK

Table 4.2 List of commands (Controller → Host computer)

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

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

(1) RAM files
- User file (program and position 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

4.3.2 Commands and Operation Modes

Table 4.3 shows the operation modes in which each command is operative.

Table 4.3 Commands and valid operation modes

[Host computer → controller]

Command	Descriptions	Operation mode				
		Teach	External automatic (HOST)		Internal/external automatic (SIG)	
			ON	OFF	ON	OFF
RN	Automatic operation start	X	X	O	X	X
SP	Stop	X	O	X	X	X
BR	Servo OFF	X	O	O	X	X
SO	Servo ON	X	X	O	X	X
RS	Program reset	X	X	O	X	X
	Step reset	X	X	O	X	X
	Cycle reset	X	X	O	X	X
	Output signal reset	X	X	O	X	X
	Select reset	X	X	O	X	X
SL	Program selection	X	X	O	X	X
UL	RAM file upload	X	X	O	X	O
DL	RAM file download	X	X	O	X	O
CA	Directory request	O	O	O	O	O
SU	Status request	O	O	O	O	O
ER	File erase	X	X	O	X	O
EU	Error history upload	O	O	O	O	O
EC	Internal command	X	O	O	X	X
MD	Guide mode	X	X	O	X	X
RT	Guide rate	X	X	O	X	X
SC	Guidance coordinate setting	X	X	O	X	X
MW	Variable write	O	O	O	O	O
FD	Feed hold	X	O	O	X	X
SF	System total status	O	O	O	O	O
MR	Variable read	O	O	O	O	O
I/W	I/O write	X	X	O	X	O
VR	Version read	O	O	O	O	O
DO	Execution of DO statement	X	X	O	X	X

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

Note: In the external automatic (HOST) mode, all external control input signals are invalid except for the following signals.

Slow speed command
Servo ON
Servo OFF
Emergency stop

For details, see the Interface Manual.

4.3.3 Details of Commands

: Automatic Operation Start (Host computer → controller) RN

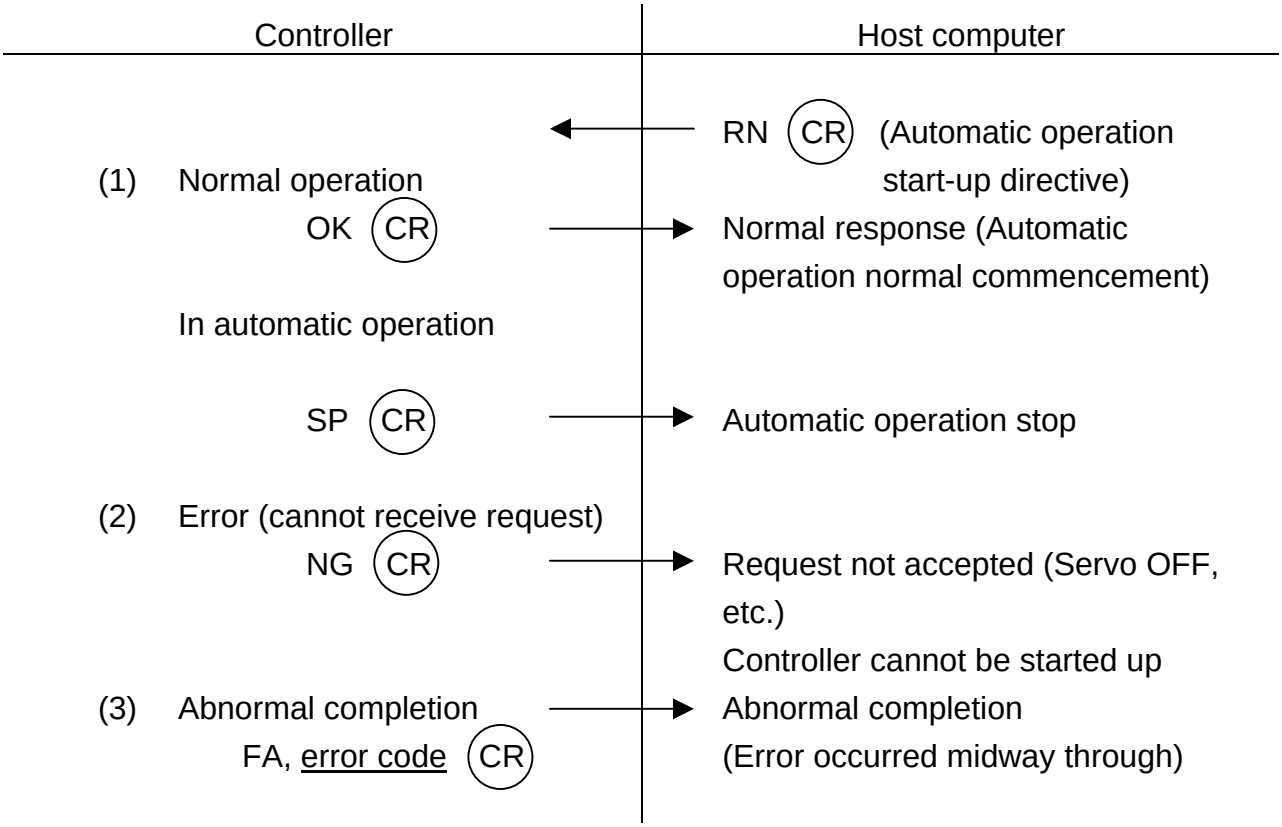
Format

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 robot 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



Note

If the power is turned on while the controller is set for the host mode (i.e., when the master mode switch on the controller panel is set to EXT), 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 (i.e., when the master mode switch is changed from INT to EXT), the operation mode of the system will be that in effect beforehand.

: Automatic Operation Stop (Host computer → Controller) SP

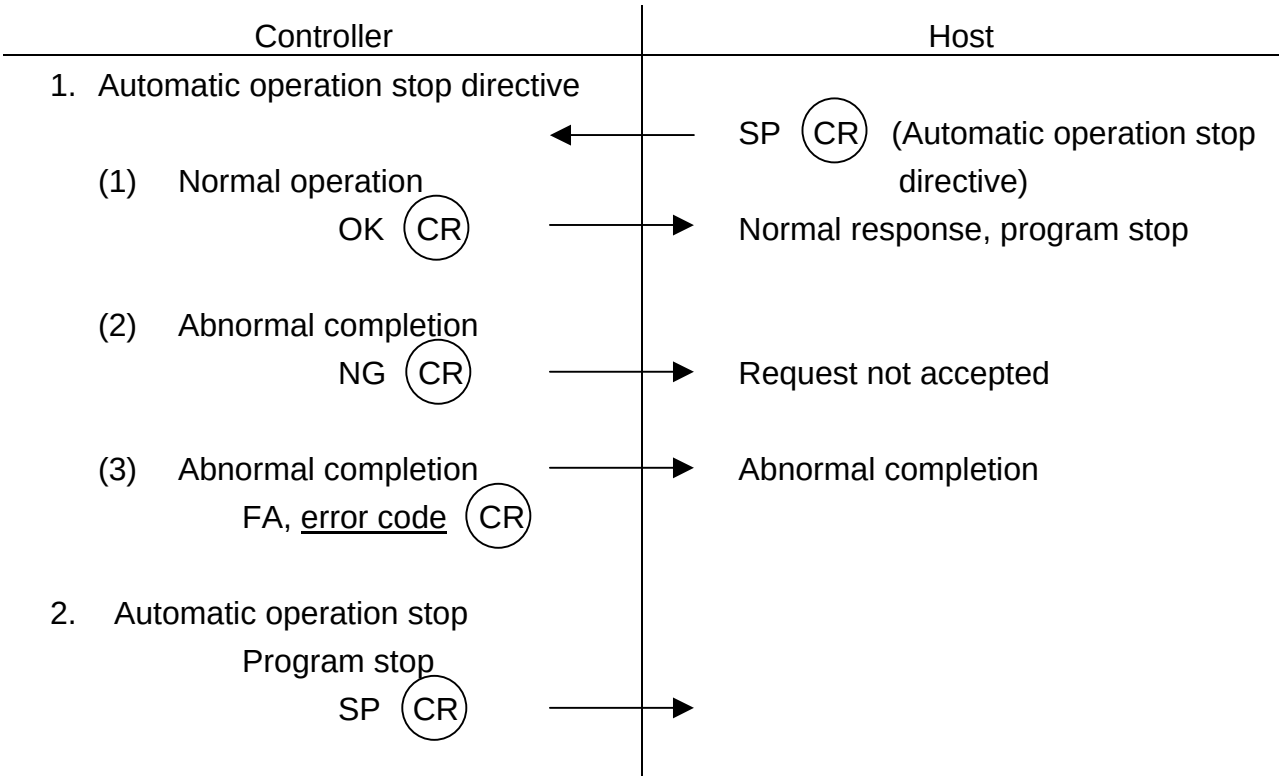
Format

SP CR

Description

The SP (Automatic Operation Stop) command is a command 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 (i.e., in the stop state). 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



: Servo OFF (Host computer → Controller) BR

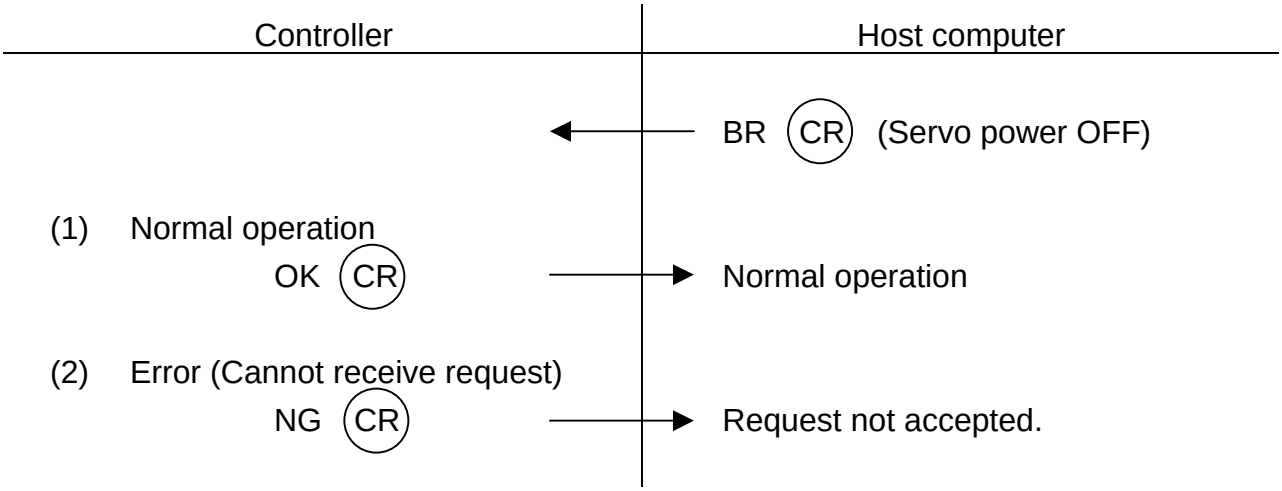
Format

BR (CR)

Description

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

Protocol



: Servo ON (Host computer → Controller) SO

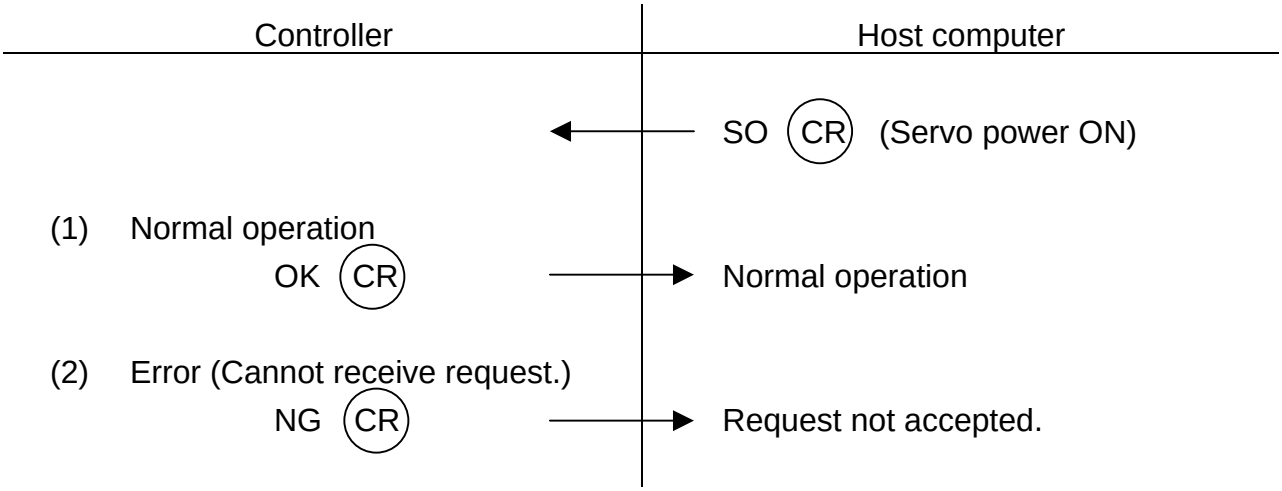
Format

SO CR

Description

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

Protocol



: 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.

CYC Reset the program back to the step marked by the label RCYCLE.

SIG Turns off the digital output signals (DO1 to DO16) 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. The list of possible suffixes to the command is presented below.

(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 reset the program to the step marked by the label RCYCLE. All program data will remain unchanged.

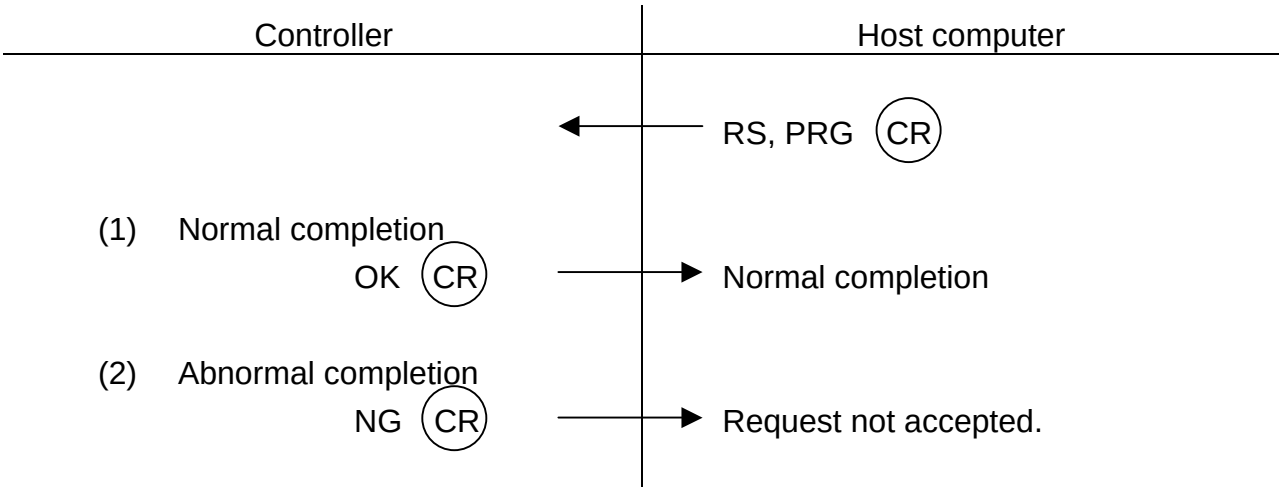
(4) SIG

SIG will put the digital outputs (DO1 to DO16) 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

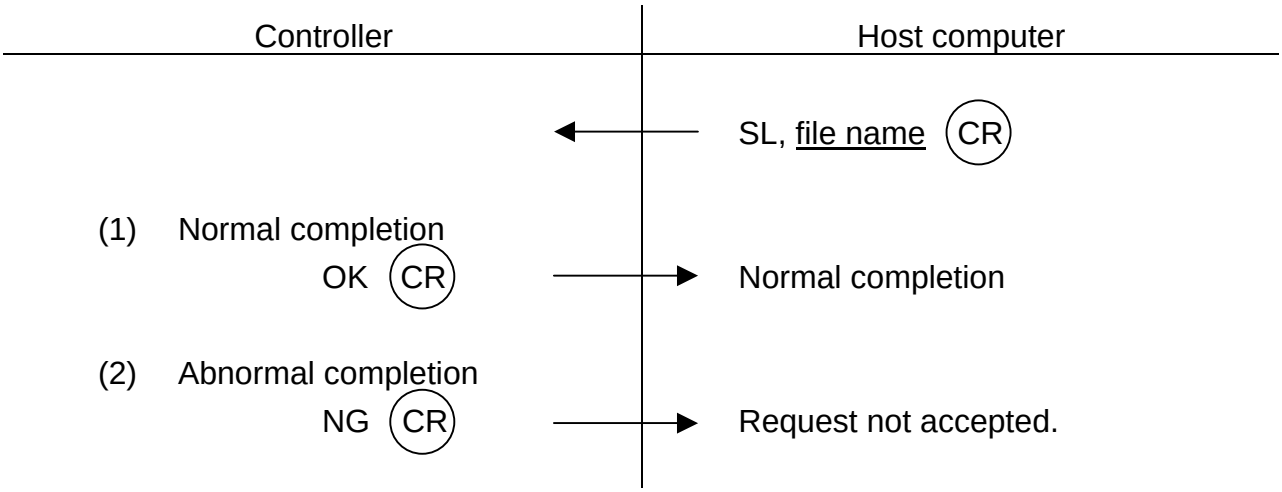
Format

SL, file name (CR)

Description

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

Protocol



: File Upload Request (Host computer → Controller) UL

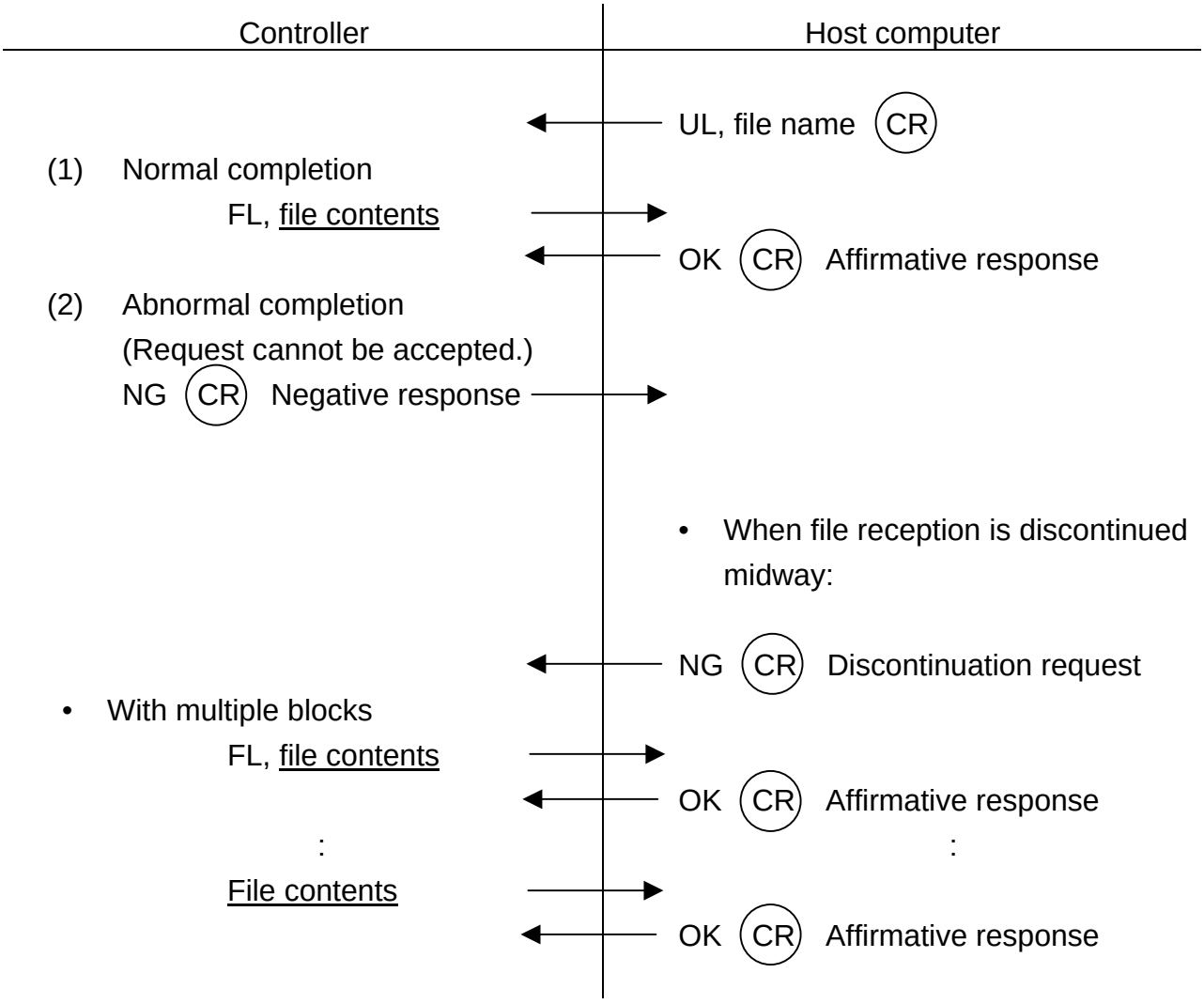
Format

UL, file name (CR)

Function

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

Protocol



Note

Put in a delay of about 50 msec to give the host computer time to send an affirmative response OK signal after having (successfully) received the file contents.

: File Download Request (Host computer → Controller) DL

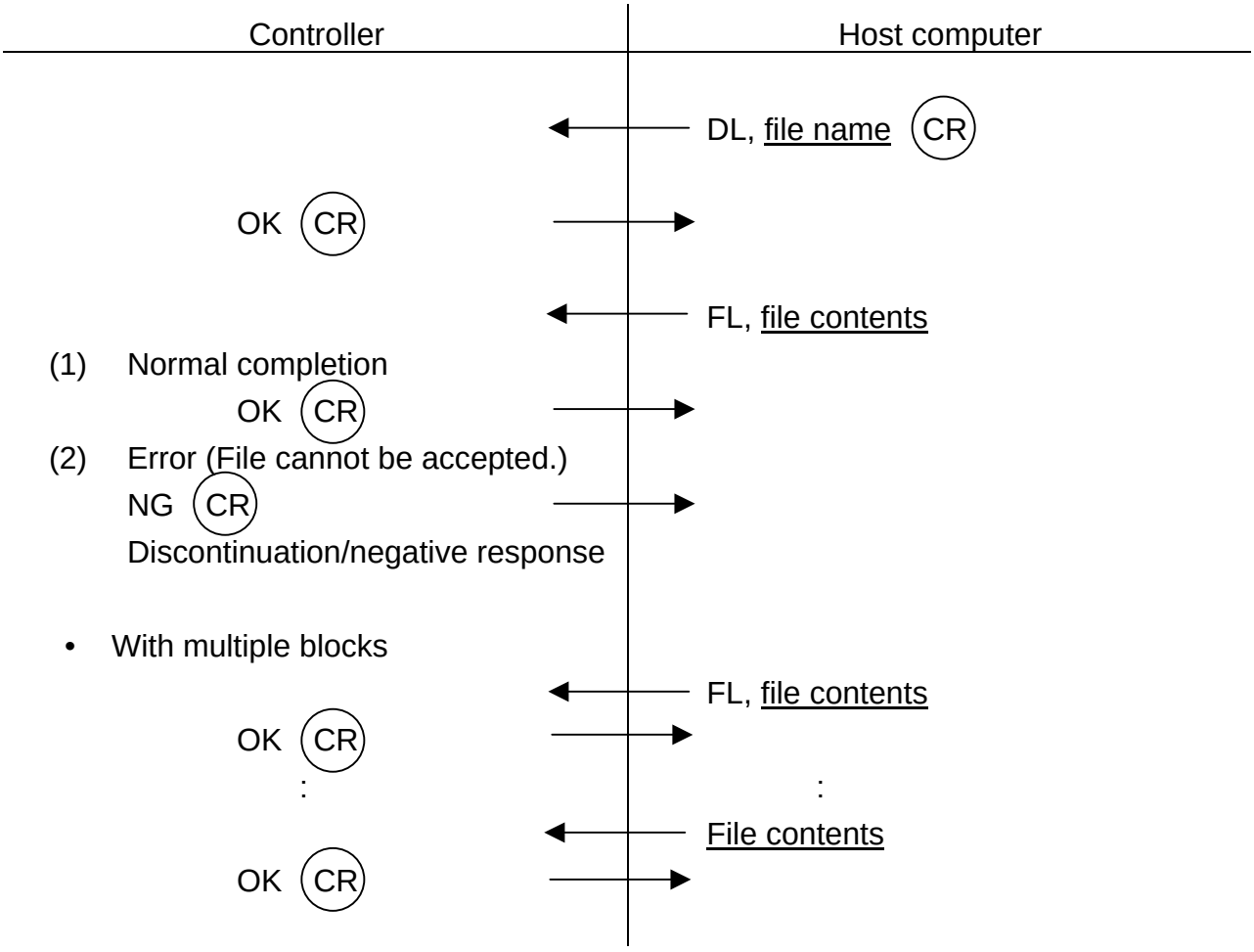
Format

DL, file name (CR)

Function

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.

Protocol



Note

Put in a delay of about 50 msec to give the host computer time to send an affirmative response OK signal after having (successfully) received the file contents.

: File Directory Request (Host computer → Controller) CA

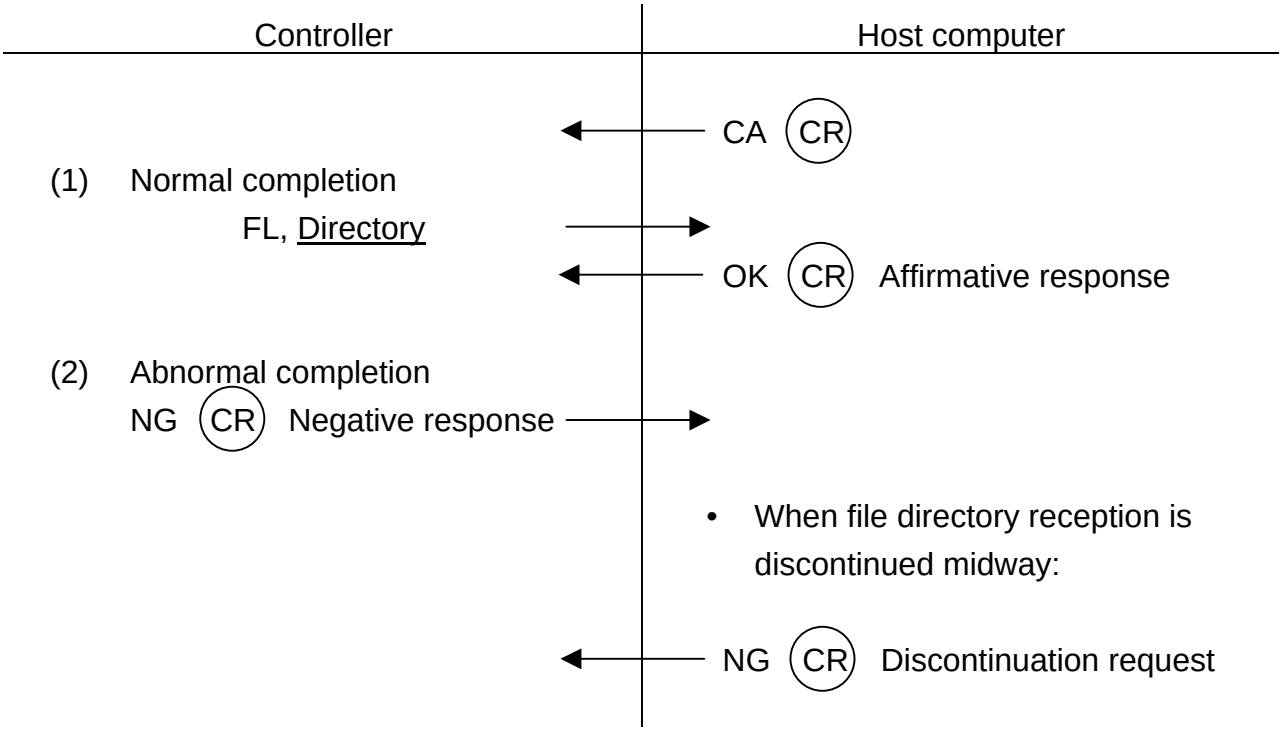
Format

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



Note

Put in a delay of about 50 msec to give the host computer time to send an affirmative response OK signal after having (successfully) received the file contents.

: Status Request	(Host computer → Controller)	SU
: System total status request	(Host computer → Controller)	SF

Format

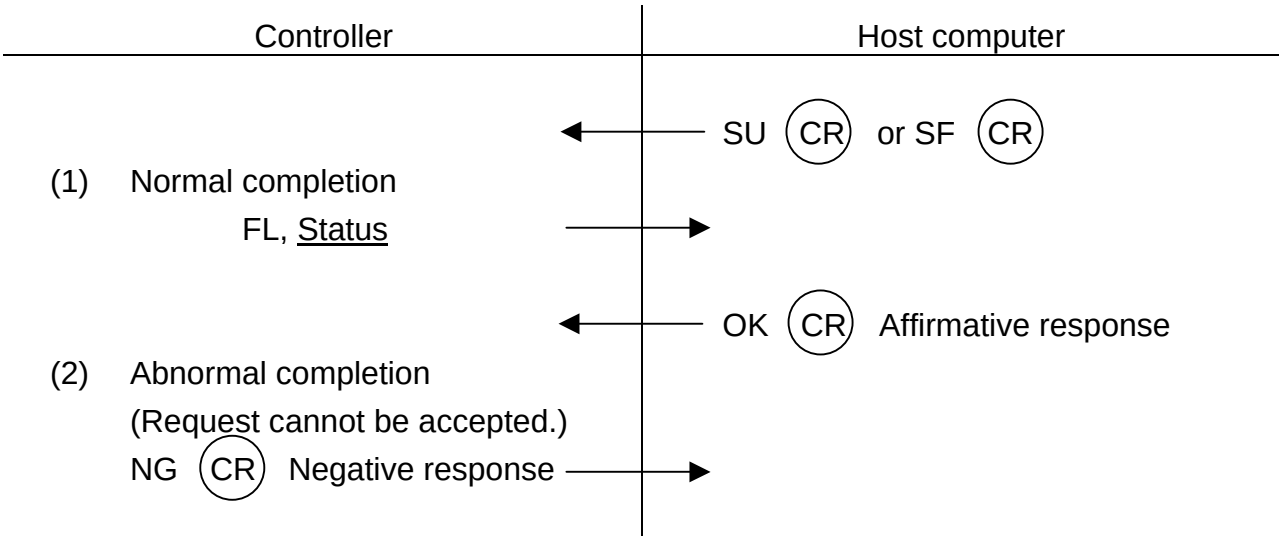
SU (CR) Status information

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



Note

Put in a delay of about 50 msec to give the host computer time to send an affirmative response OK signal after having (successfully) received the file contents.

: Error History Request (Host computer → Controller) EU

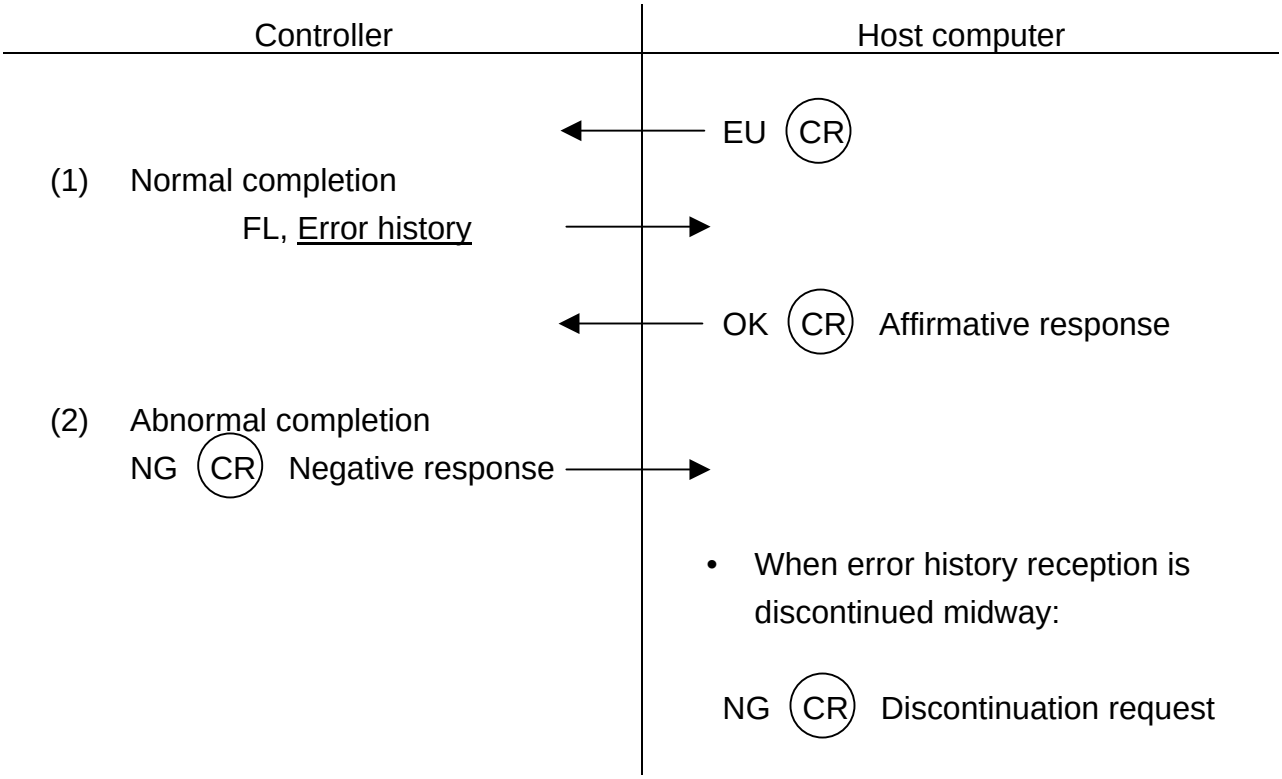
Format

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



Note

Put in a delay of about 50 msec to give the host computer time to send an affirmative response OK signal after having (successfully) received the file contents.

: File Erase (Host computer → Controller) ER

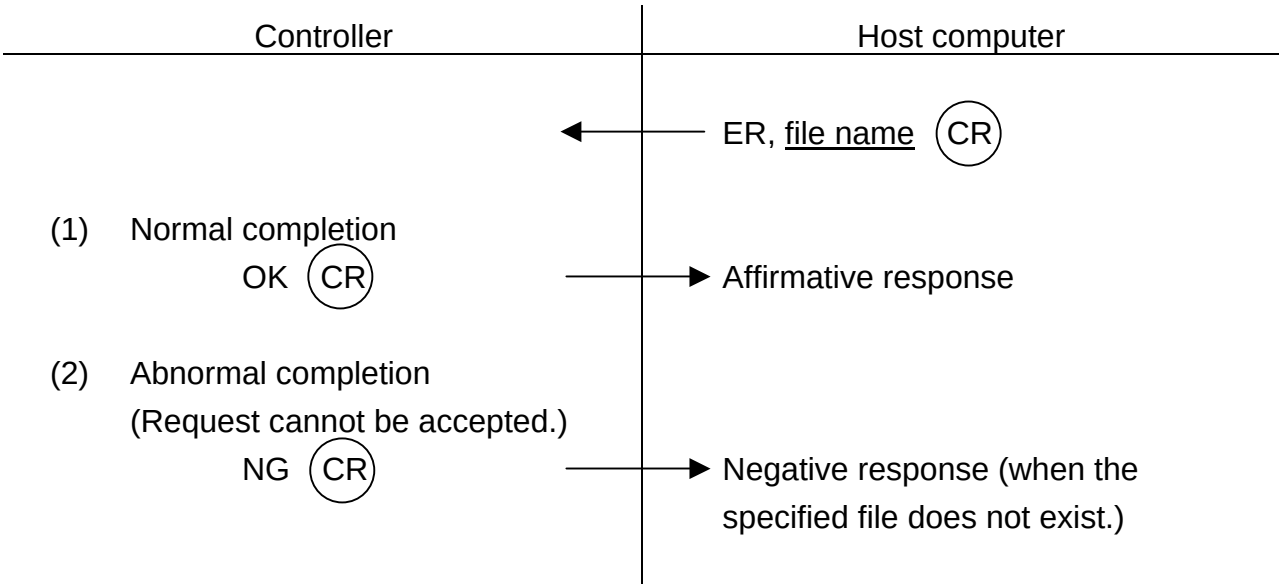
Format

ER, file name (CR)

Description

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

Protocol



: I/O Write (Host computer → Controller) IW

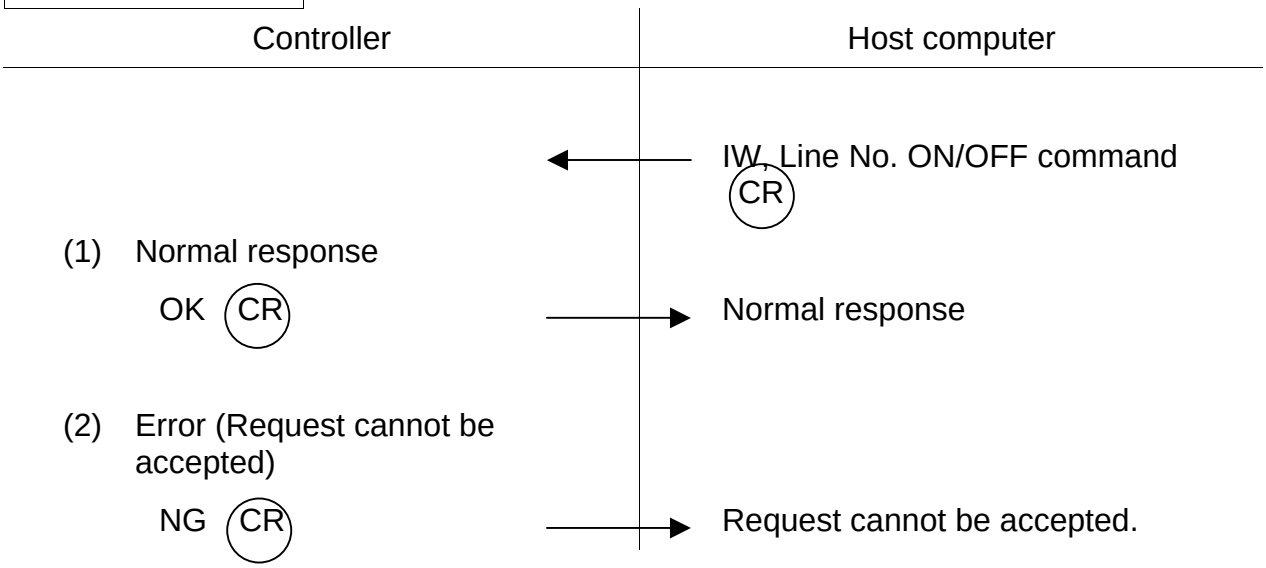
Format

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



: Version read (Host computer → Controller) VR

Format

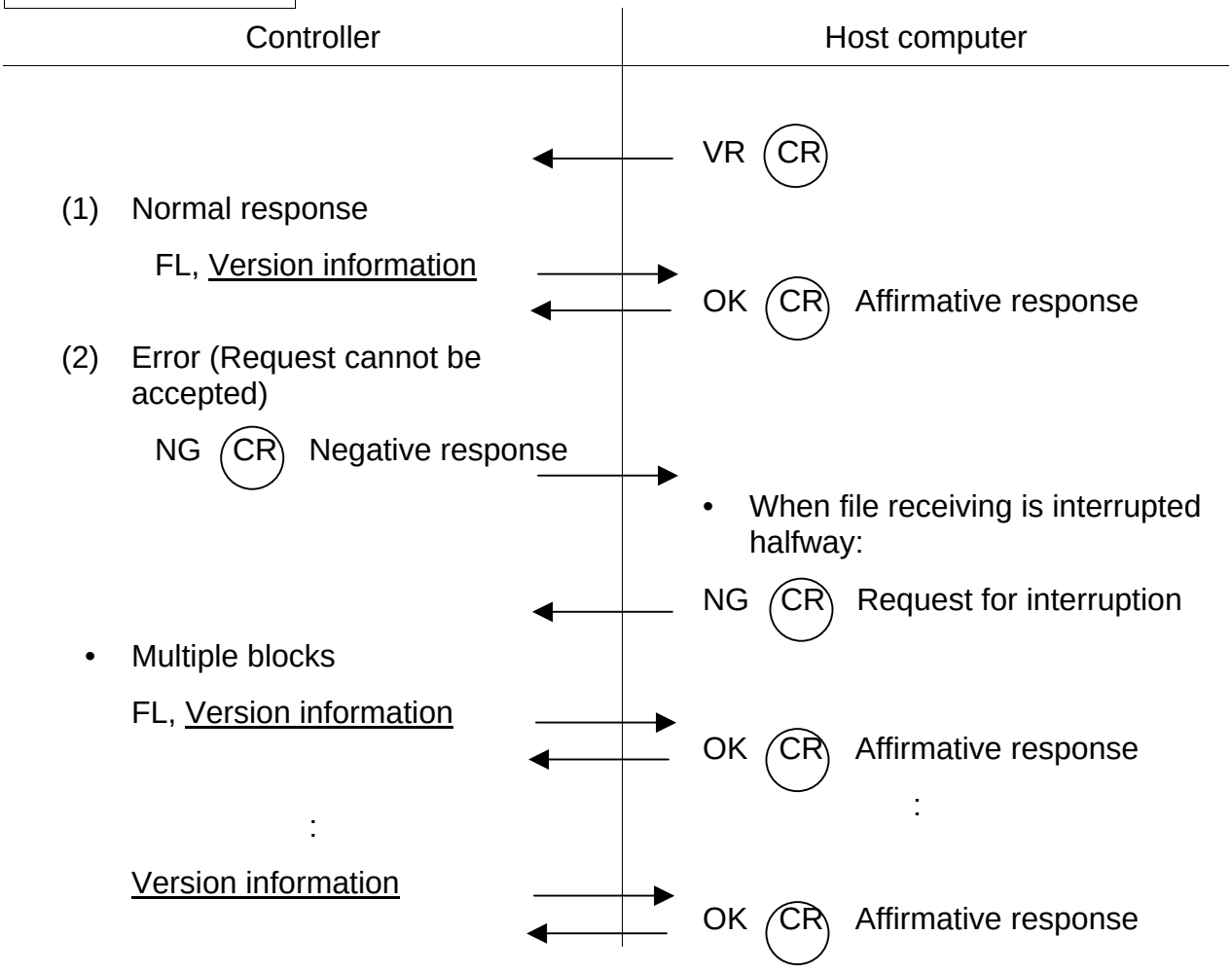
VR (CR)

Function

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

For details on the version information, see the file (FL) command.

Protocol



Note

Put in a delay of about 50 msec to give the host computer the time to send an affirmative response OK signal after having (successfully) received the file contents.

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

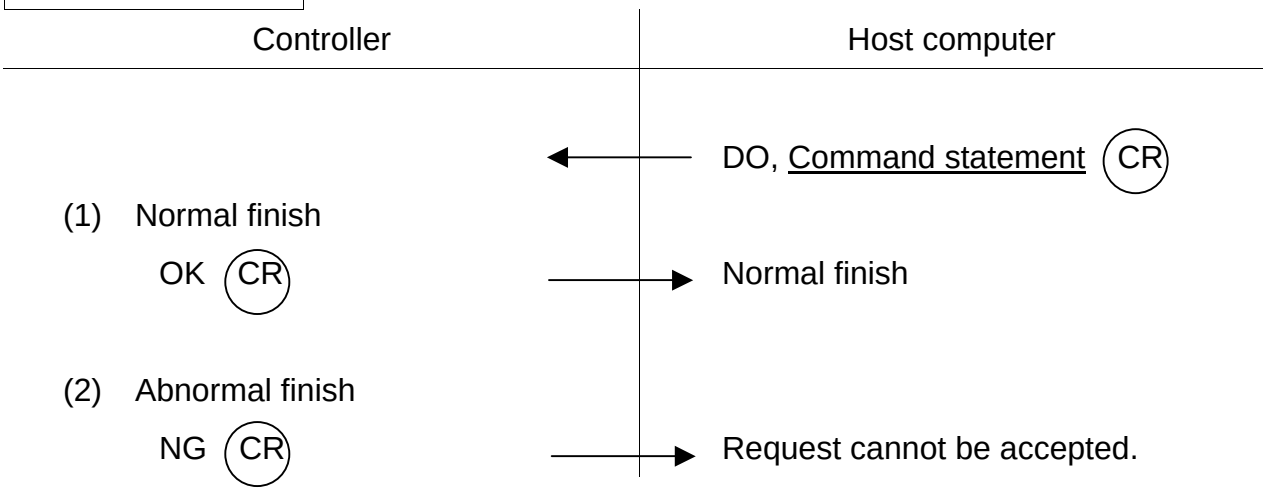
Format

DO, Command statement (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



: Variable read (Host computer → Controller) MR

Format

MR, Variable name Variable type (CR)

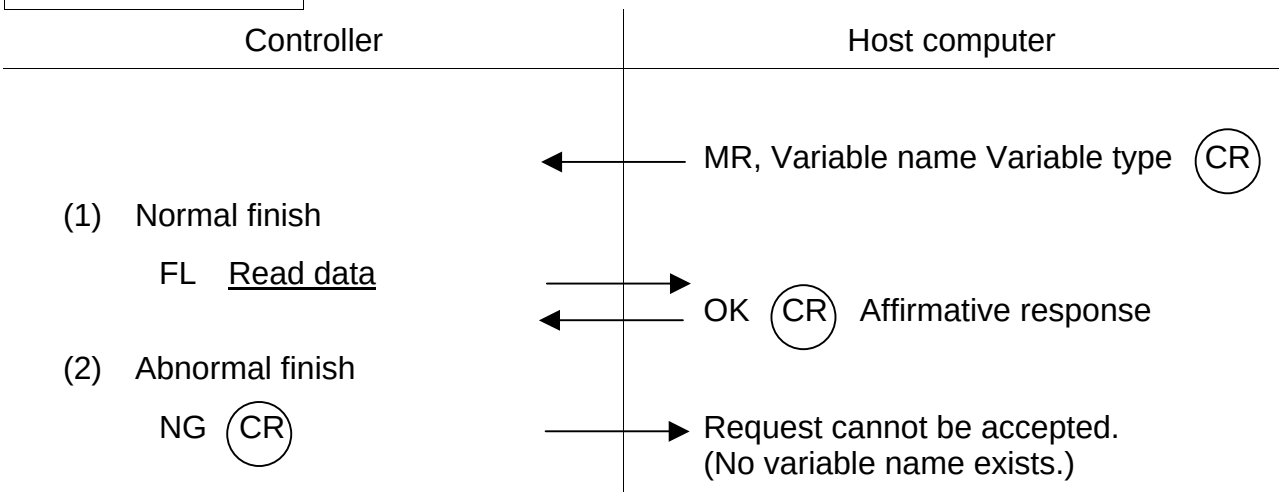
Function

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 file (FL) command.

Protocol



Note

Put in a delay of about 50 msec to give the host computer the time to send an affirmative response OK signal after having (successfully) received the file contents.

: Guide mode setting (Host computer → Controller) MD

Format

MD, Guide mode (CR)

Function

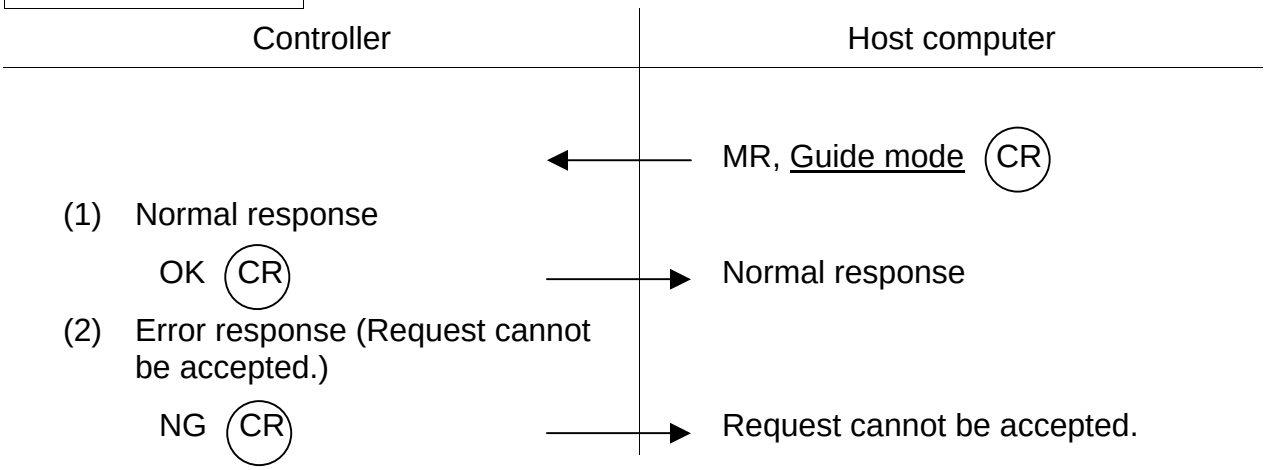
The MD command is used to instruct the guide mode setting from the host computer to the controller.

The guide mode can be specified as shown below.

0 : Jog
1 : Inching
2 : Free

In a status other than the program stop status in the external automatic mode, negative response NG is sent back.

Protocol



: Guide rate setting (Host computer → Controller) RT

Format

RT, Guide rate (CR)

Function

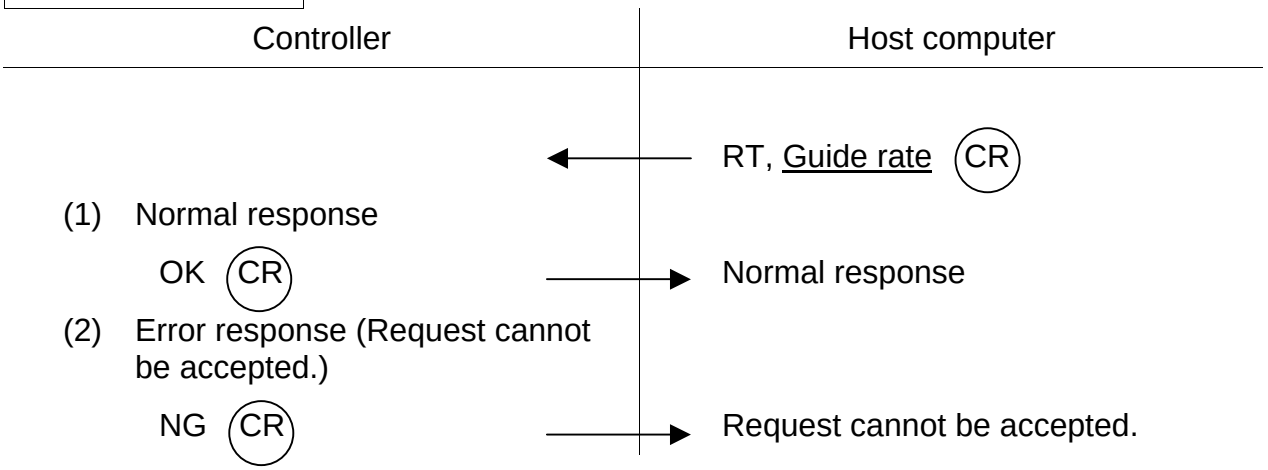
The RT command is used to instruct the guide rate setting from the host computer to the controller.

The guide rate can be specified as shown below.

0 : Slow
1 : Mid
2 : Fast

In a status other than the program stop status in the external automatic mode, negative response NG is sent back.

Protocol



: Guidance coordinate setting (Host computer → Controller) SC

Format

SC, Guidance coordinate (CR)

Function

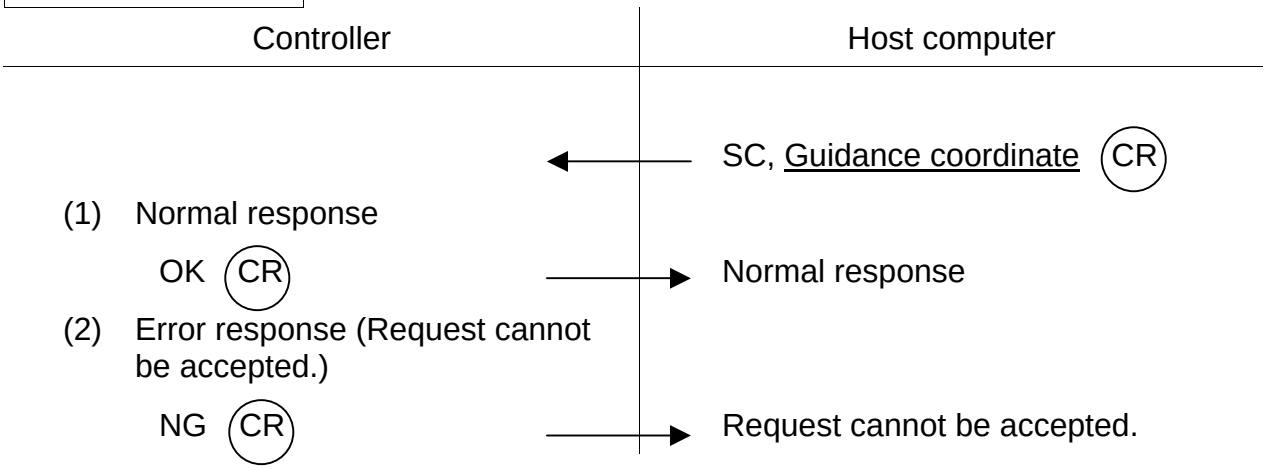
The SC command is used to instruct the guidance coordinate system setting from the host computer to the controller.

The guidance coordinate system can be specified as shown below.

0 : Joint
1 : Tool
2 : Work
3 : World

In a status other than the program stop status in the external automatic mode, negative response NG is sent back.

Protocol



: Write global variable (HOST → CNTL) MW

Format

MW, Flag _ Variable name _ Variable type _ Write data ··· (CR)

Descriptions

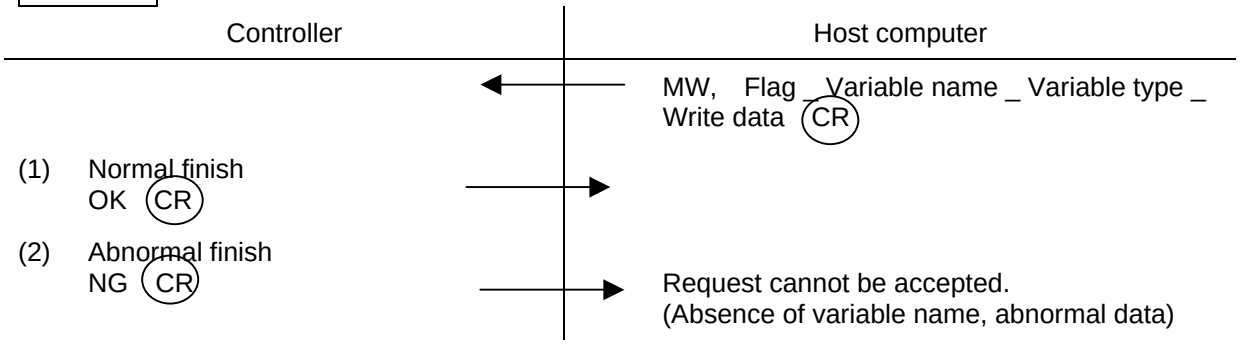
This function requests writing of data into the user's defined variable.
The flag is a control flag for restoring or non-restoring of data in the program file, as shown below.

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)

Procedures



: Feed hold (Host computer → Controller) FD

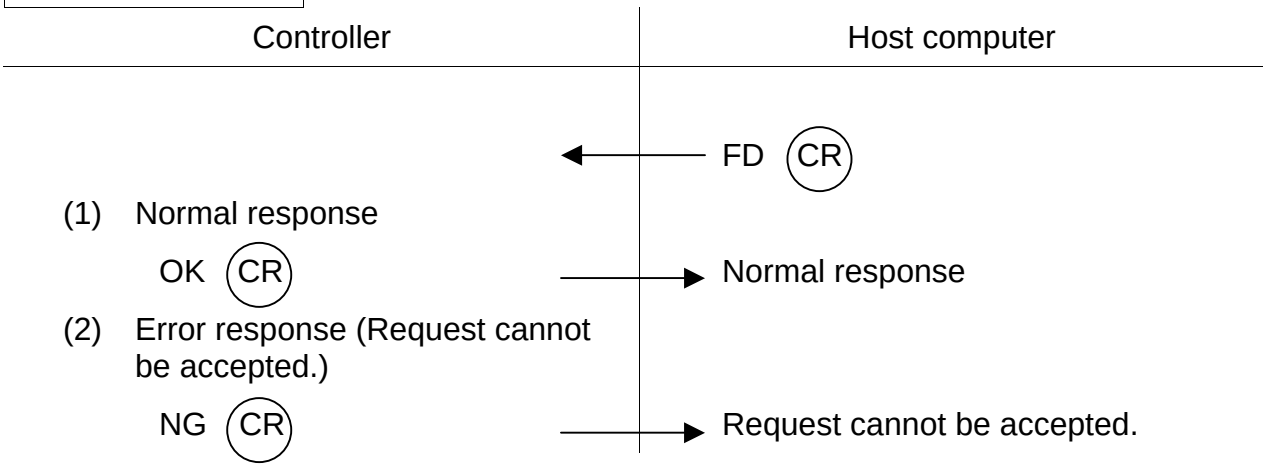
Format

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



: 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 Execution (Host computer → Controller) EC

Format

EC (CR)

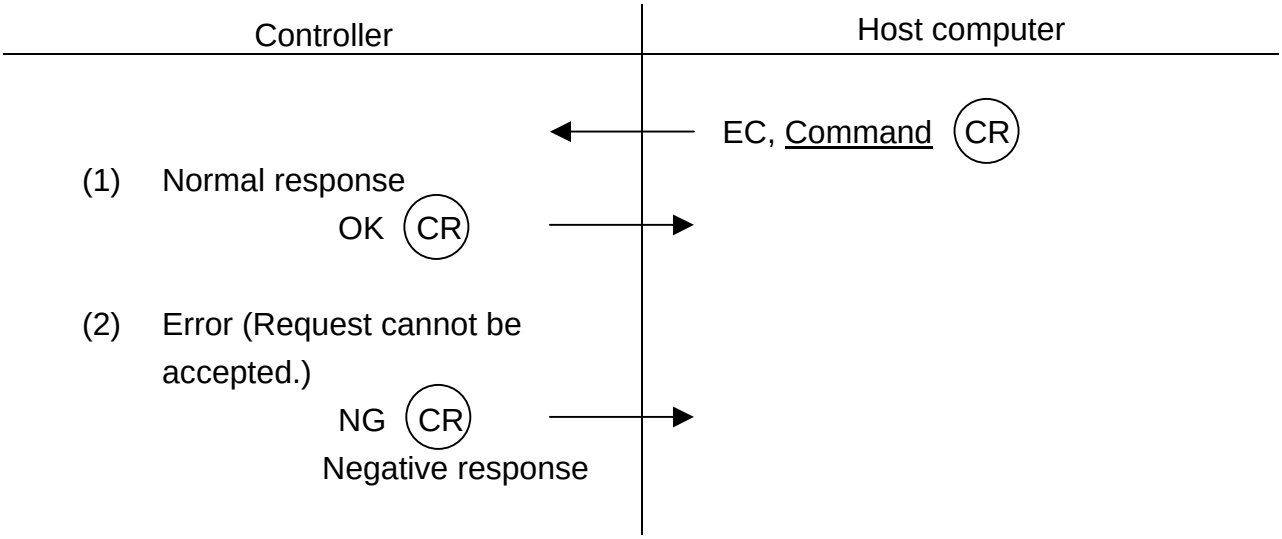
Description

The internal command can be executed from the host computer.
The executable internal commands are as follows.

MODEθCONT
MODEθCYCLE
MODEθSTEP
MODEθSEG
OVRDθSet value (1 ~ 100)
BREAK

Note: "θ" signifies a space.

Protocol



: File (Host computer ↔ Controller) FL

Format

FL, file contents EOF

- When the amount of data to be transmitted exceeds 251 (253) bytes (the maximum amount of data for one text), the data is broken down into additional texts (as shown below) before being transmitted.

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

- EOF (End of File) is a 1 byte code (1AH) used to mark the end of the file.
- The STX (start), ETX (stop) and SUM (check-sum) bytes are attached to each text. (The SUM byte is only appended when check-sum has been specified.)

Descriptions

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

1. File types and contents

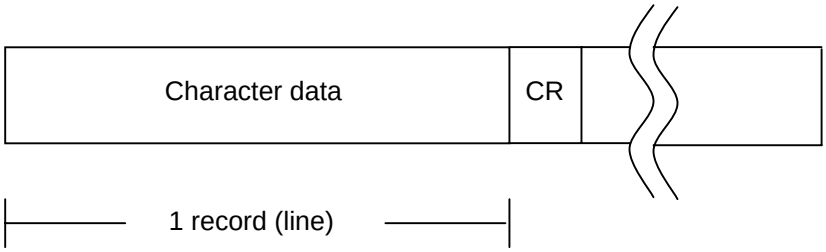
1.1 Types

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

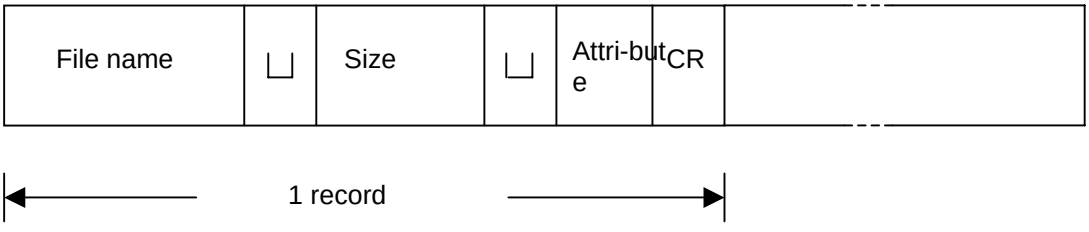


Record structure

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

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

For information on file structure, see Para. 4.4, "RAM Files."



(2) User file directory

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)	0DH (Record termination code)

(3) Status file

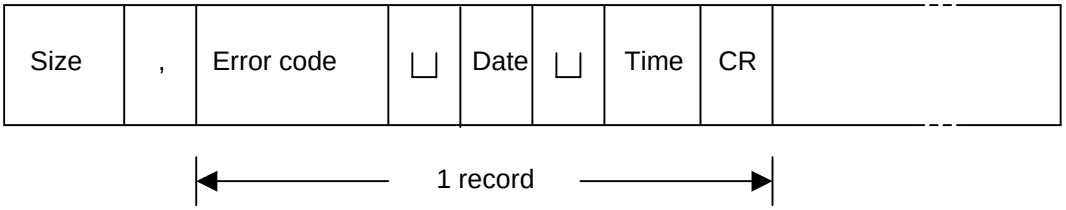
Mode	␣	File name	␣	Speed override	␣	Speed limit	␣

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 mode "external (sig)" : External Automatic Mode (External Operation Signal Mode) "external (host)" : External Automatic Mode (Host 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)	The format of the presently selected file is as follows. "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 (Speed limit)	11 (Variable)	The speed limit is shown in the following format. "LSPEED : (limit) %" (1) LSPEED : 7 byte fixed length. (2) (Limit) : Percentage for the speed limit. (3) % : 1 byte fixed length.
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 rescinded. (Status in which the robot can move.) "lock" : Status in which machine lock is in effect. (Status in which the robot cannot move.)
6	Exe-cution status	21 (Variable)	The execution 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 (Initialized 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.)

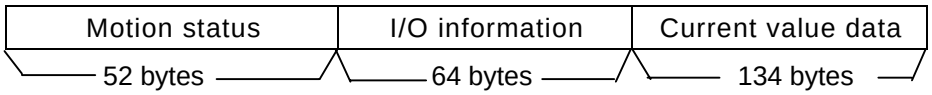
(4) Error history file



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 tow 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)	0DH (Record termination code)

(5) System total status

The following information is transmitted as the system total status information in the binary notation (250 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: 0 ~ 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

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

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>3</td><td>General input 3</td><td>2</td><td>VIData[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></td></tr>
<tr><td>4</td><td>General input 4</td><td>2</td><td>VIData[3]
<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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>5</td><td>Extension input 1</td><td>2</td><td>VIData[4]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din101</td><td>Extension input</td></tr>
<tr><td>1</td><td>Din102</td><td>Extension input</td></tr>
<tr><td>2</td><td>Din103</td><td>Extension input</td></tr>
<tr><td>3</td><td>Din104</td><td>Extension input</td></tr>
<tr><td>4</td><td>Din105</td><td>Extension input</td></tr>
<tr><td>5</td><td>Din106</td><td>Extension input</td></tr>
<tr><td>6</td><td>Din107</td><td>Extension input</td></tr>
<tr><td>7</td><td>Din108</td><td>Extension input</td></tr>
<tr><td>8</td><td>Din110</td><td>Extension input</td></tr>
<tr><td>9</td><td>Din110</td><td>Extension input</td></tr>
<tr><td>10</td><td>Din111</td><td>Extension input</td></tr>
<tr><td>11</td><td>Din112</td><td>Extension input</td></tr>
<tr><td>12</td><td>Din113</td><td>Extension input</td></tr>
<tr><td>13</td><td>Din114</td><td>Extension input</td></tr>
<tr><td>14</td><td>Din115</td><td>Extension input</td></tr>
<tr><td>15</td><td>Din116</td><td>Extension input</td></tr>
</table></td></tr>
<tr><td>6</td><td>Extension input 2</td><td>2</td><td>VIData[5]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din117</td><td>Extension input</td></tr>
<tr><td>1</td><td>Din118</td><td>Extension input</td></tr>
<tr><td>2</td><td>Din119</td><td>Extension input</td></tr>
<tr><td>3</td><td>Din120</td><td>Extension input</td></tr>
<tr><td>4</td><td>Din121</td><td>Extension input</td></tr>
<tr><td>5</td><td>Din122</td><td>Extension input</td></tr>
<tr><td>6</td><td>Din123</td><td>Extension input</td></tr>
<tr><td>7</td><td>Din124</td><td>Extension input</td></tr>
<tr><td>8</td><td>Din125</td><td>Extension input</td></tr>
<tr><td>9</td><td>Din126</td><td>Extension input</td></tr>
<tr><td>10</td><td>Din127</td><td>Extension input</td></tr>
<tr><td>11</td><td>Din128</td><td>Extension input</td></tr>
<tr><td>12</td><td>Din129</td><td>Extension input</td></tr>
<tr><td>13</td><td>Din130</td><td>Extension input</td></tr>
<tr><td>14</td><td>Din131</td><td>Extension input</td></tr>
<tr><td>15</td><td>Din132</td><td>Extension input</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>7</td><td>Extension input 3</td><td>2</td><td>VIData[6]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
<tr><td>8</td><td>Extension input 4</td><td>2</td><td>VIData[7]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>9</td><td>System input 1</td><td>2</td><td>VIData[8]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
<tr><td>10</td><td>System input 2</td><td>2</td><td>VIData[9]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>11</td><td>System input 3</td><td>2</td><td>VIData[10]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din233</td><td>JOGmove</td></tr>
<tr><td>1</td><td>Din234</td><td>JOGinching</td></tr>
<tr><td>2</td><td>Din235</td><td>JOGspeed</td></tr>
<tr><td>3</td><td>Din236</td><td>JOGcood</td></tr>
<tr><td>4</td><td>Din237</td><td>JOF+–</td></tr>
<tr><td>5</td><td>Din238</td><td>JOGaxis1</td></tr>
<tr><td>6</td><td>Din239</td><td>JOGaxis2</td></tr>
<tr><td>7</td><td>Din240</td><td>JOGaxis3</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></td></tr>
<tr><td>12</td><td>System input 4</td><td>2</td><td>VIData[11]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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>I_TEACH</td></tr>
<tr><td>14</td><td>Din263</td><td></td></tr>
<tr><td>15</td><td>Din264</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>13</td><td>Field bus input 1</td><td>2</td><td>VIData[12]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din301</td><td>Field bus input</td></tr>
<tr><td>1</td><td>Din302</td><td>Field bus input</td></tr>
<tr><td>2</td><td>Din303</td><td>Field bus input</td></tr>
<tr><td>3</td><td>Din304</td><td>Field bus input</td></tr>
<tr><td>4</td><td>Din305</td><td>Field bus input</td></tr>
<tr><td>5</td><td>Din306</td><td>Field bus input</td></tr>
<tr><td>6</td><td>Din307</td><td>Field bus input</td></tr>
<tr><td>7</td><td>Din308</td><td>Field bus input</td></tr>
<tr><td>8</td><td>Din310</td><td>Field bus input</td></tr>
<tr><td>9</td><td>Din310</td><td>Field bus input</td></tr>
<tr><td>10</td><td>Din311</td><td>Field bus input</td></tr>
<tr><td>11</td><td>Din312</td><td>Field bus input</td></tr>
<tr><td>12</td><td>Din313</td><td>Field bus input</td></tr>
<tr><td>13</td><td>Din314</td><td>Field bus input</td></tr>
<tr><td>14</td><td>Din315</td><td>Field bus input</td></tr>
<tr><td>15</td><td>Din316</td><td>Field bus input</td></tr>
</table></td></tr>
<tr><td>14</td><td>Field bus input 2</td><td>2</td><td>VIData[13]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din317</td><td>Field bus input</td></tr>
<tr><td>1</td><td>Din318</td><td>Field bus input</td></tr>
<tr><td>2</td><td>Din319</td><td>Field bus input</td></tr>
<tr><td>3</td><td>Din320</td><td>Field bus input</td></tr>
<tr><td>4</td><td>Din321</td><td>Field bus input</td></tr>
<tr><td>5</td><td>Din322</td><td>Field bus input</td></tr>
<tr><td>6</td><td>Din323</td><td>Field bus input</td></tr>
<tr><td>7</td><td>Din324</td><td>Field bus input</td></tr>
<tr><td>8</td><td>Din325</td><td>Field bus input</td></tr>
<tr><td>9</td><td>Din326</td><td>Field bus input</td></tr>
<tr><td>10</td><td>Din327</td><td>Field bus input</td></tr>
<tr><td>11</td><td>Din328</td><td>Field bus input</td></tr>
<tr><td>12</td><td>Din329</td><td>Field bus input</td></tr>
<tr><td>13</td><td>Din330</td><td>Field bus input</td></tr>
<tr><td>14</td><td>Din331</td><td>Field bus input</td></tr>
<tr><td>15</td><td>Din332</td><td>Field bus input</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>15</td><td>Field bus input 3</td><td>2</td><td>VIData[14]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din333</td><td>Field bus input</td></tr>
<tr><td>1</td><td>Din334</td><td>Field bus input</td></tr>
<tr><td>2</td><td>Din335</td><td>Field bus input</td></tr>
<tr><td>3</td><td>Din336</td><td>Field bus input</td></tr>
<tr><td>4</td><td>Din337</td><td>Field bus input</td></tr>
<tr><td>5</td><td>Din338</td><td>Field bus input</td></tr>
<tr><td>6</td><td>Din339</td><td>Field bus input</td></tr>
<tr><td>7</td><td>Din340</td><td>Field bus input</td></tr>
<tr><td>8</td><td>Din341</td><td>Field bus input</td></tr>
<tr><td>9</td><td>Din342</td><td>Field bus input</td></tr>
<tr><td>10</td><td>Din343</td><td>Field bus input</td></tr>
<tr><td>11</td><td>Din344</td><td>Field bus input</td></tr>
<tr><td>12</td><td>Din345</td><td>Field bus input</td></tr>
<tr><td>13</td><td>Din346</td><td>Field bus input</td></tr>
<tr><td>14</td><td>Din347</td><td>Field bus input</td></tr>
<tr><td>15</td><td>Din348</td><td>Field bus input</td></tr>
</table></td></tr>
<tr><td>16</td><td>Field bus input 4</td><td>2</td><td>VIData[15]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Din349</td><td>Field bus input</td></tr>
<tr><td>1</td><td>Din350</td><td>Field bus input</td></tr>
<tr><td>2</td><td>Din351</td><td>Field bus input</td></tr>
<tr><td>3</td><td>Din352</td><td>Field bus input</td></tr>
<tr><td>4</td><td>Din353</td><td>Field bus input</td></tr>
<tr><td>5</td><td>Din354</td><td>Field bus input</td></tr>
<tr><td>6</td><td>Din355</td><td>Field bus input</td></tr>
<tr><td>7</td><td>Din356</td><td>Field bus input</td></tr>
<tr><td>8</td><td>Din357</td><td>Field bus input</td></tr>
<tr><td>9</td><td>Din358</td><td>Field bus input</td></tr>
<tr><td>10</td><td>Din359</td><td>Field bus input</td></tr>
<tr><td>11</td><td>Din360</td><td>Field bus input</td></tr>
<tr><td>12</td><td>Din361</td><td>Field bus input</td></tr>
<tr><td>13</td><td>Din362</td><td>Field bus input</td></tr>
<tr><td>14</td><td>Din363</td><td>Field bus input</td></tr>
<tr><td>15</td><td>Din364</td><td>Field bus input</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>17</td><td>General output 1</td><td>2</td><td>VOData[0]
<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></td></tr>
<tr><td>18</td><td>General output 2</td><td>2</td><td>VOData[1]
<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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>19</td><td>General output 3</td><td>2</td><td>VOData[2]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout33</td><td>General output</td></tr>
<tr><td>1</td><td>Dout34</td><td>General output</td></tr>
<tr><td>2</td><td>Dout35</td><td>General output</td></tr>
<tr><td>3</td><td>Dout36</td><td>General output</td></tr>
<tr><td>4</td><td>Dout37</td><td>General output</td></tr>
<tr><td>5</td><td>Dout38</td><td>General output</td></tr>
<tr><td>6</td><td>Dout39</td><td>General output</td></tr>
<tr><td>7</td><td>Dout40</td><td>General output</td></tr>
<tr><td>8</td><td>Dout41</td><td>General output</td></tr>
<tr><td>9</td><td>Dout42</td><td>General output</td></tr>
<tr><td>10</td><td>Dout43</td><td>General output</td></tr>
<tr><td>11</td><td>Dout44</td><td>General output</td></tr>
<tr><td>12</td><td>Dout45</td><td>General output</td></tr>
<tr><td>13</td><td>Dout46</td><td>General output</td></tr>
<tr><td>14</td><td>Dout47</td><td>General output</td></tr>
<tr><td>15</td><td>Dout48</td><td>General output</td></tr>
</table></td></tr>
<tr><td>20</td><td>General input 4</td><td>2</td><td>VOData[3]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout49</td><td>General output</td></tr>
<tr><td>1</td><td>Dout50</td><td>General output</td></tr>
<tr><td>2</td><td>Dout51</td><td>General output</td></tr>
<tr><td>3</td><td>Dout52</td><td>General output</td></tr>
<tr><td>4</td><td>Dout53</td><td>General output</td></tr>
<tr><td>5</td><td>Dout54</td><td>General output</td></tr>
<tr><td>6</td><td>Dout55</td><td>General output</td></tr>
<tr><td>7</td><td>Dout56</td><td>General output</td></tr>
<tr><td>8</td><td>Dout57</td><td>General output</td></tr>
<tr><td>9</td><td>Dout58</td><td>General output</td></tr>
<tr><td>10</td><td>Dout59</td><td>General output</td></tr>
<tr><td>11</td><td>Dout60</td><td>General output</td></tr>
<tr><td>12</td><td>Dout61</td><td>General output</td></tr>
<tr><td>13</td><td>Dout62</td><td>General output</td></tr>
<tr><td>14</td><td>Dout63</td><td>General output</td></tr>
<tr><td>15</td><td>Dout64</td><td>General output</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>21</td><td>Extension output 1</td><td>2</td><td>VOData[4]
<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></td></tr>
<tr><td>22</td><td>Extension output 2</td><td>2</td><td>VOData[5]
<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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>23</td><td>Extension output 3</td><td>2</td><td>VOData[6]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
<tr><td>24</td><td>Extension output 4</td><td>2</td><td>VOData[7]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>25</td><td>System output 1</td><td>2</td><td>VOData[8]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></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></td></tr>
<tr><td>26</td><td>System output 2</td><td>2</td><td>VOData[9]
<table>
<tr><th>Bit</th><th>Line No.</th><th colspan="2">Signal name</th></tr>
<tr><td>0</td><td>Dout217</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>1</td><td>Dout218</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>2</td><td>Dout219</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>3</td><td>Dout220</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>4</td><td>Dout221</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>5</td><td>Dout222</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>6</td><td>Dout223</td><td colspan="2">Seq. parameter</td></tr>
<tr><td>7</td><td>Dout224</td><td colspan="2">Seq. parameter</td></tr>
<tr><td></td><td></td><td>TS1000</td><td>TS2000</td></tr>
<tr><td>8</td><td>Dout225</td><td></td><td></td></tr>
<tr><td>9</td><td>Dout226</td><td></td><td></td></tr>
<tr><td>10</td><td>Dout227</td><td></td><td></td></tr>
<tr><td>11</td><td>Dout228</td><td></td><td></td></tr>
<tr><td>12</td><td>Dout229</td><td></td><td>O13_SEL</td></tr>
<tr><td>13</td><td>Dout230</td><td></td><td>O14_SEL</td></tr>
<tr><td>14</td><td>Dout231</td><td></td><td>O15_SEL</td></tr>
<tr><td>15</td><td>Dout232</td><td></td><td>O16_SEL</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>27</td><td>System output 3</td><td>2</td><td>VOData[10]
<table>
<tr><th>Bit</th><th>Line No.</th><th colspan="2">Signal name</th></tr>
<tr><th></th><th></th><th>TS1000</th><th>TS2000</th></tr>
<tr><td>0</td><td>Dout233</td><td>I9_SEL</td><td>I23_SEL</td></tr>
<tr><td>1</td><td>Dout234</td><td>I10_SEL</td><td>I24_SEL</td></tr>
<tr><td>2</td><td>Dout235</td><td>I11_SEL</td><td>I33_SEL</td></tr>
<tr><td>3</td><td>Dout236</td><td>I12_SEL</td><td>I34_EL</td></tr>
<tr><td>4</td><td>Dout237</td><td>I13_SEL</td><td>I35_SEL</td></tr>
<tr><td>5</td><td>Dout238</td><td>I14_SEL</td><td>I36_SEL</td></tr>
<tr><td>6</td><td>Dout239</td><td>I15_SEL</td><td>I37_SEL</td></tr>
<tr><td>7</td><td>Dout240</td><td>I16_SEL</td><td>I38_SEL</td></tr>
<tr><td>8</td><td>Dout241</td><td>O9_SEL</td><td>O25_SEL</td></tr>
<tr><td>9</td><td>Dout242</td><td>O10_SEL</td><td>O26_SEL</td></tr>
<tr><td>10</td><td>Dout243</td><td>O11_SEL</td><td>O27_SEL</td></tr>
<tr><td>11</td><td>Dout244</td><td>O12_SEL</td><td>O28_SEL</td></tr>
<tr><td>12</td><td>Dout245</td><td>O13_SEL</td><td>O29_SEL</td></tr>
<tr><td>13</td><td>Dout246</td><td>O14_SEL</td><td>O30_SEL</td></tr>
<tr><td>14</td><td>Dout247</td><td>O15_SEL</td><td>O31_SEL</td></tr>
<tr><td>15</td><td>Dout248</td><td>O16_SEL</td><td>O32_SEL</td></tr>
</table></td></tr>
<tr><td>28</td><td>System output 4</td><td>2</td><td>VOData[11]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout249</td><td>EMG_ST</td></tr>
<tr><td>1</td><td>Dout250</td><td>SV_RDY</td></tr>
<tr><td>2</td><td>Dout251</td><td>ACK</td></tr>
<tr><td>3</td><td>Dout252</td><td>TEACH</td></tr>
<tr><td>4</td><td>Dout253</td><td>INT</td></tr>
<tr><td>5</td><td>Dout254</td><td>EXTSIG</td></tr>
<tr><td>6</td><td>Dout255</td><td>EXTHOST</td></tr>
<tr><td>7</td><td>Dout256</td><td>SYS_RDY</td></tr>
<tr><td>8</td><td>Dout257</td><td>AUTORUN</td></tr>
<tr><td>9</td><td>Dout258</td><td>CYC_END</td></tr>
<tr><td>10</td><td>Dout259</td><td>LOW_ST</td></tr>
<tr><td>11</td><td>Dout260</td><td>CYC_ST</td></tr>
<tr><td>12</td><td>Dout261</td><td>BT_ALM</td></tr>
<tr><td>13</td><td>Dout262</td><td>ALARM</td></tr>
<tr><td>14</td><td>Dout263</td><td></td></tr>
<tr><td>15</td><td>Dout264</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>29</td><td>Field bus output 1</td><td>2</td><td>VOData[12]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout301</td><td>Field bus output</td></tr>
<tr><td>1</td><td>Dout302</td><td>Field bus output</td></tr>
<tr><td>2</td><td>Dout303</td><td>Field bus output</td></tr>
<tr><td>3</td><td>Dout304</td><td>Field bus output</td></tr>
<tr><td>4</td><td>Dout305</td><td>Field bus output</td></tr>
<tr><td>5</td><td>Dout306</td><td>Field bus output</td></tr>
<tr><td>6</td><td>Dout307</td><td>Field bus output</td></tr>
<tr><td>7</td><td>Dout308</td><td>Field bus output</td></tr>
<tr><td>8</td><td>Dout310</td><td>Field bus output</td></tr>
<tr><td>9</td><td>Dout310</td><td>Field bus output</td></tr>
<tr><td>10</td><td>Dout311</td><td>Field bus output</td></tr>
<tr><td>11</td><td>Dout312</td><td>Field bus output</td></tr>
<tr><td>12</td><td>Dout313</td><td>Field bus output</td></tr>
<tr><td>13</td><td>Dout314</td><td>Field bus output</td></tr>
<tr><td>14</td><td>Dout315</td><td>Field bus output</td></tr>
<tr><td>15</td><td>Dout316</td><td>Field bus output</td></tr>
</table></td></tr>
<tr><td>30</td><td>Field bus output 2</td><td>2</td><td>VOData[13]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout317</td><td>Field bus output</td></tr>
<tr><td>1</td><td>Dout318</td><td>Field bus output</td></tr>
<tr><td>2</td><td>Dout319</td><td>Field bus output</td></tr>
<tr><td>3</td><td>Dout320</td><td>Field bus output</td></tr>
<tr><td>4</td><td>Dout321</td><td>Field bus output</td></tr>
<tr><td>5</td><td>Dout322</td><td>Field bus output</td></tr>
<tr><td>6</td><td>Dout323</td><td>Field bus output</td></tr>
<tr><td>7</td><td>Dout324</td><td>Field bus output</td></tr>
<tr><td>8</td><td>Dout325</td><td>Field bus output</td></tr>
<tr><td>9</td><td>Dout326</td><td>Field bus output</td></tr>
<tr><td>10</td><td>Dout327</td><td>Field bus output</td></tr>
<tr><td>11</td><td>Dout328</td><td>Field bus output</td></tr>
<tr><td>12</td><td>Dout329</td><td>Field bus output</td></tr>
<tr><td>13</td><td>Dout330</td><td>Field bus output</td></tr>
<tr><td>14</td><td>Dout331</td><td>Field bus output</td></tr>
<tr><td>15</td><td>Dout332</td><td>Field bus output</td></tr>
</table></td></tr>
</table>

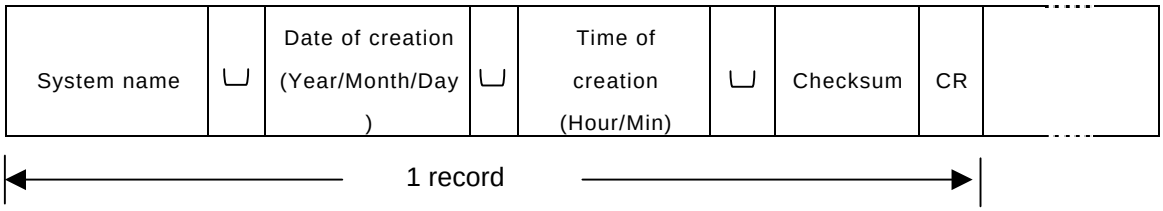
<table>
<tr><th>No.</th><th>Name</th><th>Size (byte)</th><th>Description</th></tr>
<tr><td>31</td><td>Field bus output 3</td><td>2</td><td>VOData[14]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout333</td><td>Field bus output</td></tr>
<tr><td>1</td><td>Dout334</td><td>Field bus output</td></tr>
<tr><td>2</td><td>Dout335</td><td>Field bus output</td></tr>
<tr><td>3</td><td>Dout336</td><td>Field bus output</td></tr>
<tr><td>4</td><td>Dout337</td><td>Field bus output</td></tr>
<tr><td>5</td><td>Dout338</td><td>Field bus output</td></tr>
<tr><td>6</td><td>Dout339</td><td>Field bus output</td></tr>
<tr><td>7</td><td>Dout340</td><td>Field bus output</td></tr>
<tr><td>8</td><td>Dout341</td><td>Field bus output</td></tr>
<tr><td>9</td><td>Dout342</td><td>Field bus output</td></tr>
<tr><td>10</td><td>Dout343</td><td>Field bus output</td></tr>
<tr><td>11</td><td>Dout344</td><td>Field bus output</td></tr>
<tr><td>12</td><td>Dout345</td><td>Field bus output</td></tr>
<tr><td>13</td><td>Dout346</td><td>Field bus output</td></tr>
<tr><td>14</td><td>Dout347</td><td>Field bus output</td></tr>
<tr><td>15</td><td>Dout348</td><td>Field bus output</td></tr>
</table></td></tr>
<tr><td>32</td><td>Field bus output 4</td><td>2</td><td>VOData[15]
<table>
<tr><th>Bit</th><th>Line No.</th><th>Signal name</th></tr>
<tr><td>0</td><td>Dout349</td><td>Field bus output</td></tr>
<tr><td>1</td><td>Dout350</td><td>Field bus output</td></tr>
<tr><td>2</td><td>Dout351</td><td>Field bus output</td></tr>
<tr><td>3</td><td>Dout352</td><td>Field bus output</td></tr>
<tr><td>4</td><td>Dout353</td><td>Field bus output</td></tr>
<tr><td>5</td><td>Dout354</td><td>Field bus output</td></tr>
<tr><td>6</td><td>Dout355</td><td>Field bus output</td></tr>
<tr><td>7</td><td>Dout356</td><td>Field bus output</td></tr>
<tr><td>8</td><td>Dout357</td><td>Field bus output</td></tr>
<tr><td>9</td><td>Dout358</td><td>Field bus output</td></tr>
<tr><td>10</td><td>Dout359</td><td>Field bus output</td></tr>
<tr><td>11</td><td>Dout360</td><td>Field bus output</td></tr>
<tr><td>12</td><td>Dout361</td><td>Field bus output</td></tr>
<tr><td>13</td><td>Dout362</td><td>Field bus output</td></tr>
<tr><td>14</td><td>Dout363</td><td>Field bus output</td></tr>
<tr><td>15</td><td>Dout364</td><td>Field bus output</td></tr>
</table></td></tr>
</table>

III) Current value data

No.	Name	Size (byte)	Description
1	Joint coordinate value	4 × 6 axes	As shown below, values are set in the order of axis 1 to axis 6.
2	World coordinate value	4 × 6 axes	
3	Work coordinate value	4 × 6 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		134	

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

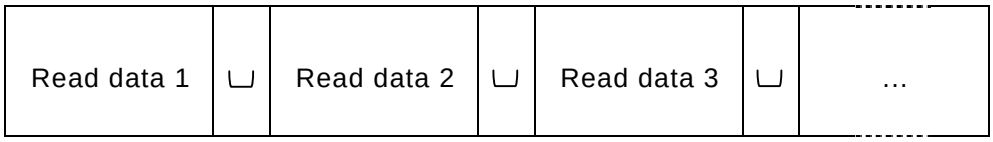
(6) Version information



[Contents of record]

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 (0DH)

(7) Variable read data



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

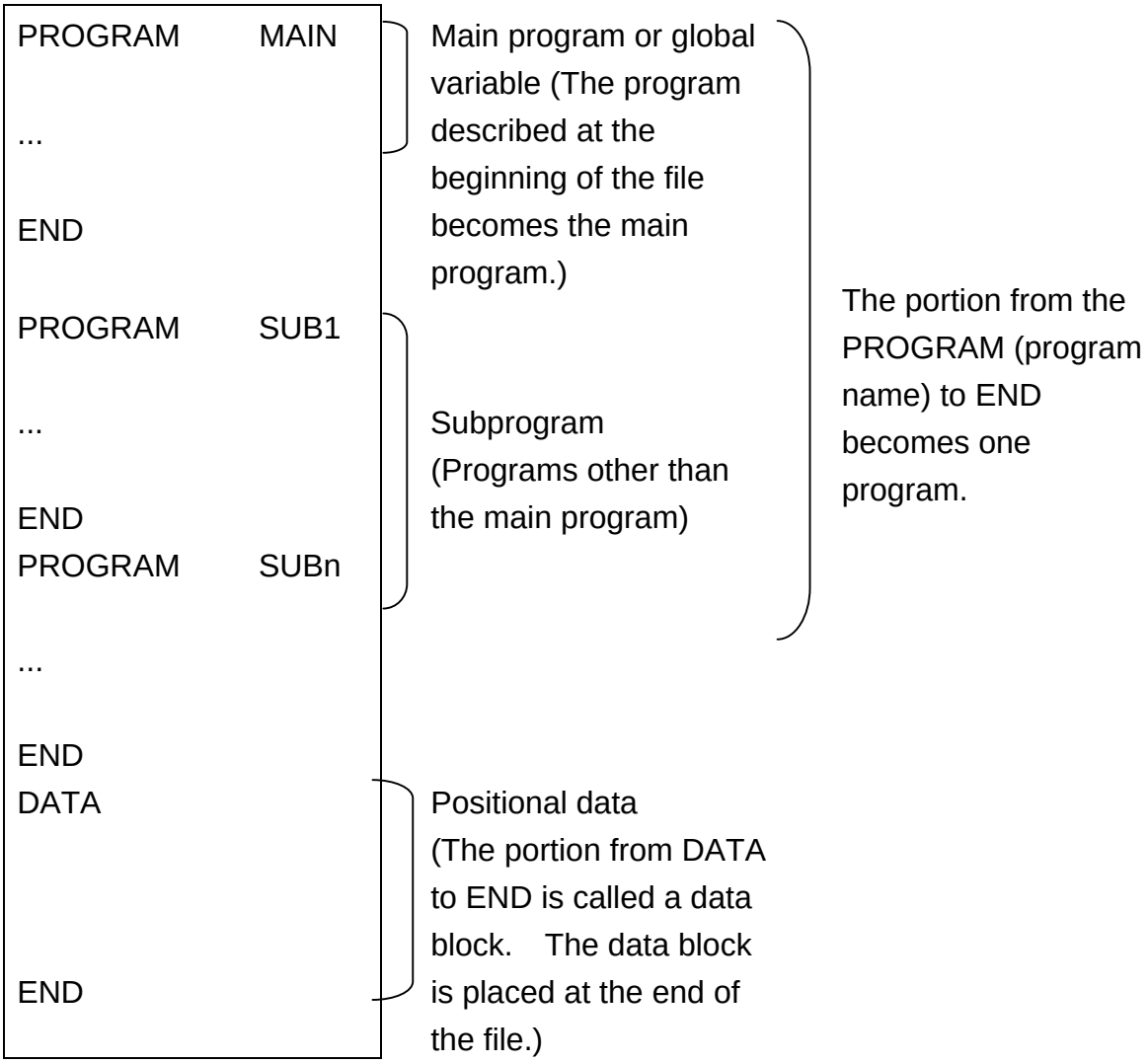
Variable type data	Type	Read data
0	Integer type	1 pc. (long)
1	Real number type	1 pc. (float)
2	Load type	2 pcs. (float␣float)
3	Coordinate type	4 pcs. (float␣float␣float␣float)
4	Position type	6 pcs. (float␣float␣float␣float␣float␣float)
5	Array integer type	1 pc. (long)
6	Array real number type	1 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.

4.4 Ram Files

4.5.1 User Files (Work files)

The user files store some programs and positional data.



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 program, two or more programs can be described.
For details of the robot language, see the manual, "Robot Language".

(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>/<config>

<Position name>	:	Name of positional data
<X>	:	Coordinate values of X, Y, and Z (unit: mm)
<Y>	:	
<Z>	:	
<C>	:	Rotating angle of tool end (unit: deg)
<T>	:	Data of axis 5
<Physical orientation>	:	Configuration of arm of horizontal articulated type robot
	Omitted	... Not defined
	LEFTY	... Left handed system
	RIGHTY	... Right handed system

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>	:	Coordinate values in directions of axes X, Y, and Z (unit: mm)
<Y>	:	
<Z>	:	
<C>	:	Rotating angle around axis Z (unit: deg)

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

[3] Load data

PLAYLOAD
<Load name>=<mass>, <center-of-gravity offset>

<Load name>	:	Name of load data
<Mass>	:	Weight of load applied at the end of the robot hand (unit: kg)
<Center-of-gravity offset>	:	Offset of the center of gravity of load applied to 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 treated 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 (0AH) is attached to the end of each line (or block).

File name: TEST

```
GLOBAL
A = 10
END
PROGRAM SAMPLE
SPEED = 20
MOVE A1
MOVE A2
MOVE A3
SUB1
MOVE A3
MOVE A2
MOVE A4
MOVE A5
SUB2
MOVE A5
MOVE A4
MOVE A1
END
PROGRAM SUB1
PAYLOAD = HAND 1
MOVE A11
MOVE A12
MOVE A13
FREELOAD
END
PROGRAM SUB2
PAYLOAD = HAND 2
MOVE A21
MOVE A22
MOVE A23
FREELOAD
END
```

Global variable block

Main program

Subprogram [SUB1]

Subprogram [SUB2]

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 world coordinate system

Coordinate data W1

Coordinate data W2

Load data HAND1

Load data HAND2

Position of work coordinate system = W1

Position of work coordinate system = W2

Data block

Section 5

Robot Operation Sequences

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

5.1 Basic Operation

Figure 5.1 presents a flow sequence in which a certain block is selected and executed repeatedly.

5.2 Program Download

Figure 5.2 presents a flow sequence in which a program is downloaded at the completion of each cycle.

5.3 Reinitializing and Starting a Stopped Program

Figure 5.3 presents a flow sequence which reinitializes and starts a program which has been stopped.

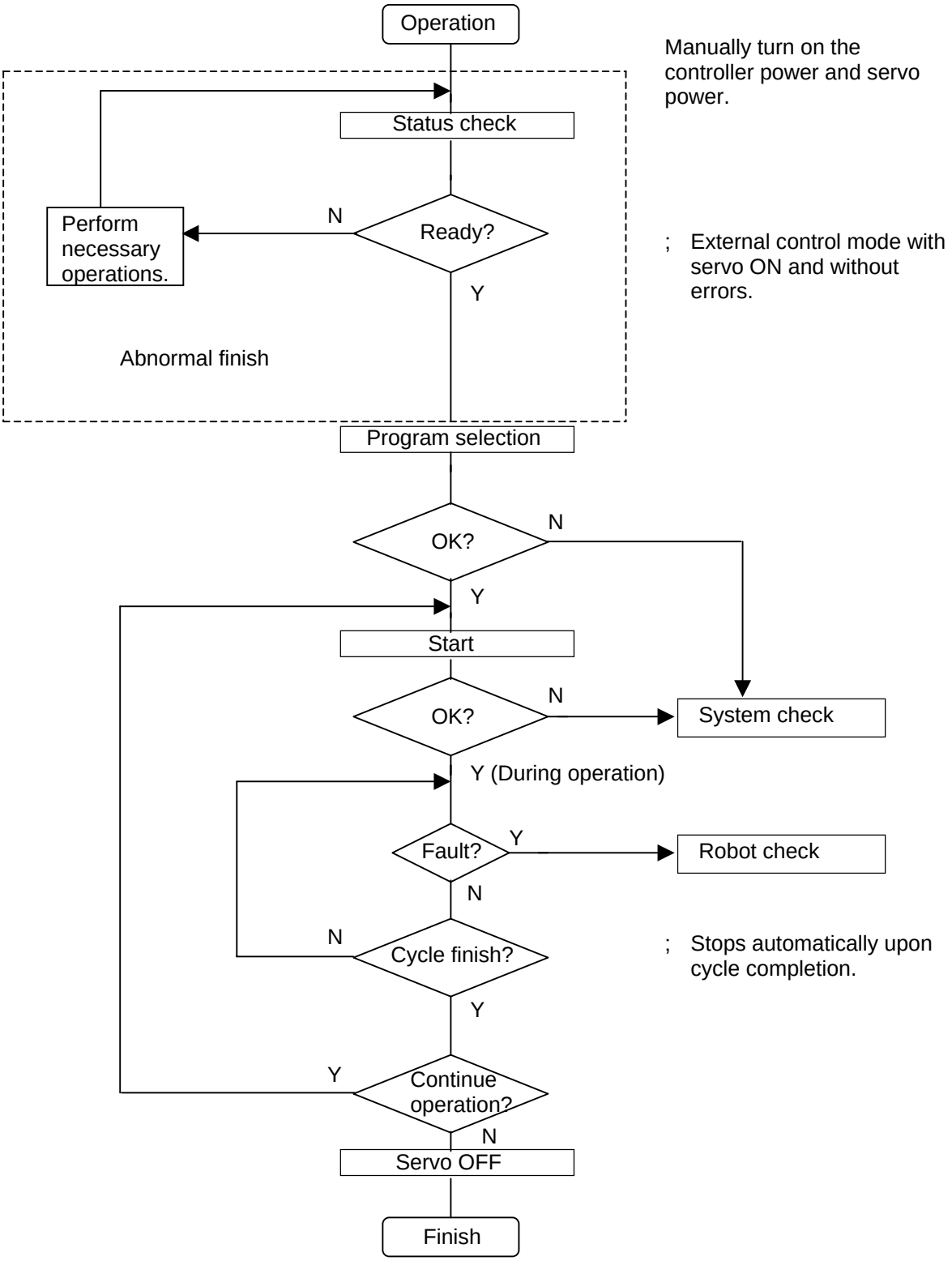


Fig. 5.1 Basic operation sequence

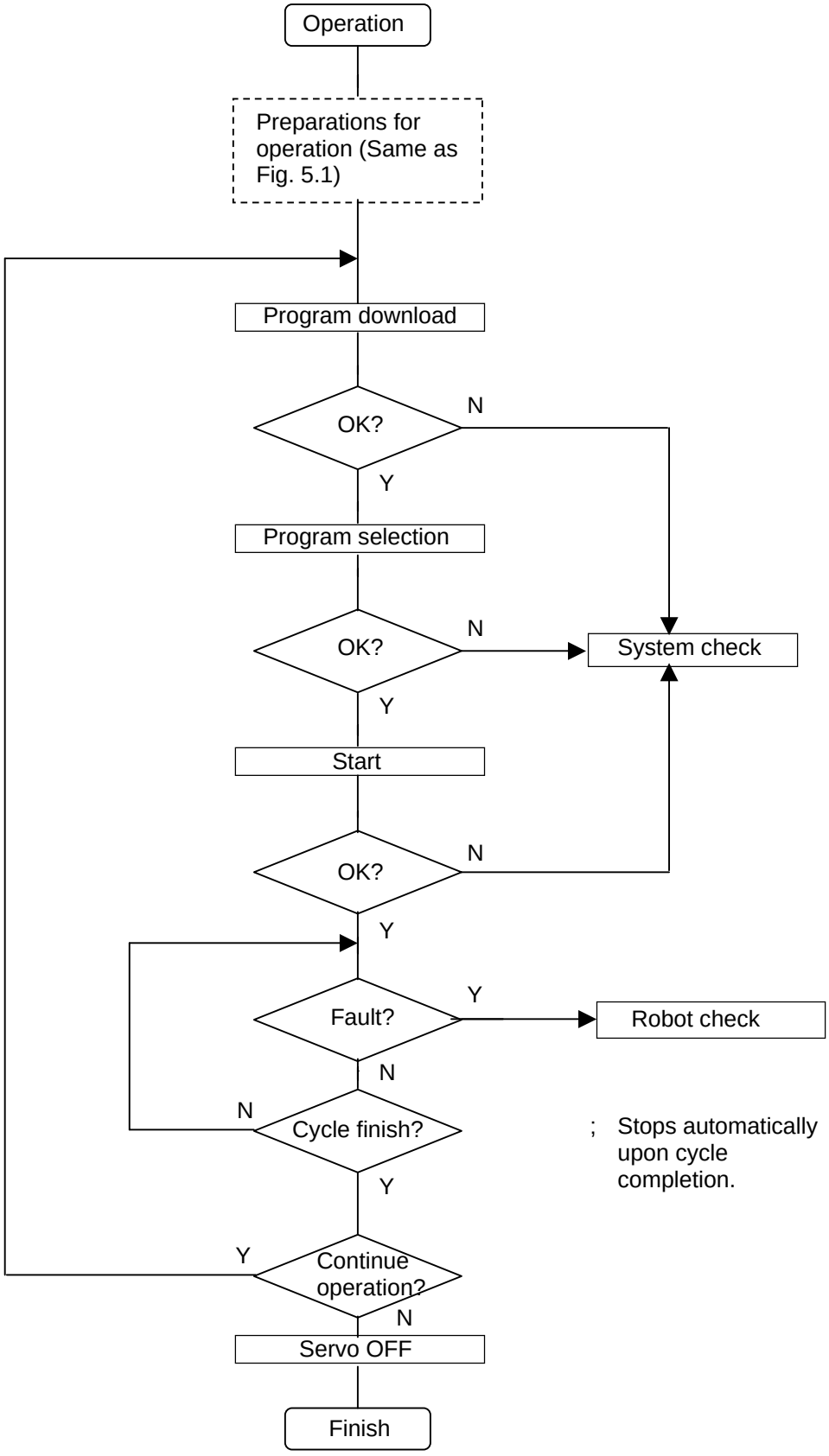


Fig. 5.2 Operation sequence including program download

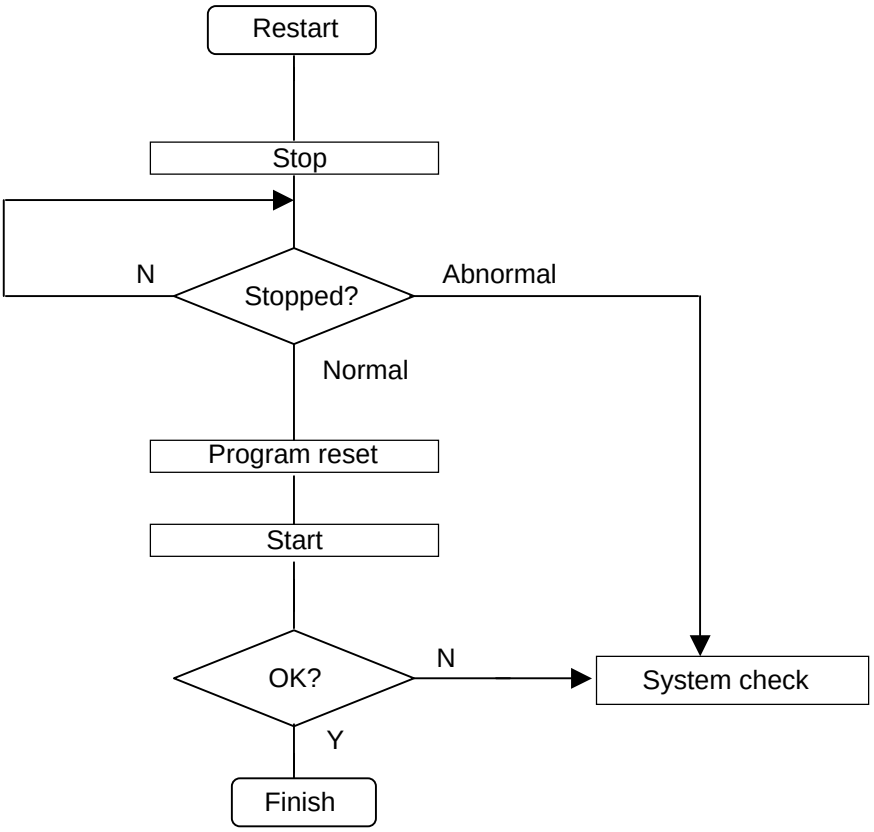


Fig. 5.3 Restart after program interruption

Section 6 Appendix

6.1 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

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

APPROVED BY: Y. Yamaguchi

CHECKED BY: K. Kato

PREPARED BY: K. Katsumata