```
'*****************************************************************************
                                                                             *
                         "QN" Command Example                                *
                                 for                                         *
'*                   Mitsubishi "E" Robot Controllers                        *
'*                                                                           *
'*    The "N" command will allow you to select the program name that you    *
'*    have in your controller.  The "QN" command with the progam name        *
'*    after it will inquire the controller for the Total Number of Used      *
'*    Steps, Used Number of Positions and Used Number of Counters.           *
'*                                                                           *
'*    N9 = selects program number 9,  QN9 = inquires for information        *
'*    N"PUT_PAM" = selects program PUT_PAM                                   *
'*    QN"PUT_PAM" = inquires for information                                 *
'*                                                                           *
'*     Date : 4/13/95                                                        *
'*****************************************************************************

  CLOSE
  OPEN "COM2:9600,N,8,1,CS60000,DS60000,CD0" FOR RANDOM AS #2
'*****************************************************************************

  CLS
  PRINT #2, "N9"
  PRINT #2, " QN9"         'Must have a space before the QN Command to work
  LINE INPUT #2, A$
  LOCATE 15, 10: PRINT "1ST Selection using (0-9) Numbers was --  "; A$

   RINT #2, "N" + CHR$(34) + "PUT_PAM" + CHR$(34)       'CHR$(34)= the "
  PRINT #2, " QN" + CHR$(34) + "PUT_PAM" + CHR$(34)
  LINE INPUT #2, A$
  LOCATE 17, 10: PRINT "2ND Selection using (Alpha Characters) was -- "; A$

  END

'*****************************************************************************
```

CAUTION!

(TEACHING BOX CONNECTION/DISCONNECTION)

The following shows the corrigenda about the procedure of how to connect and disconnect the teaching box (T.B.) during power on. We kindly ask the user to refer to this information when reading the corresponding parts of the user's manual and the reference manual.

WHEN DISCONNECTING: *Incorrect operation will cause an emergency stop state.*

(1) Set the 'ENBL/DISABLE' switch of the T.B. to the 'DISABLE' side.
(2) Loosen the fixing screws of the T.B. connector with holding it firmly by hand.
(3) Push and lock down the EMG. CANCEL switch on the front panel. (Concave state) Confirm that the LED of the switch starts blinking
(4) Disconnect the connector quickly within five seconds from the start of blinking. The LED turns off when completed correctly.

WHEN CONNECTING: *Incorrect operation will cause an emergency stop state.*

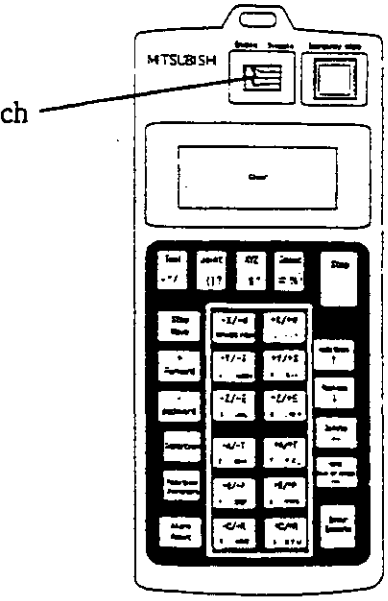
(1) Set the 'ENBL/DISABLE' switch of the T.B. to the 'DISABLE' side.
(2) Connect the T.B. connector. Confirm that the LED of the switch starts blinking.
(3) Push and release up the EMG. CANCEL switch. (Convex state) This must be done within five seconds from the connection of the T.B.. The LED turns on when connected correctly.
(4) Fix the connector of the T.B. with a set of screws firmly.

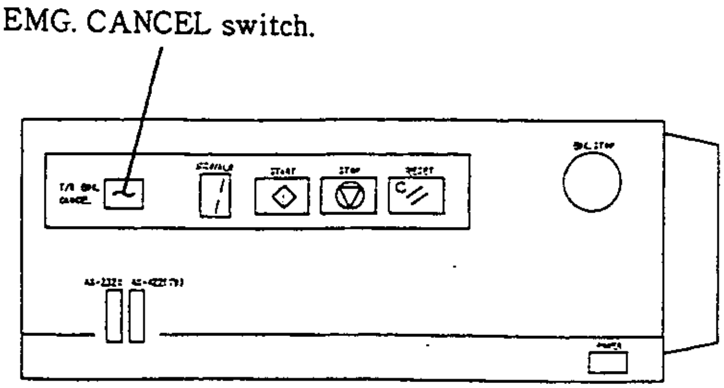
TO RESET THE EMERGENCY STOP STATE:

(1) Release up the EMG. CANCEL switch. (Convex state)
(2) Set the 'ENBL/DISABLE' switch of the T.B. to the 'ENBL' side.
(3) Push the ALARM RESET key of the T.B..

CAUTION!:

THE EMERGENCY SWITCH OF THE T.B. REMAINS DISABLED WHEN THE 'EMG. CANCEL' SWITCH IS PUSHED DOWN. THE INITIATION OF THE PROGRAM FROM OTHER EQUIPMENT THAN T.B. IS EFFECTIVE WHEN THE 'ENBL/DISABLE' SWITCH IS SET TO THE DISABLE SIDE.





NOTE TO THE USER

TO ASSURE SAFETY IN DESIGN AND CONSTRUCTION OF ROBOT SYSTEM, READ ' SAFETY MANUAL ' FIRST.

Introduction

Thank you very much for purchasing a Mitsubishi "Super E Series" industrial robot. This instruction manual explains items that are not covered in the User's manual, such as how to install and handle optional equipment, how to use the MOVEMASTER commands, and what the commands are about. Before you begin operating the robot, make sure you read this manual thoroughly, and do not attempt to operate it until you have completely understood all of the contents in this manual. The written material in this manual is compatible with the software versions shown below:

*Controller: Ver. B3
*Teaching Box: Ver. B2

<How to use the various manuals and what each of them contains>

This product comes with 4 different manuals. You can find below the contents and purpose of each of these manuals. Use them according to their applications. Since the manuals were designed to apply to all models, any differences in the specifications from model to model will be noted.

Safety Manual

The Safety Manual explains common safety precautions and caution items related to the usage of the robots, the system designs, and the fabrication of the robots, so that the safety of all the workers connected to the robot can be protected.

Specifications Manual

The Specifications Manual is valid for the entire E series. It explains standard specifications, special factory shipment specifications, option configurations, and maintenance components. It also explains safety and technical precautions when setting up the robot.

User's Manual

The User's Manual explains everything from unpacking, installation, and safety precautions for using the robot, to cabling, origin setting, and basic operations. It also explains the basic ways to use the "teaching playback method," one of the operating methods.

Reference Manual

The Reference manual explains options installations and handling methods not covered in the User's Manual, specifications of structural equipment, the teaching playback and the Movemaster command operating methods, commands, connections to I/O equipment, tooling specifications, and maintenance and inspection. In the appendix, you can find lists of commands, parameters, and alarms, and some sample programs.

(Outline of the contents of each of the manuals)

Contents	Specification Manual	User's Manual	Reference Manual
*Standard specifications, special factory shipment specifications, and maintenance parts	○		
*Specifications and types of options	○		
*Safety and technical precautions for setting up the robot	○		
*Safety precautions during usage		○	
*Unpacking the product, installation, setting the origin, and other setup operations		○	
*Basic operations using the teaching playback method		○	
*Installation and handling of optional equipment			○
*Detailed specifications for configuration machinery			○
*Applications of the teaching playback method and the Movemaster command method			○
*Explanation of commands			○
*Connections to I/O devices, tooling specifications, and maintenance and inspection			○
* Sample programs and lists of commands, parameters, and alarms			○

Contents

Chapter 1 : Handling the robot

This chapter explains handling procedures and usage points for optional equipment not covered in the User's Manual.

1.1 Handling the robot

This chapter explains how to install the optional equipment

1.1.1 Installing the 4E-HM01 motor-operated hand

For details on how to install the motor-operated hand, refer to section 3.8 " Installing the motor-operated hand" in the User's Manual.

1.1.2 Installing the 4E-HP01/4E-HP01E pneumatic hand set

The configuration for the pneumatic hand set is shown in Fig. 1.1 and in Table 1.1

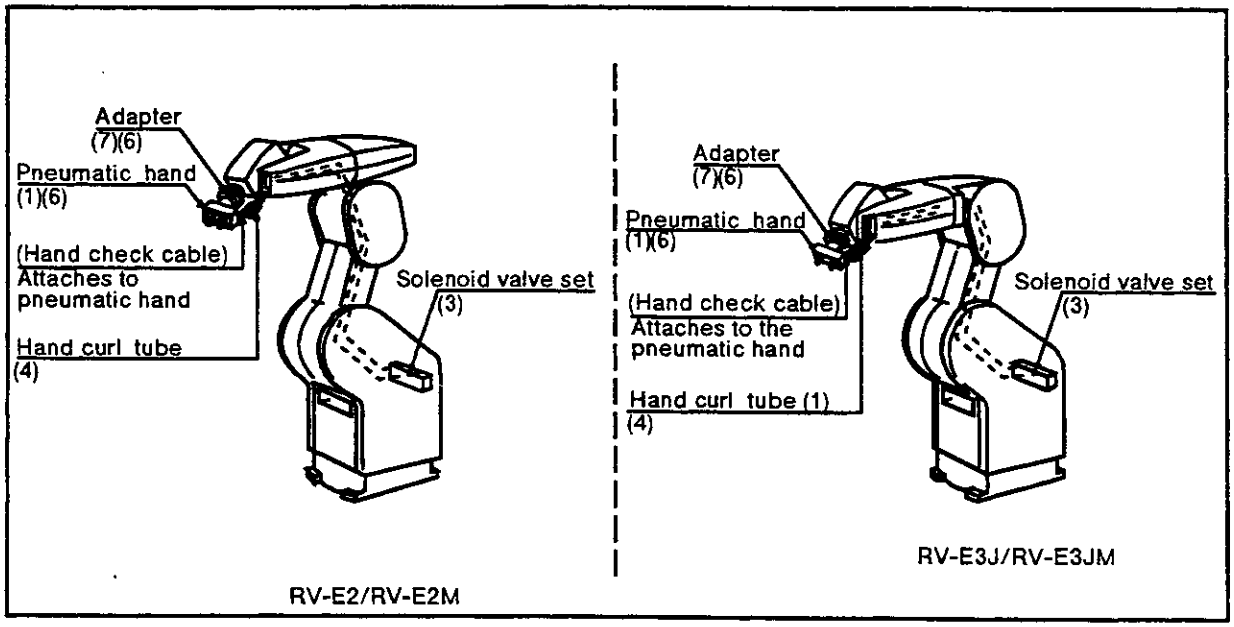


Fig. 1.1 Schematic diagram for the pneumatic hand set

Table 1.1 Configuration for the hand set

Number	Part name	Model	Quantity	Description
1	Pneumatic hand	1E-HP01/ 1E-HP01E	1	Hand input cable with pneumatic coupling
2	Pneumatic hand interface	2E-31HND/ 2E-31HNE	1	Interface board
3	Solenoid valve set	1E-VD01/ 1E-VD01E	1	Comes with (2) M3x25 socket head bolts
4	Hand curl tube (2 tubes per set)	1E-ST0402C	1	
5	Socket head bolts	M5x16	4	
6	Socket head bolts	M3x12	4	Comes with spring washers
7	Adapter	BU144D697 H01	1	This item is needed for installing (1) above.

1.1.2.1 Installing the 1E-HP01/1E-HP01E pneumatic hand

Fig. 1.2 shows the method for installing the pneumatic hand. The installation steps are as follows :

(1) Use four M5x16 hexagon socket head bolts to install the hand adapter (7) to the main robot unit's mechanical interface.

(2) Use four M3x12 hexagon socket head bolts (6) to install the pneumatic hand to the hand adapter. Use Fig. 1.3 to find the where the ends of the cables should be connected to the adapter.

(3) As shown in Fig. 1.2, connect the hand curl tubes to coupling 1 and coupling 2, which protrudes from the cover of the forearm. The opposite end of the tube connected to coupling 1 should be attached to the hand's OPEN coupling, and the opposite end of the tube connected to coupling 2 should be attached to the hand's CLOSED coupling.

(4) As shown in Fig. 1.2, connect the hand check cable, which is attached to the main unit of the hand, to the connector CON1, which protrudes from the cover of the forearm. This completes the installation of the hand.

1.1.2.2 Installing the 1E-VD01/1E-VD01E solenoid valve set

To install the solenoid valve set, refer to section 1.1.3"Installing the solenoid valve set" in this manual.

1.1.2.3 Installing the 2E-31HND/2E-31HNE pneumatic hand interface

To install the pneumatic hand interface, refer to section 1.1.4 "Installing the pneumatic hand interface" in this manual.

1.1.2.4 Setting the parameters

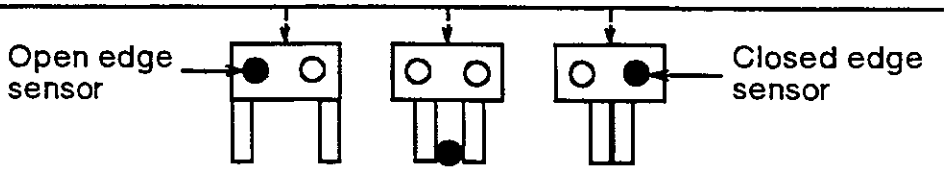
For the pneumatic hand (1E-HP01/1E-HP01E) , the tool data is 107mm. Refer to section 2.6 "Other functions, (11) Parameter settings" in this manual and set the value of the XTL parameter to 107mm.

1.1.2.5 Open/close setting of the hand and settings for input signals

The connections for the optional pneumatic hand sensor, and the bit settings for input signals and for the hands open/close setting, are shown in Table 1.2. The hand connector numbers have been assigned numbers from the connector pins located at the end of the forearm. To make the bit settings for the open/close setting of the hand and for the input signals for hand you are preparing for use, refer to chapter 4, Fig. 4.3 "Hand circuit specifications" in the Specifications Manual.

Table 1.2 The hand's open/close setting and input signal settings

Hand check connector numbers	Signal	Setting pins for the open/close setting of the hand			Comments
		Open	Half open	Closed	
1	General purpose input 900	0(ON)	1	1	Corresponds to hand 1
2	General purpose input 901	1	1	0(ON)	



1.1.2.6 Confirming operability

(1) Operate the teaching box to confirm that the hand opens and closes.

(2) To operate the teaching box, refer to section 2.2.2 "How to operate the teaching box" in this manual.

(3) If the open/close settings are reversed, confirm the connection settings with Table 1.3 and reset the settings.

⚠ Caution

While operating the hand, depending on the pitch axis and the roll axis, the hand curl- tube and the hand check cable can get entangled with the hand adapter and forearm. In this case, temporarily remove the hand, and reposition it to its correctly installed position, as shown in Fig. 1.3.

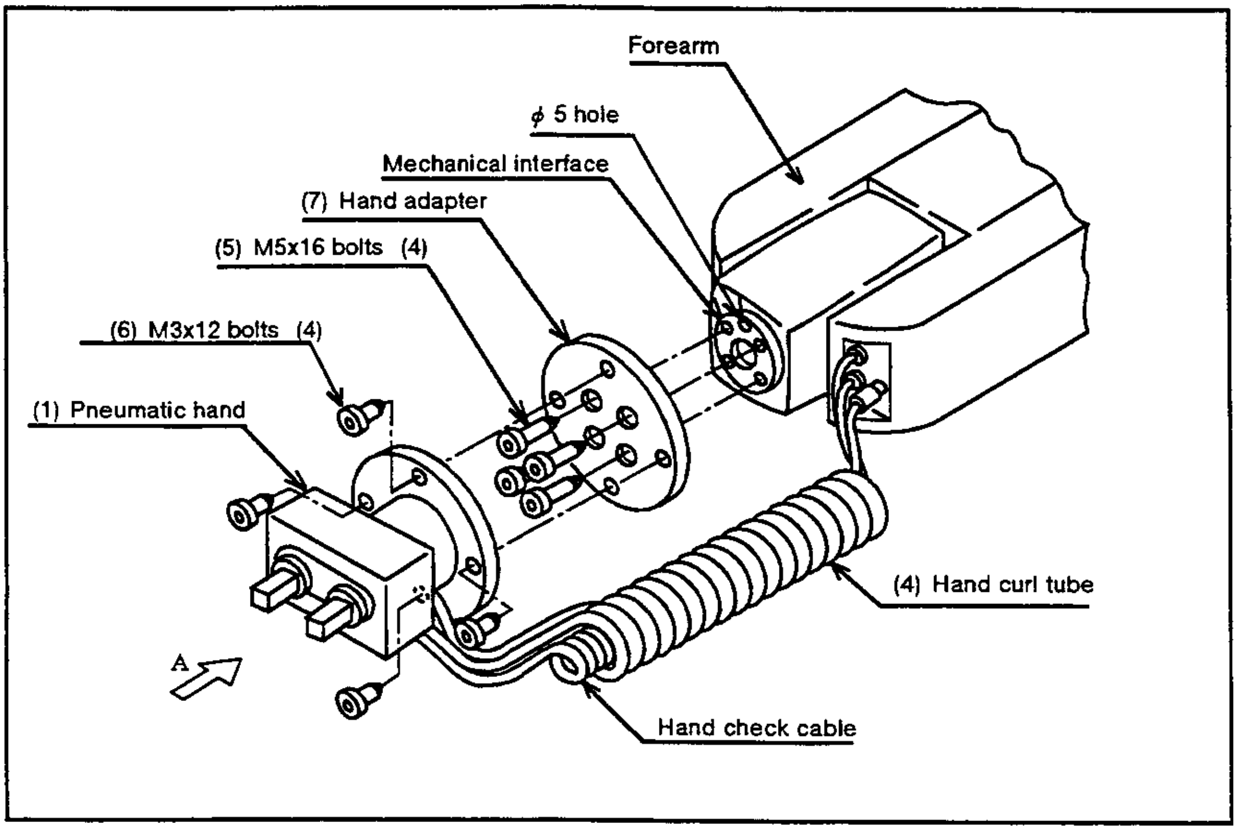


Fig. 1.2 How to install the pneumatic hand

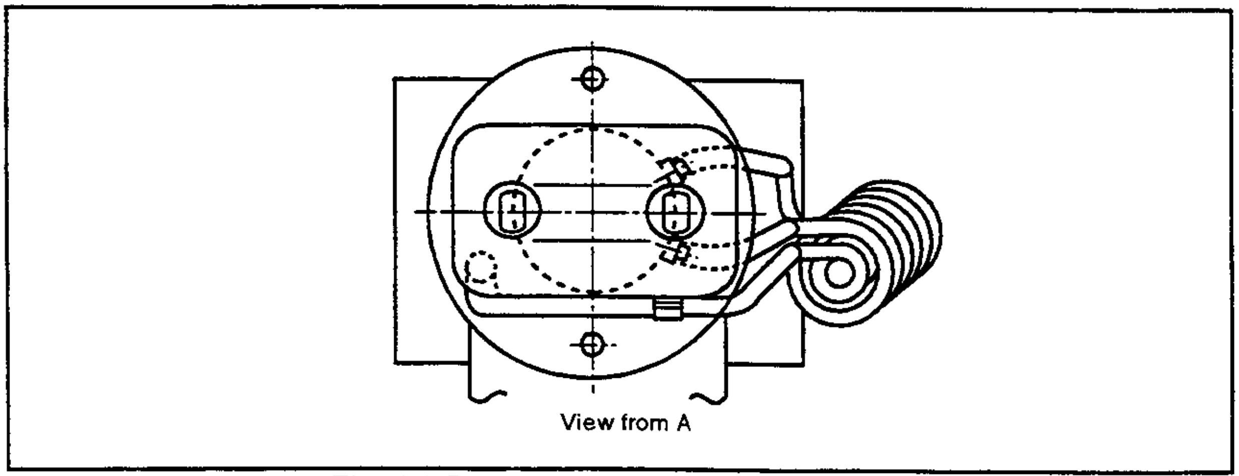


Fig. 1.3 Position of correctly installed hand

Tip

There is no mechanical stopper for the roll axis. When installing the pneumatic hand, if the servo is OFF, do not rotate the roll axis more by more than its operation range (+/- 200 degrees) . If you rotate it more than its operation range and turn ON the servo, there may be an encoder alarm.

1.1.3 Installing the 1E-VDO* solenoid valve set

1.1.3.1 Installing the 1E-VD01/1E-VD02 solenoid valve set for the RV-E2, RV-E3J

Fig. 1.4 shows how to install the solenoid valve.

The installation method is as follows:

(1) Remove the two (M3x6) screws in (2) and the two (M3x8) bolts in (3) , and remove the shoulder cover (B) in (1) .

(2) Install the solenoid valve by fastening it with the two attached (M3x25) screws in (5) to the screw holes in (4) on the top of the plate of the main robot unit.

(3) Connect one of the two (ϕ 6) pneumatic hoses in section A --the one marked "AIR IN"-- to the quick coupling (port P) in (9) of the solenoid valve in (6) . The primary air flow can now be supplied from the pneumatic port at the base.

(4) Connect the hose marked "1" to port A (7) on the primary solenoid valve (6) .
Connect the hose marked "2" to port B (8) on the primary solenoid valve (6) .
If you are using the double type valves (1E-VD02) , you need to do the following:
Connect the hose marked "3" to port A (11) on the secondary solenoid valv (10) .
Connect the hose marked "4" to port B (12) on the secondary solenoid valv (10) .

(5) Remove the plugs extending from the forearm which are connected to connectors GR1 to GR4.
Connect the GR1 plug protruding from the primary solenoid valve (6) to the connector GR1.
Connect the GR2 plug protruding from the primary solenoid valve (6) to the connector GR2.
If you are using double type valves (1E-VD02) ,
Connect the GR3 plug protruding from the secondary solenoid valve (10) to the connector GR3.
Connect the GR4 plug protruding from the secondary solenoid valve (10) to the connector GR4.

(6) When you have completed the installation, reinstall the shoulder cover (B) in (1) to its original position, and be careful not to entangle the cables when you do so.

The connections after the installation appear as in Table 1.3. For single type valves, hand 2 is not applicable.

Table 1.3A Solenoid valve ports and hoses: Connection table for couplings and hand ports for the RV-E2, RV-E3J

Hand	Hand port	Forearm coupling number	Hose number	Solenoid valve port	
Hand 1	OPEN	1	1	A	First set
	CLOSE	2	2	B	
Hand 2	OPEN	3	3	A	Second set
	CLOSE	4	4	B	

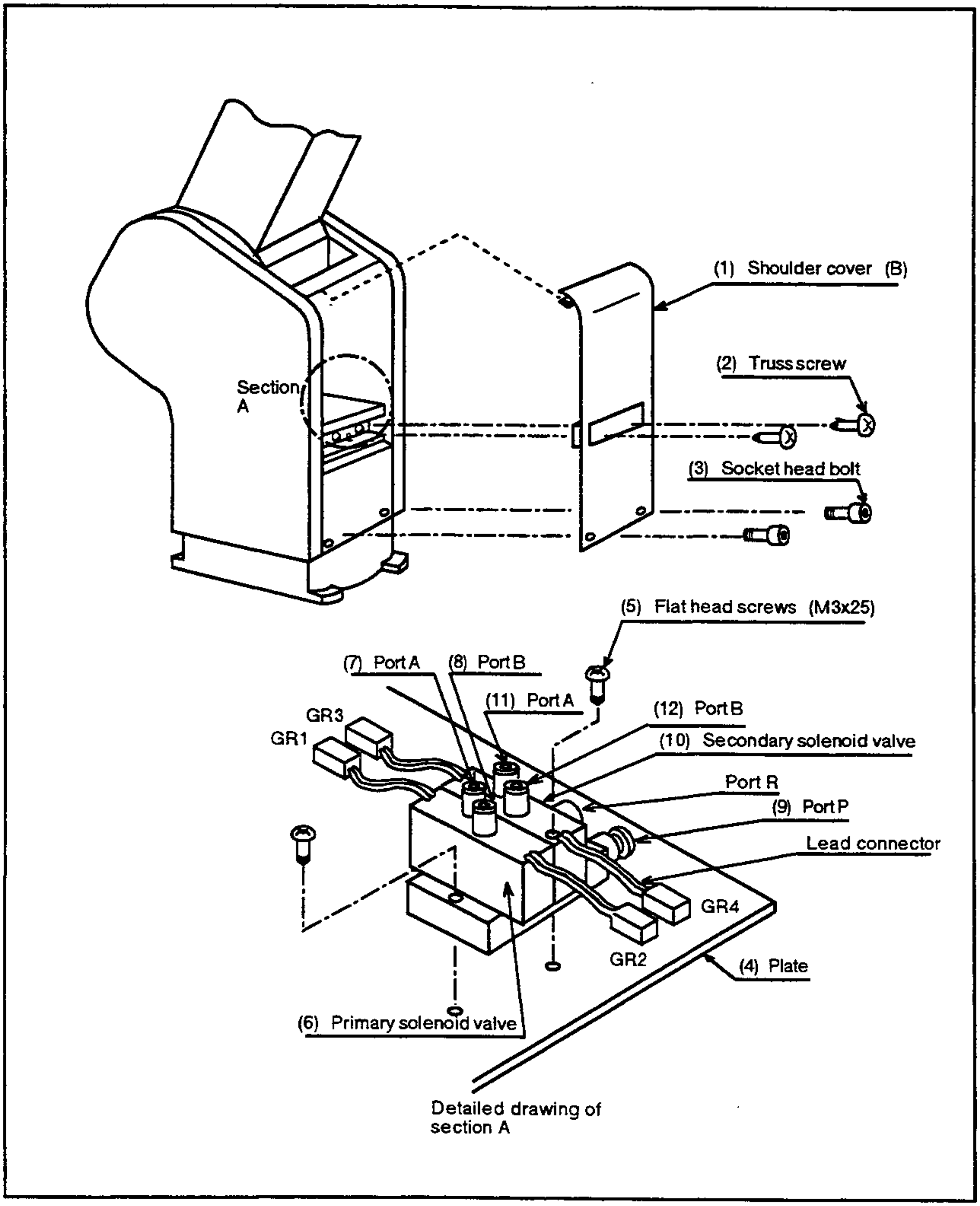


Fig. 1.4A How to install the solenoid valve for the RV-E2, RV-E3J

1.1.3.2 Installing the 1E-VD01/1E-VD02 solenoid valve for the RV-E2M, RV-E3JM

Fig. 1.4B shows how to install the solenoid valve. The installation method is as follows:

(1) Remove the two (M3x6) screws in (2) and the two (M3x8) bolts in (3) , and remove the shoulder cover (B) in (1) .

(2) Unfasten the two (M3x6) bolts in (11) holding the battery cover in (10) , and remove the cover.

(3) Disconnect the hand's output connector which is stored in the battery cover.

(4) Install the solenoid valve by fastening it with the two attached (M3x25) screws in (5) to the screw holes in (4) on the top of the plate of the main robot unit.

(5) Connect one of the two (ϕ 6) pneumatic hoses in section A -the one marked "AIR IN"-to the quick coupling (port P) in (9) of the solenoid valve in (6) . The primary air flow can now be supplied from the pneumatic port at the base.

(6) Connect the hose marked "1"to port A (7) on the primary solenoid valve (6) .
Connect the hose marked "2" to port B (8) on the primary solenoid valve (6) .
If you are using the double type valves (1E-VD02) , you need to do the following :
Connect the hose marked "3" to port A (11) on the secondary solenoid valve (10) .
Connect the hose marked "4" to port B (12) on the secondary solenoid valve (10)
.

(7) Remove the plugs extending from the forearm which are connected to connectors GR1 to GR4.
Connect the GR1 plug protruding from the primary solenoid valve (6) to the connector GR1.
Connect the GR2 plug protruding from the primary solenoid valve (6) to the connector GR2.
If you are using double type valves (1E-VD02):
Connect the GR3 plug protruding from the secondary solenoid valve (10) to the connector GR3.
Connect the GR4 plug protruding from the secondary solenoid valve (10) to the connector GR4.

(8) Store the newly attached connectors in the battery cover in (13) , and reinstall the battery cover.

(9) When you have completed the installation, reinstall the shoulder cover (B) in (1) to its original position, and be careful not to entangle the cables when you do so.

The connections after the installation appear as in Table 1.3B. For single type valves, hand 2 is not applicable.

Table 1.3B Solenoid valve ports and hoses : Connection table for couplings and hand ports for the RV-E2M, RV-E3JM

Hand	Hand port	Forearm coupling number	Hose number	Solenoid valve port	
Hand 1	OPEN	1	1	A	First set
	CLOSE	2	2	B	
Hand 2	OPEN	3	3	A	Second set
	CLOSE	4	4	B	
Spare		5	5		
		6	6		

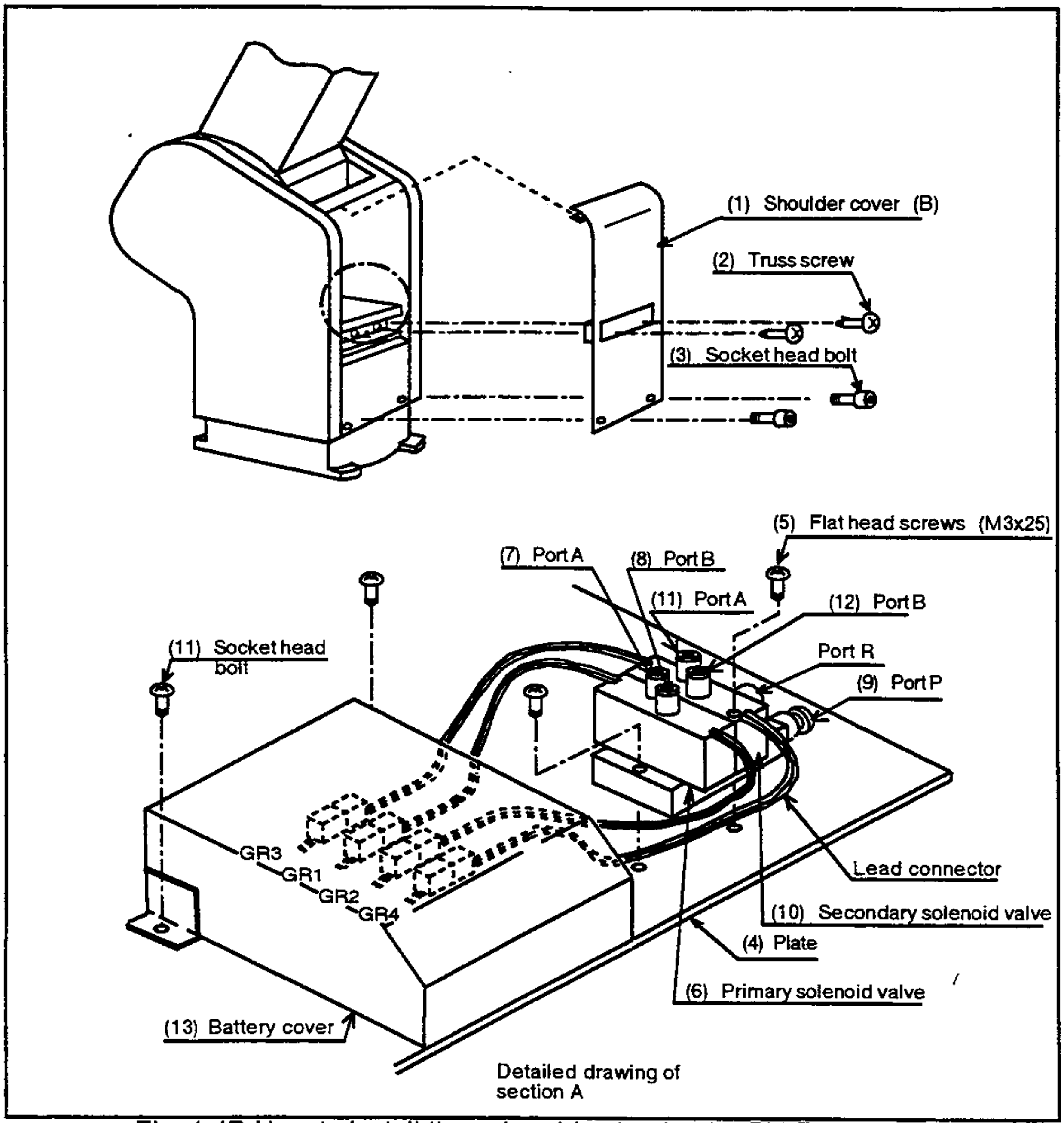
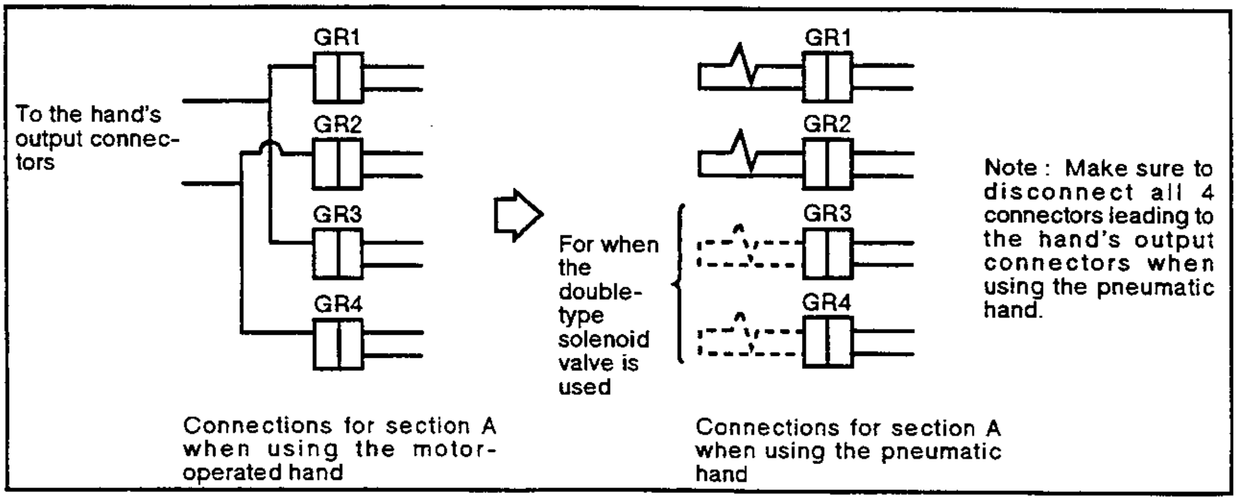


Fig. 1.4B How to install the solenoid valve for the RV-E2M, RV-E3JM


GR1
GR2
GR3
GR4
To the hand's output connectors
For when the double-type solenoid valve is used
GR1
GR2
GR3
GR4
Connections for section A when using the motor-operated hand
Connections for section A when using the pneumatic hand

Note : Make sure to disconnect all 4 connectors leading to the hand's output connectors when using the pneumatic hand.

Fig. 1.5 How to install the solenoid valve connectors

1.1.4 Installing the 2E-31HND/2E-31HNE pneumatic hand interface

Fig. 1.6 shows how to install the pneumatic hand interface

(1) Turn OFF the controller power. Also, turn OFF the power source.

(2) After turning the power OFF, wait about 3 minutes for the charged parts to discharge.

(3) Remove the top cover of the controller.

(4) Remove the two fixing screws (1) on the right-most slot on the back of the controller. Remove the dust cover for the expansion slot. (Store it away.)

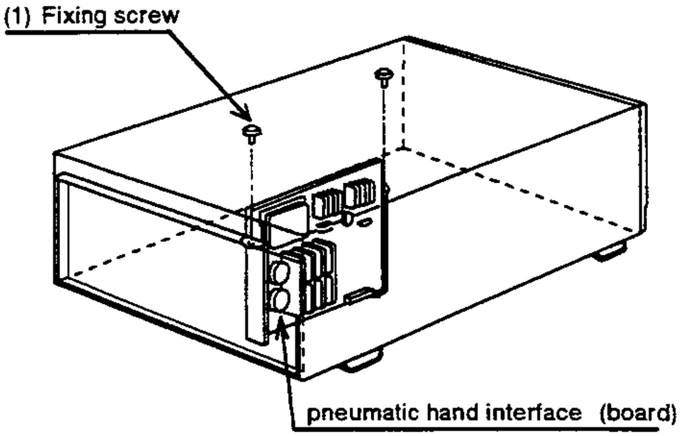
(5) Insert the pneumatic hand interface into the connector of the above-mentioned slot.

(6) Fasten and secure the two fixing screws (1) .

(7) Reinstall the top cover.

(8) Turn ON the controller power.

(9) If the hand is attached, with the teaching box, use jog operations to confirm whether the pneumatic hand can open and close.



⚠Caution

Wait at least 3 minutes after turning the power off before removing the top panel. Don't turn the power on until the top panel has been installed. Other methods of handling may lead to electric shocks.

Fig. 1.6 Installing the motor-operated hand interface board

1.1.5 Installing the P6TB-TE teaching box

For the installation method, refer to chapter 3 "Preliminary preparations" in the User's Manual.

1.1.6 Installing the 2E-31IO/2E-31IOE parallel I/O interface

Fig. 1.7 shows how to install the parallel I/O interface board.

(1) Turn OFF the controller power. Also, turn OFF the power source.

(2) After turning OFF the controller power, wait about 3 minutes for the charged parts to discharge.

(3) Remove the top cover of the controller.

(4) Remove the two fixing screws (1) on the slot for the parallel I/O interface board and remove the blind caps (2) for the expansion slot. (Store it away.)

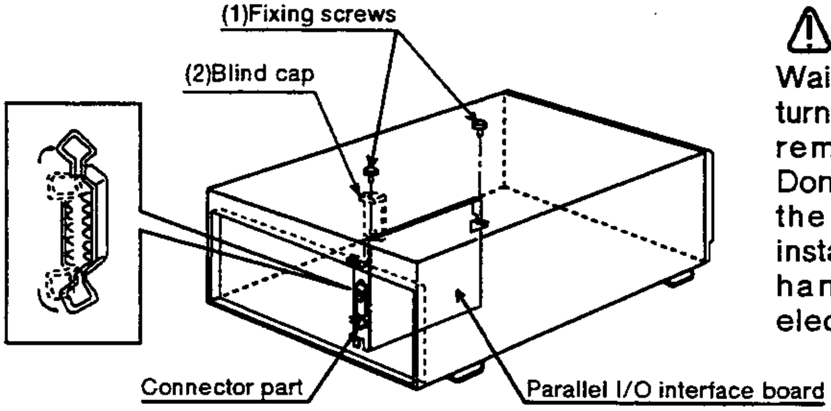
Remove the dust cover on the expansion slot's connector. (Store it away.)

(5) Insert the parallel I/O interface board into the connector of the above-mentioned slot. Option boards other than hand interface boards should be installed in the order of OPT2 and OPT3.

(6) Securely tighten the two fixing screws shown in (1) .
(7) Install the top cover of the controller.

⚠Caution

When you insert the I/O interface board, open up the hardware in the direction of the arrows, as shown in Fig. 1.7 so that you don't catch onto the hardware of the connector part.



⚠Caution

Wait at least 3 minutes after turning the power off before removing the top panel. Don't turn the power on until the top panel has been installed. Other methods of handling may lead to electric shocks.

Fig. 1.7 Installation of parallel I/O interface (board)

1.1.7 Installing the rack adapter (Model type : 2E-RACK)

Fig. 1.8 shows the procedure for installing the rack adapter.

(1) Remove the three fixing screws (M4*8) on the left side of the front panel of the controller.
(2) With the screws that are removed, attach the rack adapter to the side of the controller.
(3) Do the right side in the same fashion. The rack adapters sre identical and can be used for either the left or right sides.

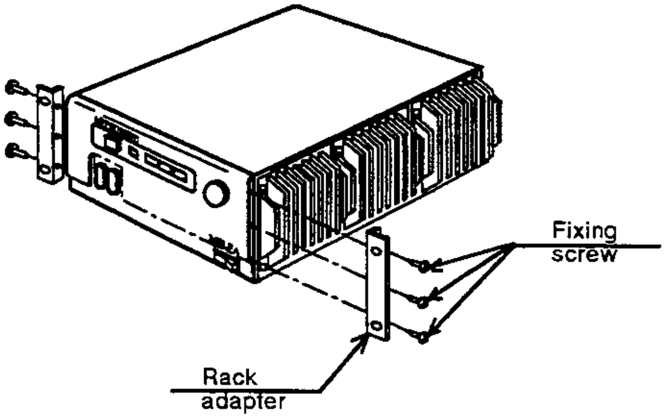


Fig. 1.8 Procedure for installing the rack adapter

⚠Caution

Do not use the rack adapter as the sole means to support the controller (approx. 28kgf) . Use a side rail or use other such means to provide overall support for the controller.

1.1.8 Installing the RS-**-CBL personal computer cable

Fig. 1.9 shows how to install the personal computer

(1) Confirm that the personal computer cable is made for your personal computer.
(2) Attach the personal computer connector to the controller's RS-232C connector.
(3) Securely fasten the screws above and below the connector so that they will not come off.
(4) Be careful about exposing the cables. Pulling strongly on the cables or bending them excessively can damage the connectors or rupture the cables.

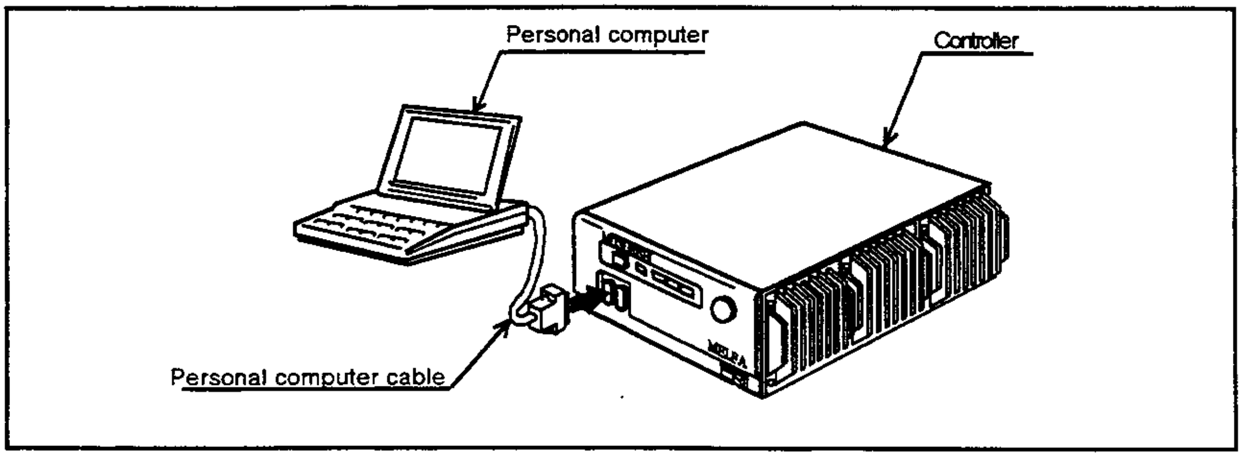


Fig.1.9 How to install the personal computer

GS (Go Sub)

[Function]

Carries out subroutine beginning with the specified line number.

[Input Format]

```
GS  [‹line number›]  [,  [‹program name›]]
```

[Term]

‹Line number› Specify line number of subroutine in integer value.
1 ≦ line number ≦ 9999

‹Program name› Specify program name of subroutine in integer value or characters. (Less than 8 characters)
1 ≦ program name ≦ 8 (characters)
Possible letter used : Digit (0-9)
Character (A - Z)
Symbol (! @ # ,etc.)
Impossible letter used : ＊+,. ／: ; [¥]' "
Special specification :When you specified only numeric value, the program name is handled as number.
Need to enclose program name with " " in the case of character used.

[Explanation]

(1) Allows the program to jump to the specified line of the specified program and execute subroutine. The program returns to the main program after executing the subroutine. When you specified program number, returns to the main program by ED command and when you specified only line number, returns by RT command.

(2) Use the RT command to terminate the subroutine existing in the same program. Use the ED command to terminate the subroutine existing in other program.

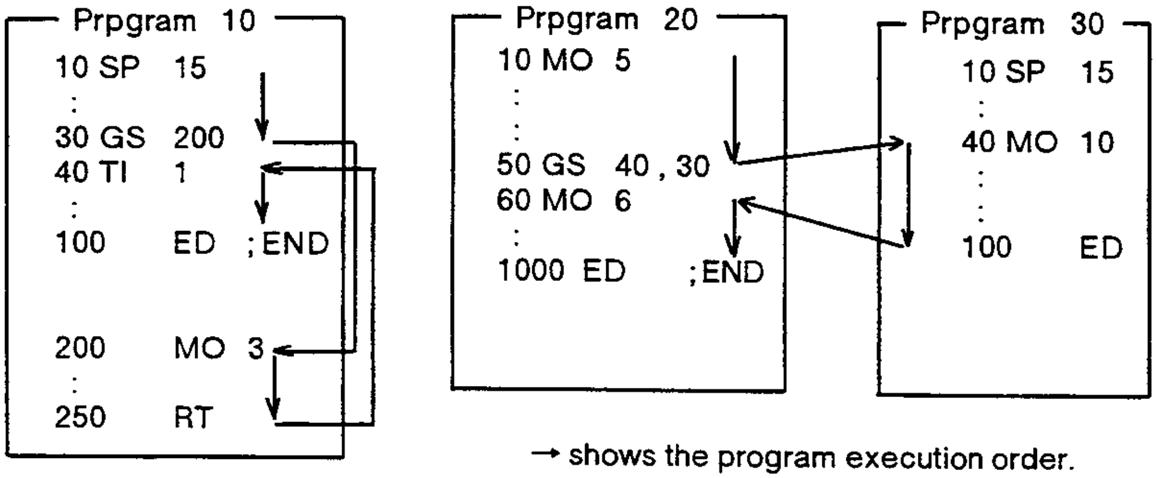
(3) If the specified line or the specified program does not exist, alarm occurs at the time of GS execution.

(4) When you omitted line number, executes the specified program from the top line.

(5) When you omitted line number and program name, nothing occurs .

(6) To call subroutines in other subroutines is called "nesting". Up to 9 nesting levels are possible.

‹The same program call› ‹Program to program call›



```
Prpgram 10
10 SP  15
 :
30 GS  200
40 TI  1
 :
100    ED  ;END

200    MO  3
 :
250    RT
```

```
Prpgram 20
10 MO  5
 :
 :
50 GS  40 , 30
60 MO  6
 :
1000  ED    ;END
```

```
Prpgram 30
10 SP   15
 :
40 MO   10
 :
 :
100     ED
```

→ shows the program execution order.

※ In the above example of ‹The same program call›, executes the program from line 10 to 30, then calls the subroutine of line 200. When the RT command is executed in the subroutine, the program returns to the main program and continues from line 40. The program ends when the ED command is executed.

※ Program can call other program from inside the program using GS command. In the above example of ‹Program to program call›, executes the program from line 20 to 50, then calls the program 30. Executes the program from line 40 to 100 and returns to the main program, i.e., line 60 of program 20. The program ends when the ED command is executed.

【 Sample program 】 (Movemaster command)

```
10  GS100          ; Carry out subroutine beginning with line number 100.
    :
90  ED             ; Ends program.
100 MO   11        ; Moves to position 11. ⎫
110 MO   12        ; Moves to position 12. ⎬    Subroutine
120 MO   13        ; Moves to position 13. ⎪
130 RT             ; Ends subroutine       ⎭
```

GT (Go To)

【 Function 】

Jumps to the specified line number unconditionally.

【 Input Format 】

```
GT ‹line number›
```

【Term】

‹Line number› Specify the line number to which the program jumps.
1 ≦ line number ≦ 9999

【 Explanation 】

(1) Causes the program to jump to the specified line number.

(2) If the specified line number does not exist, alarm occurs at the time of GT execution.

【 Sample program 】 (Movemaster command)

```
10  MO 1            ; Moves to position 1.
20  GT 100          ; Jumps to line 100 unconditionally.
     :
100 MO 12           ; Moves to position 12.
110 MO 15           ; Moves to position 15.
     :
```

HE (Here)

[Function]

Defines the current coordinates as the specified position.

[Input Format]

```
        HE  ‹position number›
```

[Term]

‹Position number›   Specify the position number to be registered.
0 ≦ position number ≦ 999
Registers the current position to the user-defined origin in case of zero.

[Explanation]

(1) The coordinates of the current position are calculated on the basis of the currently set tool length (see the TL command) . In the initial condition, the tool length is 123 mm, a point away from the hand mounting surface toward the end of the hand.

(2) If a single number is assigned to two different positions, the one defined last takes precedence with the former cleared.

(3) The open/close position of the hand and the structure flag data are also stored as the position data.

(4) Alarm occurs if the HE command is executed before the origin setting.

(5) When you specified zero position number, current position data in joint coordinates are defined to user-defined origin parameter UOG. In this case, you must change the permission parameter HOE to permit the origin setting at first. After that, return the permission parameter to the former value. (Does not permit the setting.) The above operation is effective only by direct command execution.

[Sample program] (Movemaster command

```
10 MO  10        ; Moves to position 10.
20 DW  10,0,0    ; Moves to +X direction by 10 mm
30 HE  11        ; Defines above location as position 11.
```

HLT (Halt)

【 Function 】

Interrupts the motion of the robot and the operation of the program.

【 Input Format 】

```
        HLT
```

【 Explanation 】

(1) Interrupts the operation of the program and decelerates the robot to a stop. (It becomes the same condition that the external stop signal is input or the STOP switch of the controller front panel is pushed.)

(2) To restart the program, push the START switch, input the starting signaling, or execute the RN command. Program restarts from the next line of HLT command.

(3) If the HLT command is directly executed from the personal computer during program running, the program is interrupted and the robot stops with deceleration.

(4) The robot does not stop by the HLT command, however, during the execution of the direct motion command.

【 Sample program 】 (Movemaster command)

```
10 MO  1          ; Moves to position 1.
20 HLT            ; Stops
30 MO  2          ; Moves to position 2.
40 ED             ; Ends program.
```

The program restarts with START switch from line 30.

HO (Home)

【 Function 】
Defines the current location and the attitude as origin point.

【 Input Format 】

```
        HO   [‹origin setting approach›]
```

【Term】
‹Origin setting approach›   Specify the method to set origin in integer value.
0 : Mechanical stopper origin
1 : Jig origin
2 : User-defined origin

【 Explanation 】
(1) Establishes the reference position for origin setting.
(2) If you have replaced the robot or changed the combination of robot and controller, you must carry out origin setting again using this command. There is another approach using teaching box to execute origin setting. See "1.2: origin setting" for detail.
(3) Change the parameter HOE to permit origin setting at first, then execute the HO command directly. When the origin setting is completed, return the parameter to the former value, otherwise the program can not be started.To find the change operation, refer to section 2.6 "Other Functions (11) Setting the parameters" in this Manual.

【 Relating Parameters 】
Permits the origin setting from the command (HO) .
Parameter name HOE : Origin setting permission parameter
0 : Does not permit the use of HO command. (Default)
1 : Permits the use of HO command.

【 Sample program 】 (BASIC)

```
10 OPEN" COM1 : E83" AS #1  ;Opens the RS-232C communication file from the
                            personal computer in BASIC.
20 PRINT #1," HO"           ;Executes the "HO" command from the personal
                            computer.
30 END                      ;Ends

 RUN                        ; Run the BASIC program.
```

IC (Increment Counter)

【 Function 】

Adds 1 to the value of the specified counter.

【 Input Format 】

```
        IC ‹counter number›
```

【Term】

‹Counter number›	Specify counter number in numeric value. 1 ≦ counter number ≦ 99

【 Explanation 】

(1) Alarm occurs if the counter value exceeds 32767.

(2) Used to count the number of workpieces and job sequence and to set the number of grid point in the pallet.

(3) The contents of the counter can be changed, compared, or read by the relevant command. (See SC、DC、CP、CR、CL、AN、OR、XO commands.)

【 Sample program 】 (Movemaster command)

```
10 SC  21,15        ; Sets value 15 to counter 21.
20 IC  21           ; Add 1 to the contents of counter 21.
```

ID (Input Direct)

【Function】

Fetches data unconditionally from the external input and hand check input.

【Input Format】

```
        ID  [‹input bit number›]
```

【Term】

‹Input bit number›	Specify the bit number of input port in integer value. Fetches data of 16 bits width including the specified bit. 0 ≦ input bit number ≦ 32767 (0 for default)

【Explanation】

(1) Fetches signals from the external equipment, e.g., programmable controller, unconditionally. The data from the hand check input can be fetched by specifying the 900th number to the input bit number.

(2) The fetched data is loaded into the internal register and is subsequently used for comparison, bit test, etc. (See EQ、NE、LG、SM、TB commands.)

【Sample program】 (Movemaster command)

```
100  ID                 ; Fetches the input data into the internal resister for
                          omparison.
110  EQ  100,130        ; If the input data equals 100, then jumps to line number 130.
120  ED                 ; Else ends program.
130  MO  1              ; Moves to position 1.
140  ID  100            ; Fetches the input data into the internal resister for
                          comparison. (Input signals 100 to 115.)
150  TB  +0,180         ; If the input bit 100 is ON, then jumps to line 180.
160  TB  +5,200         ; If the input bit 105 is ON, then jumps to line 200.
170  ED                 ; Else ends program.
180  MO  2              ; Moves to position 2.
190  ED                 ; Ends program.
200  MO  3              ; Moves to position 3.
210  ED                 ; Ends program.
```

INP (Input)

【 Function 】

The specified counter value, the coordinate value of the position number or the data of the specified character string is received according to the PRN command. (Using RS-232-C)

【 Input Format 】

```
INP ‹channel number›, ‹counter number/position number/character string number›
                                                        [, [‹contents selection›]]
```

【Term】

‹Channel number› Specify the channel number opened by the OPN command.
0 ≦ channel number ≦ 2

‹Counter number› Specify counter number.
1 ≦ counter number ≦ 99

‹Position number› Specify position number.
1 ≦ position number ≦ 999

‹Character string number› Specify character string number in numerical value which $" is added to the head.
$1 ≦ character string number ≦ $99

‹Contents selection› Select either counter or position or character string number corresponding to ‹Counter number/ position number/character string number›.
0 : Counter number (Default)
1 : Position number
2 : character string number

【 Explanation 】

(1) This command receives the specified counter value, the coordinate value of the position number or the data of the specified character string is received according to the PRN command through the RS-232-C port.

(2) The OPN command must be executed first to open the RS-232-C channel.

(3) If the counter number is omitted, the data will be read into the internal register. If the character string number is omitted, the data will be read into the character string register. If the position number is omitted, an alarm will occur during execution.

(4) The data is sent from an external device such as a personal computer using the PRM command. The robot program will stop while the data is being red.

(5) The PRM command can be executed before the INP command while the program is running. In that case, the sent PRM command data will be registered once, and then will be led into the specified counter, position or character string when the INP command is executed.A max. of 256 characters can be registered in the robot. If the PRM command is executed in succession and the number of registered characters exceeds 256 characters, the robot will be set to the "L" level based on the RS-232-C ER (DRT) and RS (RTS) signal lines (DR (DSR) and CS (CTS) signal lines on the personal computer side) . Temporarily stop the data transmission from the personal computer during this time.

(6) If there is an error in the data sent by the PRN command, an alarm will occur when the INP command is executed.

【 Sample program 】 (Movemaster command)

```
10 OPN   2,1        ; Opens the RS-232C port.
20 INP   2,1,0      ; Reads the data of counter 1 from the RS-232C port.
30 INP   2,5,1      ; Reads the data of position 5 from the RS-232C port.
40 IC    1          ; Adds 1 to the contents of counter 1.
50 MO    5          ; Moves to position 5.
60 OPN   1,1        ; Opens the RS-232C port.
70 INP   1,$10,2    ; Reads the data of character string 10 from the RS-232C port.
```

IP (Increment Position)

【 Function 】

Moves the robot to a predefined position with a position number greater than the current one. (Joint interpolation)

【 Input Format 】

```
        IP
```

【 Explanation 】

(1) Moves the robot to a predefined position with a position number greater than, and closest to, the current one. (See the DP command.)

(2) Alarm occurs if there is no predefined position which is greater in position number than the current position.

(3) Even if an alarm occurs, the current position number still remains unchanged.

【 Sample program 】 (Movemaster command)

```
10 MO  5          ; Moves to position 5.
20 MO  4          ; Moves to position 4.
30 MO  3          ; Moves to position 3.
40 IP             ; Moves to position 4.
50 IP             ; Moves to position 5.
```

JRC (Joint roll change)

[Function]

Overwrites the current position by adding +/-360 degrees to the joint position of the R-axis. This is done when you want to use shortcut control of the R-axis, or when you want to use endless control.

[Input format]

```
        JRC‹ [+] 1/-1›
```

[Terms]

‹+1›   Adds 360 degrees to the current joint position on the R-axis.
‹-1›   Subtracts 360 degrees to the current joint position on the R-axis.

[Explanation]

(1)When you use this command, you need to change the operation range of the R-axis by +/- 720 degrees beforehand. Set the -J6 setting to -720 on the joint operation range parameter JAR, and +J6 to 720. For information on how to change parameters, refer to section 2.6 Other functions" under (11) " Parameter settings." If you set a value that is mor than +/- 720 degrees, it could result in position slips.

(2)Even though the current coordinate value changes, the robot won't move.

(3)If the joint coordinate value on the R-axis exceeds +/- 720 degrees, the current coordinate value won't change, and the out-of-bounds alarm will be generated.

(4)If you are using the RV-E2/E2M, and you want to move the R-axis by more than +/ - 180 degrees with this command, the move up to +/- 180 degrees should be by joint interpolation commands. If you use linear interpolation commands or circular interpolation commands, when you execute the program, alarms will be generated. If you are using RV-E3J /RV-E3JM, you don't have such a restriction.

(5)If you use this command while doing continuous movements without acceleration/ deceleration (passing) , it will not be able to do continuous movements with acceleration/deceleration. (Refer to CNT of the SP command)

(6)When the controller power is a disconnected state, if you move the R axis by more than +/- 7 turns, the next time the power source is supplied, the position of R's axis will slip. In this case, it is necessary to reset the origin for only the R-axis.
Refer to 3.9 "Setting the origin" in the User's Manual and 1.2 "Setting the origin" in this manual, and reset the origin.

[Sample program] (with Movemaster commands)

```
10  MO   1              ; Move to position 1  (R-axis is -150 degrees)
20  MO   2              ; Move to position 2  (R-axis is +170 degrees)
30  JRC  -1             ; Subtract 360 degrees from the current position on the
                          R-axis (R-axis is -190 degrees)
40  MO   1              ; Move to position 1
```

LG (If Larger)

【 Function 】

This compares the value of the internal register with a specified value. If larger, the program will jump. The character string register and the numbers of characters in a specified character string are compared. If the character string register is larger, the program will jump.

【 Input Format 】

```
LG ‹compared value/character string number›, ‹branching line number›
```

【Term】

‹Compared value› Specify the value compared with the internal register.
-32768 ≦ Compared value (decimal) ≦ 32767
& 8000 ≦ Compared value (hexadecimal) ≦ & 7FFF

‹Character string number› Specify character string number in numerical value which "$" is added to the head.
$1 ≦ character string number ≦ $99

‹Branching line number› Specify the line number to which the program jumps when the value of the internal register is larger than compared value.
1 ≦ branching line number ≦ 9999

【 Explanation 】

‹When compared value is specified›

(1) Causes a jump to occur conditionally in accordance with the external input data or the internal counter value.

(2) If the internal register value is larger than the compared value (i.e., when the condition is met) , the program jumps to the specified line. Otherwise (i.e., when the condition is not met) , the program continues in sequence.

(3) A value can be loaded into the internal register by executing the input command (See ID) for the external input data or by executing the compare counter command (See CP) for the counter data. Accordingly when you carry out conditional branching, need to execute either of the above commands beforehand.

(4) The compared value may be defined either in decimal or hexadecimal. A hexadecimal value must be headed by "&".

‹When character string number is specified›

(1) The conditions will jump depending on the data input from an external source or the number of characters in a specified character string.

(2) If the number of characters in the character string register is larger than the number of characters in a specified character string (when the conditions are established) , the program will jump to the specified line number. If the number is smaller (when conditions are not established) , the next line will be executed. If the specified line number is not registered, an alarm will occur when jumping.

(3) By executing an INP command, the data input from an external device will be set in the character string register. The details of the character string number will be set by executing a CP command. Thus, when executing condition jumping, one of these commands must be executed first.

【 Sample program 】 (Movemaster command)

```
100  ID           ; Fetches the data from the external input port.
110  LG  100,130  ; If the input data is larger than 100, jumps to line 130.
120  ED           ; Else program ends.
130  MO  1        ; Moves to position 1.
140  OPN 1,1      ; Opens the RS-232C port.
150  INP  1, ,2   ; Reads the data of character string register from the RS-
                    232C port.
160  LG  $5,200   ; Jumps to line 200 if the data length large than character
                    string number 5.
 :
200  ED           ; Ends program.
```

LR ※ (Line Read)

[Function]

Reads the program of the specified line number. (Using RS-232C)

[Input Format]

```
        LR   [<line number>]
```

[Term]

<Line number>   Specify the line number to be read.
0 ≦ line number ≦ 9999 (If omitted, reads the current line number stopping)

[Explanation]

(1) Outputs the program of the specified line number (or the current stopping line number) from the RS-232C port.

(2) The output format is ASCII coded as follows ;
- If you specify the line number, ----------------- Program content is read.
- If you omit the line number (or specify zero) -- Current stopping line number is read.

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

(4) The hexadecimal 0D is read out when the specified line has not been defined.

(5) If an alarm takes place, you can confirm the line number in which the alarm occurs by executing the LR command without line number.

[Sample program] (BASIC)

```
10  OPEN" COM1 : E83" AS#1      ; Opens the RS-232C communication file from
                                the personal computer in BASIC.
20 INPUT "Start line " ;        ; Enter the top line number that you want to read.
30 INPUT "End line" ;           ; Enter the last line number that you want to read.
40 FOR I=S TO E                 ; Repeatedly
50  PRINT #1," LR" +STR$ (I)    ; Transmit "LR" + "line number" to the controller
                                side.
60  LINE INPUT #1,A$            ; Saves the received data to A $.
70  IF A$="" THEN 90            ; If there is no data, jumps to line 90.
80  PRINT I ; : PRINT A$        ; Displays the data on the personal computer
                                screen.
90 NEXT                         ; Repeats and jumps to line 40.
100 END

RUN                             ; Run the BASIC program.
```

MA (Move Approach)

【 Function 】

Moves the hand tip to the added position. (Linear interpolation)

【 Input Format 】

```
MA ‹position number (a) ›, ‹position number (b) › [, [‹O/C›]]
```

【Term】

‹Position number (a) ›	Specify the position number to be added. (The reference position)
‹Position number (b) ›	Specify the position number to add. (The increments position) 1 ≦ position number (a) (b) ≦ 999
‹O/C›	Specify open or close state of the hand. O : Hand open C : Hand close

【 Explanation 】

(1) Moves to the added position, i.e., the coordinates of positions (a) and (b) are added to make the destination, although positions (a) and (b) remain unchanged after executing the MA command. (See the SF command.)

(2) If the open/close state of the hand has been specified, the robot moves after executing the hand control command. If it has not been specified, the hand state in position (a) remains valid.

(3) If the calculating results exceed the robot's operational space, alarm occurs before the robot moves.

(4) Alarm also takes place if positions (a) and (b) have not been defined.

(5) The position of the hand tip is decided by the tool length currently established.

【 Sample program 】 (Movemaster command)

(1) RV-E2/RV-E2M

```
10 HE  1              ; Sets the current coordinates to position 1.
20 PD  5,0,0,30,0,0,0 ; Defines the Z coordinate of position 5 as 30 mm.
30 MA  1,5,O          ; Moves to the position that only Z direction added to
                        coordinate value of position 1 by 30 mm with the hand opened.
```

※ Coordinates values of position 1 and position 5 do not change.

(2) RV-E3J/RV-E3JM

```
10 HE  1              ; Sets the current coordinates to position 1.
20 PD  5,0,0,30,0,0   ; Defines the Z coordinate of position 5 as 30 mm.
30 MA  1,5,O          ; Moves to the position that only Z direction added to
                        coordinate value of position 1 by 30 mm with the hand opened.
```

※ Coordinates values of position 1 and position 5 do not change.

MC (Move Continuous))

【 Function 】

Moves the robot continuously through the predefined intermediate points between two specified position numbers. (Linear interpolation)

【 Input Format 】

```
MC ‹position number (a) ›, ‹position number (b) › [, [‹O/C›]]
```

【Term】

‹Position number (a) › Specify the top position number moving continuous.

‹Position number (b) › Specify the last position number moving continuous.
1 ≦ position number (a) (b) ≦ 999
| Position number (a) - position number (b) | ≦ 99

‹O/C› Specify open or close state of the hand.
(If omitted, the hand data of each position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) Moves the robot along the series of positions via (a) to (b) without acceleration and deceleration. (Linear interpolation)

(2) Depending on whether position number of (a) is greater than that of (b) , or vice versa, the robot moves through the intermediate points in descending or ascending order. The robot decelerates to a stop as it reaches the end position.

(3) When the hand open/close setting has been done, hand control is executed before the movement.

(4) Since the robot does not accelerate or decelerate during motion, alarm may occur when the path involves a great change in direction of any of the joints at high speed.

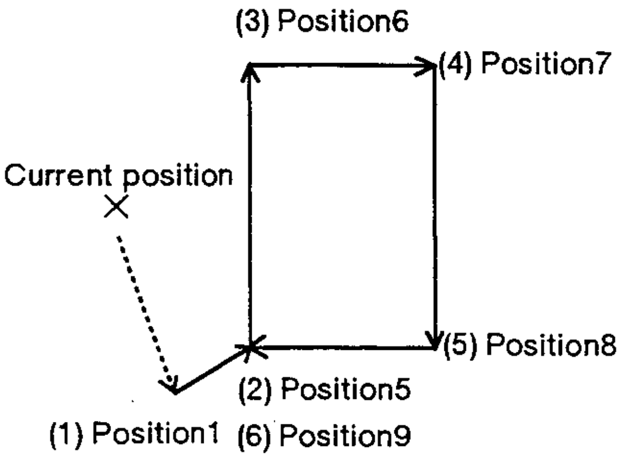
(5) The speed of travel during linear interpolation is determined by the SP or SD command. (Hand tip at constant speed)

(6) Alarm occurs if specified positions (a) and (b) have not been defined or if the difference between the position numbers (a) and (b) exceeds 99.

(7) Alarm also takes place during movement if the movement path goes beyond the robot's operational space.

【 Sample program 】 (Movemaster command)

```
10 SP   10          ; Sets speed to 10.
20 MO   1           ; Moves to position 1 in joint interpolation.
30 MC   5,9         ; Moves continuously from position 5 to 9 in linear interpolation.
```



MJ (Move Joint)

【 Function 】

Turns each joint the specified angle from the current position. (Joint interpolation)

【 Input Format 】

(1) RV-E2/RV-E2M

```
MJ     [‹waist joint angle›] , [‹shoulder joint angle›] , [‹elbow joint angle›] ,
       [‹twist joint angle›] , [‹pitch joint angle›] , [‹roll joint angle›]
```

(2) RV-E3J/RV-E3JM

```
MJ     [‹waist joint angle›] , [‹shoulder joint angle›] , [‹elbow joint angle›] ,
       [‹pitch joint angle›] , [‹roll joint angle›]
```

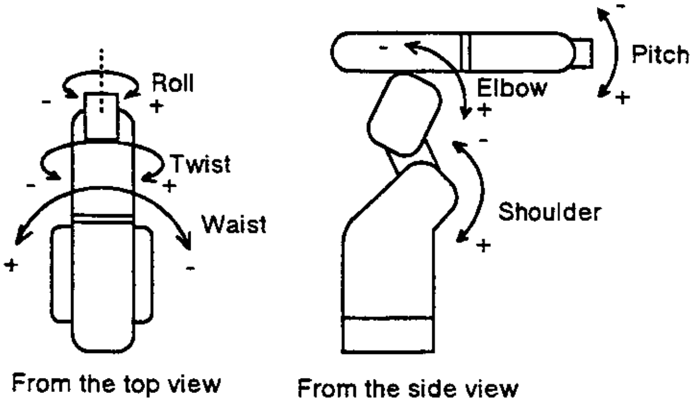
【Term】

‹Each joint angle› Specify relative amount of each joint turning from the current position.

【 Explanation 】

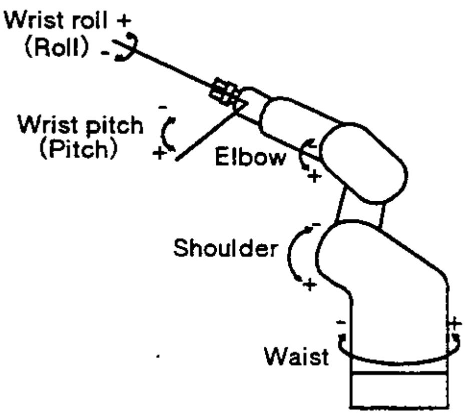
(1) The least increment of the turning angle is 0.01 degree, e.g., specify 15.02 for 15.02 degree.
(2) The open/close state of the hand does not change before and after the movement. Alarm occurs before the joint motion if any turning angle entry exceeds the robot's operational space.
(3) The default turning angle is 0.
(4) The positive and negative directions of each joint of motion are as follows;

RV-E2/RV-E2M



From the top view   From the side view

RV-E3J/RV-E3JM



【 Sample program 】 (Movemaster command)

(1) RV-E2/RV-E2M

```
10 MJ  90,0,0,0,0,0      ;Turns the waist joint + 90 degrees.
20 MJ  0,-30,0,0,0,0     ;Turns the shoulder joint - 30 degrees.
```

(2) RV-E3J/RV-E3JM

```
10 MJ  90,0,0,0,0        ;Turns the waist joint + 90 degrees.
20 MJ  0,-30,0,0,0       ;Turns the shoulder joint - 30 degrees.
```

MO (Move)

【 Function 】

Moves the hand tip to the specified position. (Joint interpolation)

【 Input Format 】

```
        MO ‹position number› [, [‹O/C›]]
```

【Term】

‹Position number› Specify the destination position number in integer value.
1 ≦ position number ≦ 999

‹O/C› Specify open or close state of the hand. (If omitted, the hand state of the position is valid)
O : Hand open
C : Hand close

【 Explanation 】

(1) Moves the tip of hand to the coordinates of the specified position by joint interpolation.The hand tip is decided by the tool length currently established.

(2) If open/close state of the hand has been specified, the robot moves after executing the hand control command.

(3) If it has not been specified, the definition of the specified position is executed.

(4) Alarm takes place if the specified position has not been predefined or the movement exceeds the robot's operational space.

【 Sample program 】 (Movemaster command)

```
10 SP   10          ; Sets speed to 10.
20 MO   20,C        ; Moves to position 20 with hand closed.
30 MO   30,O        ; Moves to position 30 with hand opened.
```

MP (Move Position)

【 Function 】

Moves the tip of hand to a position whose coordinates (position and angle) have been specified. (Joint interpolation)

【 Input Format 】

(1) RV-E2/RV-E2M

```
    MP     [‹ X  coordinate value›] ,  [‹ Y  coordinate value›] ,
           [‹ Z  coordinate value›] ,  [‹A turn angle›] ,  [‹B turn angle›] ,
           [‹C turn angle›]   [, [‹R/L›]   [, [‹A/B›]   [, [‹N / F›]]]
```

(2) RV-E3J/RV-E3JM

```
    MP     [‹ X  coordinate value›] ,  [‹ Y  coordinate value›] ,
           [‹ Z  coordinate value›] ,  [‹A turn angle›] ,  [‹B turn angle›] ,
           [, [‹R/L›]   [, [‹A/B›]]]
```

【Term】

‹X, Y, Z coordinate› Specify the position in XYZ coordinates (mm) of the robot. (Zero for default)

‹A, B,C turning angle› Specify the turning angle of roll and pitch joints in XYZ coordinates (degree) of the robot. (Zero for default) (RV-E2/RV-E2M only.)

‹A, B turning angle› Specify the turning angle of roll and pitch joints in XYZ coordinates (degree) of the robot. (Zero for default) (RV-E3J/RV-E3JM only.)

‹R/L› Specify the structure flag of the robot. (Right or Left)
R : Right (Default)
L : Left

‹A/B› Specify the structure flag of the robot. (Above or Below))
A : Above (Default)
B : Below

‹N/F› Specify the structure flag of the robot. (Nonflip or Flip) (RV-E2/RV-E2M only.)
N : Non flip (Default)
F : Flip

【 Explanation 】

(1) The least increment of the coordinate value is 0.01 mm or 0.01 degree.

(2) If the structure flag has not been specified, Right and Above flag is selected.

(3) If the specified value exceeds the robot's operational space, alarm occurs at the execution of the MO command.

(4) The open or close state of the hand remains the same before and after the movement.

(5) The position of hand tip is decided by the tool length currently established.

【 Sample program 】 (Movemaster command)

(1) RV-E2/RV-E2M

```
10 MP 400,0,300,0,0,0            ; Moves to the specified coordinates.
20 MP 200,200,500,0,0,0,R,A,N    ; Moves to the specified coordinates. (Structure
                                   flags are also specified.)
```

(2) RV-E3J/RV-E3JM

```
10 MP 400,0,300,0,0              ; Moves to the specified coordinates.
20 MP 200,200,500,0,0,R          ; Moves to the specified coordinates. (Structure
                                   flags  are also specified.)
```

MPB (Move Playback)

[Function]

Moves to the specified position with specified interpolation, specified speed, specified timer, and specified input and output signal.

[Input Format]

(1) RV-E2/RV-E2M

```
MPB  [‹speed›] , [‹timer›] , [‹output ON›] , [‹output OFF›] , [‹input ON›] ,
     [‹input OFF›]  [, [‹interpolation›] , [‹X coordinate›] , [‹Y coordinate›] ,
     [‹Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
     [‹C turning angle›] [, [‹R/L›]  [, [‹ A/B ›]  [, [‹ N/F ›]]]]  [, [‹O/C›]]]
```

(2) RV-E3J/RV-E3JM

```
MPB  [‹speed›] , [‹timer›] , [‹output ON›] , [‹output OFF›] , [‹input ON›] ,
     [‹input OFF›]  [, [‹interpolation›] , [‹X coordinate›] , [‹Y coordinate›] ,
     [‹Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
     [, [‹R/L›]  [, [‹ A/B ›]]]  [, [‹O/C›]]]
```

[Term]

‹Speed› Specify the interpolation speed to the destination position.
0 ≦ speed ≦ 32767 (Joint interpolation : %, Linear interpolation : mm/s)

‹Timer› Set timer at the destination position after the movement.
0 ≦ timer ≦ 255

‹Output ON› Set the output signal that turns ON.
0 ≦ output ON (Hexadecimal) ≦ & FFFF
1 : Setting, 0 : Not setting

‹Output OFF› Set the output signal that turns OFF.
0 ≦ output OFF (Hexadecimal) ≦ & FFFF
1 : Setting,0 : Not setting

‹Input ON› Set the input waiting signal that turns ON.
0 ≦ input ON (Hexadecimal) ≦ & FFFF
1 : Setting,0 : Not setting

‹Input OFF› Set the input waiting signal that turns OFF.
0 ≦ input OFF (Hexadecimal) ≦ & FFFF
1 : Setting,0 : Not setting

‹Interpolation› Specify the interpolation mode to the destination position.
Joint interpolation : 0 (Default)
Linear interpolation : 1
Circular interpolation : 2

‹X, Y, Z coordinate› Specify the location (mm) in XYZ coordinates of the robot. (Zero for default)

‹A, B,C turning angle › Specify the turning angle around X (A) , Y (B) , Z (C) axes in XYZ coordinates (degree) of the robot. (0 for default) (RV-E2/RV-E2M only.)

‹A, B turning angle › Specify the turning angle around roll (A) , pitch (B) axes in XYZ coordinates (degree) of the robot. (0 for default) (RV-E3J/RV-E3JM only.)

‹R/L› Specify the structure flag of the robot. (Right or Left)
R : Right (Default)
L : Left

‹A/B› Specify the structure flag of the robot. (Above or Below)
A : Above (Default)
B : Below

‹N/F› Specify the structure flag of the robot. (Nonflip or Flip)
(RV-E2/RV-E2M only.)
N : Nonflip (Default)
F : Flip

‹O/C› Specify open or close state of hand 1.
O : Hand 1 open (Default)
C : Hand 1 close

【 Explanation 】

(1) The least increment of the coordinate value is 0.01 mm or 0.01 degree (e.g., specify 20.01 for 20.01 mm) .

(2) Alarm occurs if the specified coordinates exceed the robot's operational space.

(3) The default structure flags are R (Right) , A (Above) , N (NonFlip) .

(4) The default coordinate value is zero.

(5) If you set any condition of speed, timer, or input/output in the teaching playback method, the MPB command is generated at each step .

(6) The input and output setting must be hexadecimal code headed with "&".

(7) If the input/output signal is not set, the bit corresponding on the teaching box's screen display will be "-".

(8) The order that the condition data specified with this command is executed is as follows.
1) Interpolation method 2) Speed 3) Movement to target position 4) Timer 5) Signal output (including hand control) 6) Signal input wait

MPC (Move Playback Continuous)

【 Function 】

Moves to the specified position with specified interpolation.

【 Input Format 】

(1) RV-E2/RV-E2M

```
MPC  [‹interpolation›] , [‹ X coordinate›] , [‹ Y coordinate›] ,
     [‹ Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
     [‹C turning angle›]  [, [‹R/L›]  [, [‹ A/B ›]  [, [‹ N/F ›]]]  [, [‹O/C›]]
```

(2) RV-E3J/RV-E3JM

```
MPC  [‹interpolation›] , [‹ X coordinate›] , [‹ Y coordinate›] ,
     [‹ Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
     [, [‹R/L›]  [, [‹ A/B ›]]]  [, [‹O/C›]]
```

【Term】

‹Interpolation› Specify the interpolation mode to the destination position.
0 : Joint interpolation (Default)
1 : Linear interpolation
2 : Circular interpolation

‹X, Y, Z coordinate› Specify the location (mm) in XYZ coordinates of the robot. (0 for default)

‹A, B,C turning angle › Specify the turning angle around X (A) , Y (B) , Z (C) axes in XYZ coordinates (degree) of the robot. (0 for default) (RV-E2/RV-E2M only.)

‹A, B turning angle › Specify the turning angle around roll (A) , pitch (B) axes in XYZ coordinates (degree) of the robot. (0 for default) (RV-E3J/RV-E3JM only.)

‹R/L› Specify the structure flag of the robot. (Right or Left)
R : Right (Default)
L : Left

‹A/B› Specify the structure flag of the robot. (Above or Below)
A : Above (Default)
B : Below

‹N/F› Specify the structure flag of the robot. (Nonflip or Flip) (RV-E2/RV-E2M only.)
N : Non flip (Default)
F : Flip

‹O/C› Specify open or close state of the hand 1.
O : Hand open (Default)
C : Hand close

【 Explanation 】

(1) The least increment of the coordinate value is 0.01 mm or 0.01 degree (e.g. specify 20.01 for 20.01 mm) .

(2) Alarm occurs if the specified coordinates exceed the robot's operational space.

(3) The default coordinate value is zero.

(4) Use the MPB command If the setting of speed, timer, or input/output are needed.

(5) If you do not set any condition of speed, timer, or input/output in the teaching playback method, the MPC command is generated at each step.

(8) The order that the condition data specified with this command is executed is as follows. 1) Interpolation method 2) Speed 3) Movement to target position 4) Timer 5) Signal output (including hand control) 6) Signal input wait

MR (Move R)

【 Function 】

Moves the tip of hand through the predefined intermediate positions in circular interpolation.

【 Input Format 】

```
MR ‹position number (a) › , ‹position number (b) ›, ‹position number (c) ›
  [, [‹ O/C ›]]
```

【Term】

‹Position number› Specify the positions on the circle.
1 ≦ position number ≦ 999

‹O/C› Specify open or close state of the hand. (If omitted, the hand state of the position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) Moves the tip of hand through specified positions from (a) via (b) to (c) drawing an arc.

(2) The moving speed of circular interpolation is decided by the SP or SD command. (The tip of hand at constant speed.) Since the locus accuracy depends on the speed of circulari nterpolation, set the moving speed lower if you need high accuracy.

(3) The open or close state of the hand does not change before and after the movement.

(4) If the starting position (a) is different from the current position, the robot moves to the starting position by linear interpolation.

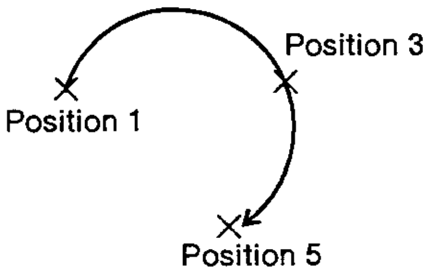
(5) If the circular interpolation is interrupted by the stop signal and restarted by the start signal, the robot moves the remaining arc. If the tip of hand is kept away from the stopping position by JOG operation in the above case, the robot moves to the stopping position by joint interpolation then moves the remaining arc.

(6) Alarm takes place if the specified position has not been predefined or exceeds the robot's operational space. The robot moves by linear interpolation if three positions (a) , (b) and (c) are located on a straight line or if two of three positions are the same.

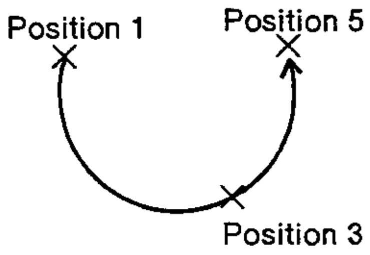
(7) If the moving direction of each joint changes greatly at the beginning of circular interpolation, alarm may occur. Set speed lower or set timer at the beginning in this case.

(8) The drawing direction and the locus of the arc depend on the order of the specified positions.

‹In the case of MR 1, 3, 5›



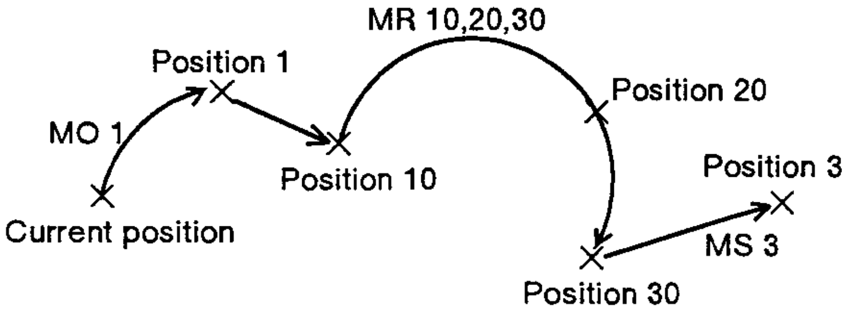
(Example 1)



(Example 2)

【 Sample program 】 (Movemaster command)

```
10 SP   8            ; Set speed to 8.
20 MO   1            ; Moves to position 1.
30 MR   10,20,30     ; Moves to position 10 by linear interpolation.
                     Moves the arc determined by position 10, 20, 30 by circular
                     interpolation.
40 MS   3            ; Moves to position 3 by linear interpolation.
50 ED                ; Ends program.
```



MRA (Move R A)

【 Function 】
Moves to the specified position in circular interpolation.

【 Input Format 】

```
        MRA ‹position number ›  [,  [‹O/C›]]
```

【Term】

‹Position number› Specify the destination position.
1 ≦ position number ≦ 999

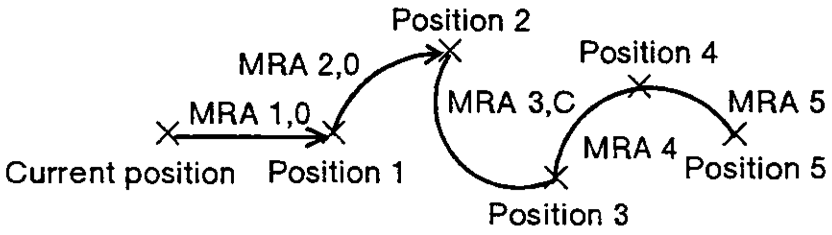
‹O/C› Specify open or close state of the hand. (If omitted, the hand state of the position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) Moves the tip of hand on the arc which is defined by the former and the latter positions of the MRA commands. The tip of hand is decided by the tool length currently established.
(2) If the open or close state of the hand has been specified, the robot moves after executing the hand control.
(3) Alarm takes place if the specified position has not been predefined.
(4) If the MRA command does not continue more than three, it becomes similar to the MC command. The following command, however, can be executed between the MRA command.
SD, SP, TI, OVR, OB, OC, OD, GC, GO
(5) If the execution of the MRA command is interrupted and the tip of hand is kept away from the stopping position by JOG operation, the robot moves, when restarted, to the stopping position by linear interpolation then moves the remaining arc.

【 Sample program 】 (Movemaster command)

```
10 MRA 1,O      ; Defines the arc with positions 1, 2, 3.
                  Moves to position 1 by linear interpolation.
20 MRA 2,O      ; Moves to position 2 by circular interpolation.
30 MRA 3,C      ; Moves to position 3 by circular interpolation.
40 TI     3     ; Timer 0.3 second.
50 MRA 4        ; Moves to position 4 by circular interpolation.
60 MRA 5        ; Moves to position 5 by circular interpolation.
70 ED           ; Ends program.
```



MS (Move Straight)

【 Function 】

Moves the tip of hand to the specified position. (Linear interpolation)

【 Input Format 】

```
        MS ‹position number› [, [‹O/C›]]
```

【Term】

‹Position number› Specify the destination position number in integer value.
1 ≦ position number ≦ 999

‹O/C› Specify open or close state of hand. (If omitted, the hand state of the position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) Moves the tip of hand to the specified position by linear interpolation.The tip of the hand is decided by the tool length currently established. (See the TL command.)

(2) Alarm occurs before or during movement if the destination or movement path goes beyond the robot's operational space.

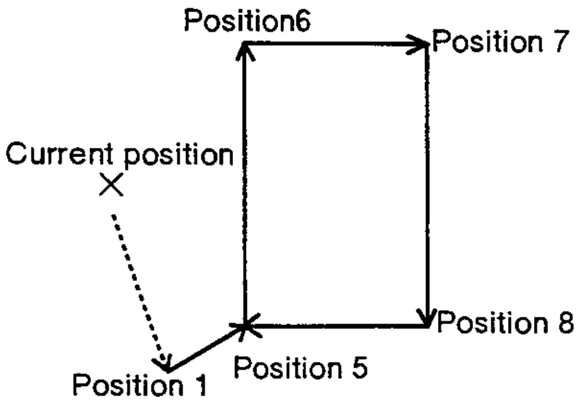
(3) If the open or close state of the hand has been specified, the robot moves after executing the hand control.

(4) The moving speed is decided by the SP or SD commands. (The tip of hand at constant speed.)

(5) Use the MC command to move continuously through several positions by linear interpolation.

【 Sample program 】 (Movemaster command)

```
10 SP  15        ; Sets speed to 15.
20 MO  1         ; Moves to position 1 by joint interpolation.
30 MS  5         ; Moves to position 5 by linear interpolation.
40 MS  6         ; Moves to position 6 by linear interpolation.
50 MS  7         ; Moves to position 7 by linear interpolation.
60 MS  8         ; Moves to position 8 by linear interpolation.
70 MS  5         ; Moves to position 5 by linear interpolation.
```



MT (Move Tool)

【 Function 】

Moves the tip of hand to a position away from the specified position by the distance as specified in the tool direction. (Joint interpolation)

【 Input Format 】

```
        MT ‹position number›, [‹travel distance›]   [, [‹O/C›]]
```

【Term】

‹Position number› Specify the destination position number in integer value.
1 ≦ position number ≦ 999

‹Travel distance› Specify the distance in tool direction from the specified position to the destination point. (Zero for default)
-3276.80 ≦ travel distance ≦ 3276.70

‹O/C› Specify open or close state of the hand. (If omitted, the hand state of the position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) The least increment of the distance is 0.01 mm.

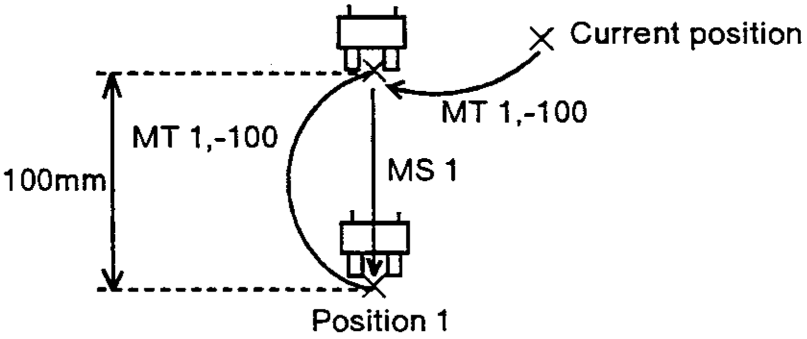
(2) When the distance is positive, the tip of hand advances in the tool direction. When the distance is negative, the tip of hand retracts in the tool direction.

(3) If the open or close state of the hand has been specified, the robot moves after executing the hand control.

(4) Alarm occurs when the MT command is executed if the specified position has not been predefined or if the destination exceeds the robot's operational space.

【 Sample program 】 (Movemaster command)

```
  10 MT   1,-100        ; Moves to the point away from the position 1 by 100 mm.
  20 MS   1             ; Moves to position 1.
  30 MT   1,-100        ; Moves to the point away from the position 1 by 100 mm.
```



MTS (Move Tool Straight)

【 Function 】

Moves the tip of hand to a position away from the specified position by the distance as specified in the tool direction. (Linear interpolation)

【 Input Format 】

```
MTS ‹position number›, [‹travel distance›]  [, [‹O/C›]]
```

【Term】

‹Position number› Specify the destination position number in integer value.
1 ≦ position number ≦ 999

‹Travel distance› Specify the distance in tool direction from the specified position to the destination point. (Zero for default)
-3276.80 ≦ travel distance ≦ 3276.70

‹O/C› Specify open or close state of the hand. (If omitted, the hand state of the position is valid.)
O : Hand open
C : Hand close

【 Explanation 】

(1) The least increment of the distance is 0.01 mm.

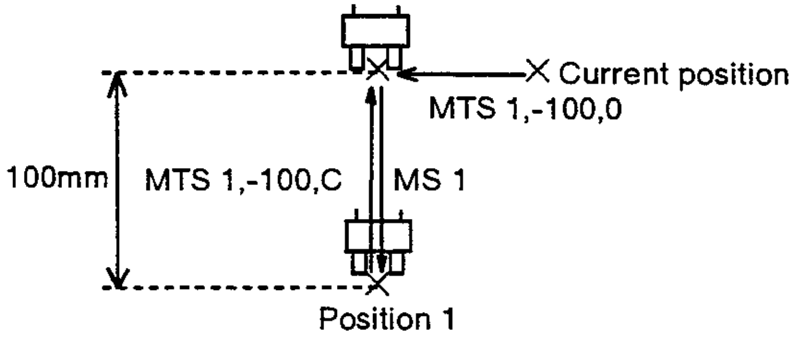
(2) When the distance is positive, the tip of hand advances in the tool direction. When the distance is negative, the tip of hand retracts in the tool direction.

(3) If the open or close state of the hand has been specified, the robot moves after executing the hand control.

(4) Alarm occurs when the MT command is executed if the specified position has not been predefined or if the destination exceeds the robot's operational space.

【 Sample program 】 (Movemaster command)

```
10  MTS  1,-100,O    ; Moves to the point away from the position 1 by 100 mm with
                       hand opened. Moves straight.
20  MS   1           ; Moves to position 1.
30  MTS  1,-100,C    ; Moves to the point away from the position 1 by 100 mm with
                       hand closed. Moves straight.
```



N ※ (Number)

【 Function 】
Select the specified program.

【 Input Format 】

```
        N ‹program name›
```

【Term】

‹Program name› Specify the robot program name. (Less than 8 characters)
Possible letter used : Digit (0-9)
Character (A - Z)
Symbol (! @ # ,etc)
Impossible letter used : ＊+,. ／： ；＝？ [¥]' "
Special specification : When you specified only numeric value, is handled for number.
Need to enclose program name with " " in the case of character used.

【 Explanation 】

(1) Select the specified program. The program selected here becomes an object of the implementation, modification and operation. The program selected once remains till other program number is selected afresh. (Even if the power turns OFF, the program number remains unchanged.)
(2) You can confirm the current program number using the QN command from the personal computer. (See the QN command.)
(3) Program 1 is selected at product line shipping.
(4) The following name is identified as the same.
An example : Handled as the same.   1,01,001,00000001 (Only numeric value)
Handled as not the same. 1,1 A, A0 _ 001 (Includes characters)
(5) The letters that controller can indicate to the LED are 0-9, A ~ Z (simplified) .

【 Sample program 】 (BASIC)

```
10 OPEN " COM1 : E83"  AS #1     ; Opens the RS-232C communication file in
                                 BASIC.
20 PRINT #1," N10"               ; Selects the program 10.
30 PRINT #1," 10 MO 1"           ; Implementation of program.  (Line 10)
40 PRINT #1," 20 MS 2"           ; Implementation of program.  (Line 20)
50 PRINT #1," 30 ED"             ; Implementation of program.  (Line 30)
60 END                           ; Ends
```

NE (If Not Equal)

【 Function 】

This compares the value of the internal register with a specified value. If not equal, the program will jump. The character string register and details of a specified character string are compared. If not equal, the program will jump.

【 Input Format 】

NE ‹compared value/character string number›, ‹branching line number›

【Term】

‹Compared value› Specify the value that the internal register compares contents with.
-32768 ≦ compared value (decimal) ≦ 32767
& 8000 ≦ compared value (hexadecimal) ≦ &7FFF

‹Character string number› Specify character string number in numerical value which "$" is added to the head.
$1 ≦ character string number ≦ $99

‹Branching line number› Specify the line number to which the program jumps when the comparison result is not equal.
1 ≦ branching line pair ≦ 9999

【 Explanation 】

‹When compared value is specified›

(1) Causes a jump to occur conditionally in accordance with the external input data or the internal counter value.

(2) If the internal register value does not equal to the compared value (i.e. when the condition is met), the program jumps to the specified line. Otherwise (i.e. when the condition is not met), the program continues in sequence.

(3) A value can be loaded into the internal register by executing the input command (See ID) for the external input data or by executing the compare counter command (See CP) for the counter data. Accordingly when you carry out conditional branching, need to execute either of the above commands beforehand.

(4) The compared value may be defined either in decimal or hexadecimal. A hexadecimal value must be headed by "&".

‹When character string number is specified›

(1) The conditions will jump depending on the data input from an external source or the number of characters in a specified character string.

(2) If the number of characters in the character string register is not equal to the details of a specified character string (when the conditions are established), the program will jump to the specified line number. If not (when conditions are not established), the next line will be executed.

(3) By executing an INP command, the data input from an external device will be set in the character string register. The details of the character string number will be set by executing a CP command. Thus, when executing condition jumping, one of these commands must be executed first.

(4) If the specified line number is not registered, an alarm will occur when jumping.

【 Sample program 】 (Movemaster command)

```
  10  ID              ; Fetches data from external input port.
  20  NE   80,100     ; Jumps to line 100 if the data does not equal 80.
  30  ED              ; Ends the program if above condition is not met.
  100 MO  7           ; Moves to position 7.
  110 OPN 1,1         ; Opens the RS-232C port.
  120 INP 1, ,2       ; Reads the data of character string register from the RS-
                      232C port.
  130 NE  $2,200      ; Jumps to line 200 if the data not equals character string
                      number 2.
    :
  200 ED              ; Ends program.
```

NT (Nest)

【 Function 】

Carry out origin setting. (The robot moves to the user-defined origin.)

【 Input Format 】

```
        NT
```

【 Explanation 】

(1) The moving sequence of each joint is fixed beforehand. Origin setting of the shoulder, the elbow and the twist joint is first executed, which is followed by that of the waist, the pitch, and the roll joint.
Note.) RV-E3J/RV-E3JM don't have the twist joint.

(2) If the arm can interfere with the object surrounding the robot, use the teaching box to move it to a safe location before origin setting.

(3) You can change the moving sequence of the origin setting by the parameter UNG, and also can change the attitude by the parameter UOG. The parameter UOG can be set with the teaching box operation, too. (See section 1.2"Origin setting", and the "HE" command.)

(4) The NT command is not required in the usual operation.

【 Relating Parameters 】

The sequential order of user origin setting can be changed by the following parameter.

```
Parameter name  UNG : Sequential order
                      2,1,1,1,2,2   (Default)
```

The attitude of user origin setting can be changed by the following parameter.

```
Parameter name  UOG : User origin attitude  (degree)
                      -160.00, -45.00, 50.00, -160.00, -120.00, -200.00
                                                   (Default RV-E2/RV-E2M)
                      -160.00, -45.00, 0.00, 0.00, -120.00, -200.00
                                                   (Default RV-E3J/RV-E3JM)
```

To find the change operation, refer to section 2.6 "Other Functions (11) Setting the parameters" in this Manual.

【 Sample program 】 (Movemaster command)

```
10 NT       ; Executes origin setting.
20 MO 1     ; Moves to position 1.
30 ED       ; Ends program.
```

NW ※ (New)

【 Function 】

Deletes the specified program and position data.

【 Input Format 】

```
        NW
```

【 Explanation 】

(1) Deletes all positions and counters of the specified program. Common positions (901-999) and common counters (91-99) , however, are not deleted.

(2) Origin setting, internal register, tool length, speed setting, pallet setting, and hand setting remain unchanged even if the NW command is executed.

(3) The NW command can not be executed in the program with line number. (Only direct execution is possible.)

【 Sample program 】 (BASIC)

```
10 OPEN " COM1 : E83" AS #1 ;Opens the RS-232C communication file from the
                             personal computer in BASIC.
30 PRINT #1," NW"           ;Transmit command "NW"
60 END                      ;Ends program.
```

NX (Next)

【 Function 】
Specifies the range of a loop in a program executed by the RC command.

【 Input Format 】

```
        NX
```

【 Explanation 】

(1) Used in combination with the RC command to specify the range of a loop in a program executed by the RC command.
(2) Alarm occurs if there is no corresponding "RC" command specified.

【 Sample program 】
See the RC command.

OB (Output Bit)

【 Function 】
Sets the output state of the specified bit through an external output port.

【 Input Format 】

```
        OB  [‹+/-›]  ‹bit number›
```

【Term】

‹+/-›   Set ON or OFF state of the specified bit.
                  + : Bit ON
                  - : Bit OFF
‹Bit number› Specify the bit number of external output.
                  0 ≦ bit number ≦ 32767

【 Explanation 】

(1) Set "+" to switch on the specified bit and "-" to switch off the specified bit.

(2) All bits other than the specified one are not affected by this command. The output state of the specified bit is retained until a new setting is made by the OB or OD command.

(3) If you specify the special bit in the parameter OT1-OT3 of the external output, alarm occurs because of the conflict of signals.

(4) For the pneumatic hand, you can specify open or close state of hand 1 and 2 by the OB command. Refer to the table below. (Ordinary, use the GC or GO command.) For the motor-operated hand, these settings are not possible.

Notice : GR1-GR4 shows connector number of hand output cable in the robot arm.

Hand	Open/Close	GR1	GR2	GR3	GR4
		Output bit 900	Output bit 901	Output bit 902	Output bit 903
Hand 1	Open (GO 0)	ON	OFF	—	—
	Close (GC 0)	OFF	ON	—	—
Hand 2	Open (GO 1)	—	—	ON	OFF
	Close (GC 1)	—	—	OFF	ON

【 Sample program 】 (Movemaster command)

```
10 OD  &FFFF        ;Turns the bits  (0-15)  of external output into ON entirely.
20 OB  -10          ;Turns only bit 10 to OFF.
30 ED               ;Ends program.
```

OC (Output Counter)

[Function]

Outputs the specified counter value unconditionally through the output port.

[Input Format]

```
OC ‹counter number› [, [‹output bit›] [, [‹bit width›]]]
```

[Term]

‹Counter number› Specify the counter number to be output.
1 ≦ counter number ≦ 99

‹Output bit number› Specify the reference bit number of output data
0 ≦ bit number ≦ 32767 (0 for default)

‹Bit width› Specify bit width of output data.
1 ≦ bit width ≦ 16 (16 for default)

[Explanation]

(1) Outputs the specified counter value unconditionally through the output port. The output data retains after that.

(2) Even if the OC command is executed, the value of the specified counter and the internal register remain intact.

(3) You can specify the range of output signal by setting the bit width of the OC command.

[Sample program] (BASIC)

```
10 OPEN " COM1 : E83"  AS #1  ; Opens the RS-232C communication file from the
                                personal computer in BASIC.
20 PRINT #1," SC 5,&0008"     ; Set value 8 to counter 5
30 PRINT #1," OC 5 "          ; Outputs the value of the counter 5 unconditionally
                                to output port.
40 END                        ; Ends program.
```

OD (Output Direct)

【 Function 】
Outputs the specified data unconditionally through the output port.

【 Input Format 】

```
OD ‹output data› [, [‹output bit number›] [, [‹bit width›]]]
```

【Term】

‹Output data› Specified output data.
-32768 ≦ output data (decimal) ≦ 32767
& 8000 ≦ output data (hexadecimal) ≦ & 7FFF

‹Output bit number› Specify the reference bit number of output data.
0 ≦ bit number ≦ 32767 (0 for default)

‹Bit width› Specify the bit width of output data.
1 ≦ bit width ≦ 16 (16 for default)

【 Explanation 】

(1) Outputs a signal (parallel data) unconditionally through the output port to external equipment such as a programmable controller. The output data retains after that.

(2) output data is defined either in decimal or hexadecimal. The data defined in hexadecimal must be headed by "&".

(3) For information on connections, see 4.1.2 "Interfacing with external I/O equipment".

(4) You can specify the range of output signal by setting the bit width of the OD command..

(5) If you specify the special bit in the parameter OT1-OT3 of the external output, alarm occurs because of the conflict of signals. (If you want to reset output signal with factory default setting, execute "OD 0,0,13 " causing the output bit from 0 to 12 to OFF.)

【 Sample program 】 (Movemaster command)

```
10 OD  &FFFF          ; Sets the output port of 16 bits width from bit 0 to ON.
20 OD  &FFFF,10       ; Sets the output port of 16 bits width from bit 10 to ON.
30 OD  &FFFF,10,15    ; Sets the output port of 15 bits width from bit 10 to ON.
40 ED                 ; Ends program.
```

OG (Origin)

【 Function 】
Moves to the user-defined origin. (Joint interpolation)

【 Input Format 】

```
        OG
```

【 Explanation 】

(1) Moves to the user-defined origin specified by the parameter UOG by joint interpolation.

(2) The attitude of the robot, after executing the OG command, is the same as the attitude after executing the NT command. The parameter UOG, which defines the moving sequence of origin setting, does not effect the OG command.

(3) The parameter UOG can be set with the teaching box operation, too.
(See section 1.2"Origin setting", and the "HE" command.)

【 Relating Parameters 】
The attitude of user origin setting can be changed by the following parameter.
Parameter name UOG : User origin attitude

```
                    -160.00, -45.00, 50.00, -160.00, -120.00, -200.00
                                                  (Default RV-E2/RV-E2M)
                   -160.00, -45.00, 0.00, 0.00, -120.00, -200.00
                                                  (Default RV-E3J/RV-E3JM)
```

【 Sample program 】 (Movemaster command)

```
10 NT          ; Executes the origin setting.
20 MO  2       ; Moves to position 2.
30 OG          ; Moves to origin.
40 ED          ; Ends program.
```

OPN (Open)

【 Function 】

Opens communication channel and specify input/output device.

【 Input Format 】

```
        OPN  ‹channel number›, ‹device number›
```

【Term】

‹Channel number›   Specify input/output channel number.
$0 \leqq$ channel number $\leqq 2$

‹Device number›   Specify input/output device number.
1 : Standard RS-232C
2 : Standard RS-422

【 Explanation 】

(1) The corresponding relation of the channel number and input/output devices is specified, and the channel is opened.

(2) Up to three channels can be opened simultaneously. However, multiple ‹channel numbers› cannot be set in the same ‹input/output device number›.

(3) The counter, position and character string data can be read in with the INP command.

【 Sample program 】 (Movemaster command)

```
10 OPN   1,1          ; The standard RS-232-C is opened.
                            (Channel number 1 is assigned.)
20 INP   1,1,1        ; The position data is read from RS-232-C.
30 INP   1,1,0        ; The counter data is read from RS-232-C.
40 INP   1,$1,2       ; The character string data is read from RS-232-C.
```

OR (Or)

[Function]

ORs the specified data and the internal register data.

[Input Format]

```
        OR ‹operation data›
```

[Term]

‹Operation data › Specify the data to be operated.
-32768 ≦ operation data (decimal) ≦ 32767
& 8000 ≦ operation data (hexadecimal) ≦ &7FFF

[Explanation]

(1) Specify the data to be operated in decimal or hexadecimal. Any hexadecimal value must be headed by "&".

(2) The operation result is stored into the internal register and can be changed, compared or read by relevant commands.
(See the EQ, NE, LG, SM, CL, DR, AN, XO commands)

(3) Execution of the OR command after the input commands (ID) allows to be set to the required bits of the parallel input data fetched from the external device.

[Sample program] (Movemaster command)

```
10  ID               ; Fetches data from external input port.
20  OR  &FFF0        ; Sets 1 to all bits except lower order 4 bits.
30  EQ  &FFFF,100    ; If the above data are all bit 1, jumps to line 100.
40  ED               ; Ends program.
 :
100 MO  10           ; Moves to position 10.
```

OVR (Override)

【 Function 】

Specify program override.

【 Input Format 】

```
        OVR  ‹specified override›
```

【Term】

‹Specified override›   Specify override value. (%)
1 ≦ specified override ≦ 200

【 Explanation 】

(1) Specifies the ratio of working speed of the robot.

(2) The OVR command is effective for every interpolation mode, i.e. joint interpolation, linear interpolation and circular interpolation.

(3) The actual speed in the program eventually becomes the following.

Joint interpolation speed =
playback override × OVR command setting value × SP command setting value.

Linear interpolation speed =
playback override × OVR command setting value × SP or SD command setting value.

Here, the playback override can be specified by means of the starting display of teaching box or the external input signal. The override specified by the OVR command is called program override.

(4) The initial value of program override is 100 %.

(5) The override value once specified in the program is effective till new value is set or the program ends.

(6) Alarm occurs at the execution of the OVR command if the value 0 is set to the specified override.

(7) As the acceleration and deceleration distance required for movement are preset, when the specified speed and acceleration/deceleration are set, if the movement distance is small, the set speed may not be reached.

【 Sample program 】 (Movemaster command)

```
10 SP   30        ; Sets working speed 30  (100 %) .
20 OVR 80         ; Sets override 80 %.
30 MO  2          ; Moves to position 2.
40 ED             ; Ends program.
```

※ If the playback override is specified to 50 % in the above example, the actual override is as follows ;

Joint interpolation speed = 50 × 80 × 100 (%) = 40 (%)

The robot moves to position 2 with the speed of 40 % of maximum value.

PA (Pallet Assign)

【 Function 】

Defines the number of grid points in the column and row directions for the specified pallet.

【 Input Format 】

```
PA ‹pallet number›, ‹number of column grid points›, ‹number of row grid points›
```

【Term】

‹Pallet number›	Specify number of pallet using. 1 ≦ pallet number ≦ 9
‹Number of column grid points ›	Set grid points of column of pallet. 1 ≦ number of column grid points ≦ 32767
‹Number of row grid points›	Set grid points of row of pallet. 1 ≦ number of row grid points ≦ 32767

【 Explanation 】

(1) The PA command must be executed before the pallet calculation command (see the PT command) is executed.

(2) The number of grid points is equivalent to that of the actual workpieces arranged on the pallet. For example, with a pallet holding 15 workpieces (3 × 5) , the numbers of column and row grid points are 3 and 5, respectively.

(3) The column and row directions are decided by the directions of the terminating positions, respectively. (See the PT command)

【 Sample program 】 (Movemaster command)

```
10 PA  5,20,30   ; Defines the pallet 5 as the pallet holding 20 X 30 grid points.
20 SC  51,15     ; Sets value 15 to counter 51.  (column points)
30 SC  52,25     ; Sets value 25 to counter 52.  (row points)
40 PT  5         ; Sets the calculated coordinates value of grid point to position 5.
50 MO  5         ; Moves to position 5.  (The grid position)
60 ED            ; Ends program.
```

PC ※ (Position Clear)

【 Function 】
Clears the data of the specified position (s) .

【 Input Format 】

PC ‹position number (a) › [, [‹position number (b) ›]]

【Term】
‹Position number › Specify position number deleting.
1 ≦ position number (a) , (b) ≦ 999
Position number (a) ≦ position number (b)

【 Explanation 】
(1) Deletes all position data between positions (a) and (b) . (Position (b) included)
(2) If the position number (a) is greater than the position number (b) , alarm occur.

【 Sample program 】 (BASIC)

```
10 OPEN" 1:E83" AS #1  ;Opens the RS-232C communication file from the
                        personal computer in BASIC.
 20 PRINT #1," MO 10"   ;Moves to position 10.
 30 PRINT #1," MO 11"   ;Moves to position 11.
 40 PRINT #1," MO 12"   ;Moves to position 12.
 50 PRINT #1," PC 11"   ;Delete the position 11.
 60 PRINT 31," DP"      ;Moves to position 10.
 70 END                 ;Ends program.
```

PD (Position Define)

【 Function 】

Defines the coordinates (location and angle) of the specified position.

【 Input Format 】

(1) RV-E2/RV-E2M

```
PD ‹position number›, [‹ X coordinate›] , [‹ Y coordinate›] ,
[‹ Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
[‹C turning angle›]  [, [‹R/L›]  [, [‹A/B›]]  [, [‹N/F›]]]  [, [‹O/C›]]
```

(2) RV-E3J/RV-E3JM

```
PD ‹position number›, [‹ X coordinate›] , [‹ Y coordinate›] ,
[‹ Z coordinate›] , [‹A turning angle›] , [‹B turning angle›] ,
[, [‹R/L›]  [, [‹A/B›]]]  [, [‹O/C›]]
```

【Term】

‹Position number› Specify position number defining.
1 ≦ position number ≦ 999

‹X, Y, Z coordinate› Specify the location (mm) in XYZ coordinates of the robot. (0 for default).

‹A, B,C turning angle› Specify the turning angle (degree) of roll (A) and pitch (B) and twist (C) joints in XYZ coordinates of the robot. (0 for default)
(RV-E2/RV-E2M only.)

‹A, B turning angle› Specify the turning angle around roll (A) , pitch (B) axes in XYZ coordinates (degree) of the robot. (0 for default)
(RV-E3J/RV-E3JM only.)

‹R/L› Specify the structure flag of the robot. (Right or Left)
R : Right (default)
L : Left

‹A/B› Specify the structure flag of the robot. (Above or Below)
A : Above (Default)
B : Below

‹N/F› Specify the structure flag of the robot. (Nonflip or Flip)
(RV-E2/RV-E2M only.)
N : Nonflip (Default)
F : Flip

‹O/C› Specify open or close state of hand 1.
O : Hand 1 open
C : Hand 1 close

【 Explanation 】

(1) The least increment of the coordinate value is 0.01 mm or 0.01 degree (e.g. specify 20.01 for 20.01 mm) .

(2) Alarm does not occur even if the specified coordinates exceed the robot's operational space. The PD command, combined with the SF and the MA command, can define the amount for moving.

【 Sample program 】 (Movemaster command)

```
(1) RV-E2/RV-E2M
   10 PD   10,50,320,70,50,40,30,R,A,N,O     ; Defines the location and angle of position
                                              10.
   40 MO  10                                 ; Moves to position 10.
   50 ED                                     ; Ends program.
(1) RV-E3J/RV-E3JM
   10 PD   10,50,320,70,40,30,R,A,O          ; Defines the location and angle of position
                                              10.
   40 MO  10                                 ; Moves to position 10.
   50 ED                                     ; Ends program.
```

PL (Position Load)

【 Function 】
Replaces position (a) by position (b) .

【 Input Format 】

```
        PL ‹position number (a) ›, ‹position number (b) ›
```

【Term】
‹Position number (a) › Specify the position number. (Destination)
1 ≦ position number (a) ≦ 999
‹Position number (b) › Specify the position number. (Source)
1 ≦ position number (b) ≦ 999

【 Explanation 】
(1) After executed, the PL command causes the coordinates of position (a) to be equivalent to those of position (b) and the old coordinates of position (a) to be cleared.
(2) After executed, the PL command also assigns the hand state at position (b) to that at position (a) .
(3) Alarm occurs if position (b) is not defined.
(4) A new position is created if the position (a) is not defined.

【 Sample program 】 (Movemaster command)

```
10 HE  2        ; Sets the current coordinates and hand state to position 2.
20 PL  3,2      ; Replaces the coordinates of position 3 by position 2.
30 ED           ; Ends program.
```

PMR (Parameter Read)

【 Function 】
Reads contents of parameter specified.

【 Input Format 】

```
        PMR   ["‹parameter name›"]
```

【Term】

‹Parameter name ›   Specify parameter name.
Only parameter name defined is valid. (Defined order for default)

【 Explanation 】

(1) Outputs the specified parameter from RS-232C port.
The output format : Parameter name, contents

(2) When you omitted parameter name, a parameter defined next is output. (Parameters are defined in alphabetical order.) If the last parameter is read out, only carriage return (Hex. 0D) is output.

(3) When you specified the parameter that does not exist, contents of next parameter specifying in alphabetical order is output.

【 Sample program 】 (BASIC)

```
10 OPEN " COM1 : E83"  AS ♯1          ; Opens the RS-232C communication file from
                                      the personal computer in BASIC.
20 INPUT "Parameter name is" ; J$     ; Inputs the parameter name.
30 PRINT ♯1," PMR" +CHR$ (&H22) +J$+CHR$ (&H22)
                                      ; Transmits the PMR command followed by
                                      the parameter name.
40 LINE INPUT ♯1,A$                   ; Saves the received data to A $.
50 PRINT A$                           ; Displays the contents of A $.
60 END                                ; Ends program.

RUN                                   ; Run the program.
Parameter name is ?  ADL              ; Enter the parameter name.
ADL,0.20,0.20                         ; Outputs the contents of ADL
```

PMW (Parameter Writing)

[Function]

Renews the contents of the specified parameter.

[Input Format]

```
PMW "‹parameter name›", "‹parameter contents›"
```

[Term]

‹Parameter name›   Specify the parameter name changing contents.
‹Parameter contents› Specify the contents that you want to change to.

[Explanation]

(1) Outputs the contents of the specified parameter through the RS-232C port.
(2) When you specified the parameter that does not exist, no operation is executed.
(3) For information on parameters, see "6.2 Parameter List".
(4) The specified parameter becomes effective after having turned off and on the power supply. Give attention that even if you have changed the contents of the parameter, the old contents is still effective until the power supply is turned on again.

[Sample program] (Movemaster command)

```
PMW " ADL" ," 0.40,0.40"        ; Sets the contents of parameter ADL 0.40,0.40
```

PR (Position Read)

【 Function 】

Reads the coordinates of the specified position and the open/close state of the hand. (Using RS-232C)

【 Input Format 】

```
        PR  [‹position number›]
```

【TERM】

‹Position number› Specify the position number that you want to read.
0 ≦ position number ≦ 999 (If omitted, the current position number is valid.)

【 Explanation 】

(1) Outputs the coordinates of the specified position and the open/close state of the hand throgh the RS-232C port. If the position number is omitted or equals 0, only the current position number is output .

(2) The data is ASCII coded as follows ; The least increment is 0.01 mm or 0.01 degree.
Output format
RV-E2/RV-E2M : X, Y, Z coordinate value, A, B,C turning angle degree, R/L, A/B,N/F,O/C
RV-E3J/RV-E3JM : X, Y, Z coordinate value, A, B turning angle degree, R/L, A/B,O/C

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

(4) If you read the position that is not used in the program and not defined yet, "zero" is output with each coordinate.

RV-E2/RV-E2M	: 0,0,0,0,0,0
RV-E3J/RV-E3JM	: 0,0,0,0,0

If you read the position that is already used in the program but not defined yet, "0.00" is output with each coordinate.

RV-E2/RV-E2M	: 0.00,0.00,0.00,0.00,0.00,0.00
RV-E3J/RV-E3JM	: 0.00,0.00,0.00,0.00,0.00

(5) If you specify "zero" to the position number or omit it, you can identify the current position number when an alarm occurs in executing moving command.

【 Sample program 】 (BASIC)

(1) RV-E2/RV-E2M

```
10  OPEN " COM1 : E83"  AS #1        ; Opens the RS-232C communication file from
                                      the personal computer in BASIC.
20 INPUT  "Position number is" ; P   ; Inputs the position number that you want to
                                      read
30  PRINT #1," PR" +STR$ (P)         ; Transmits the PR command followed by the
                                             position number.
40 LINE INPUT #1,A$                  ; Saves the received data to A$.
50 PRINT A$                          ; Displays the contents to the screen.
60 END                               ; Ends program.

RUN                                  : Run the program.
Position number is ?  15             ; Enter the position number.
+10.00,+380.00,300.00,-70.00         ; Outputs the contents of the position.
,50.00,40.00,R,A,N,C
```

【 Sample program 】 (BASIC)

(2) RV-E3J/RV-E3JM

```
 10  OPEN " COM1 : E83"  AS #1          ;Opens the RS-232C communication file from
                                        the personal computer in BASIC.
 20 INPUT  "Position number is" ; P     ; Inputs the position number that you want to
                                        read
 30  PRINT #1," PR" +STR$ (P)           ; Transmits the PR command followed by the
                                        position number.
 40 LINE INPUT #1,A$                    ; Saves the received data to A$.
 50 PRINT A$                            ; Displays the contents to the screen.
 60 END                                 ; Ends program.

 RUN                                    ; Run the program.
 Position number is ?  15               ; Enter the position number.
 +10.00,+380.00,300.00,50.00            ; Outputs the contents of the position.
 ,40.00,R,A,C
```

PRN ※ (Print)

【 Function 】

The counter number setting value, position number coordinate value or the character string number data is transmitted from the personal computer in regard to the INP command. (Using RS-232-C)

【 Input Format 】

```
PRN  <counter value> | <position coordinates>| "<character string data>"
```

【Term】

<Counter value> Specify the counter value setting to a counter.
-32768 ≦ Counter value (decimal) ≦ 32767
& 8000 ≦ Counter value (hexadecimal) ≦ & 7FFF

<Position coordinates> Specify the coordinates value setting to a position. Specify the following coordinates similar to the PD command.
(See the PD command)
(1) RV-E2/RV-E2M
<X, Y, Z coordinates>, <A, B, C turning angle>, <R/L>, <A/B>, <N/F>, <O/C>
(2) RV-E3J/RV-E3JM
<X, Y, Z coordinates>, <A, B turning angle>, <R/L>, <A/B>, <O/C>

<Character string data> Specify the character string to be set.
1 ≦ Character string data ≦ 120 (number of characters)
Usable characters : numerals (0 to 9) , alphabetic characters (A to Z) , symbols (! @ ♯, etc.)
Unusable characters : *+,./ ; : [] ¥'

【 Explanation 】

(1) Transmits the setting value of counter, the coordinates value of position or the character strings from the personal computer through the RS-232C port responding to the INP command in the program.

(2) The robot becomes wait condition in the INP command till the date is transmitted from personal computer by executing the PRN command.

(3) You can execute the PRN command prior to the INP command during the execution of the program.

(4) When transmitting character string data, enclose the character string in double quotations (") .

【 Sample program 】 (BASIC)

<Personal computer program>

```
  10 OPEN " COM1 : E83"  AS ♯1  ; Opens the RS-232C communication file from the
                                  personal computer in BASIC.
  20 INPUT  "Counter data is" ; J ; Enter the setting value of counter from the
                                  personal computer.
  30 PRINT ♯1,"PRN"  +STR$ (J)  ; Transmits the setting value of counter.
  40 PRINT ♯1,"PRN 100,0,0,0,0,0"; Transmits the coordinates value of position.
  50 END                        ; Ends program.
```

<Robot program>

```
  10 OPN 2,1                    ; Opens the RS-232C port.
  20 IN 2,1,0                   ; Reads the data from RS-232C port to counter 1.
  30 INP 2,5,1                  ; Reads the data from RS-232C port to position 5.
  40 IC 1                       ; Add 1 to counter 1.
  50 MO 5                       ; Moves to position 5.
```

PT (Pallet)

[Function]

Calculates the coordinates of a grid point on the specified pallet and sets the coordinates value to the specified position.

[Input Format]

```
        PT ‹pallet number›
```

[Term]

‹Pallet number› Specify the number of pallet using.
1 ≦ pallet number ≦ 9

[Explanation]

(1) Calculates the coordinates of a grid point on the specified pallet and sets the coordinates to the position which number is corresponding to the specified pallet number. Before the PT command is executed, the pallet definition command (PA) must be executed for the pallet to be used. After the PT command has been executed, the position data previously defined for the target position is cleared.

(2) In order for the PT command to be executed, the pallet positions (grid points at four corners of the pallet) must be properly defined which identify a particular pallet and the pallet counters (column and row) be properly set that specify a particular grid point on the pallet. If the pallet positions and pallet counters are properly defined, therefore, execution of the PT command allows the coordinates of a grid point to be defined as the position number equivalent to the pallet number. The following is a listing of a combination of pallet positions and counters corresponding to each pallet number.

Pallet number	1	2	3	4	5	6	7	8	9
Pallet reference position	10	20	30	40	50	60	70	80	90
Pallet column terminating position	11	21	31	41	51	61	71	81	91
Pallet row terminating position	12	22	32	42	52	62	72	82	92
Pallet corner position opposite to reference	13	23	33	43	53	63	73	83	93
Pallet column counter	11	21	31	41	51	61	71	81	91
Pallet row counter	12	22	32	42	52	62	72	82	92
Pallet grid position	1	2	3	4	5	6	7	8	9

(3) Alarm occurs if the pallet position have not been defined and the pallet counters have not been set or have been set to have values exceeding those defined by the PA command. Alarm does not occur, however, even when the coordinates obtained for the grid point exceed the robot's operational space.

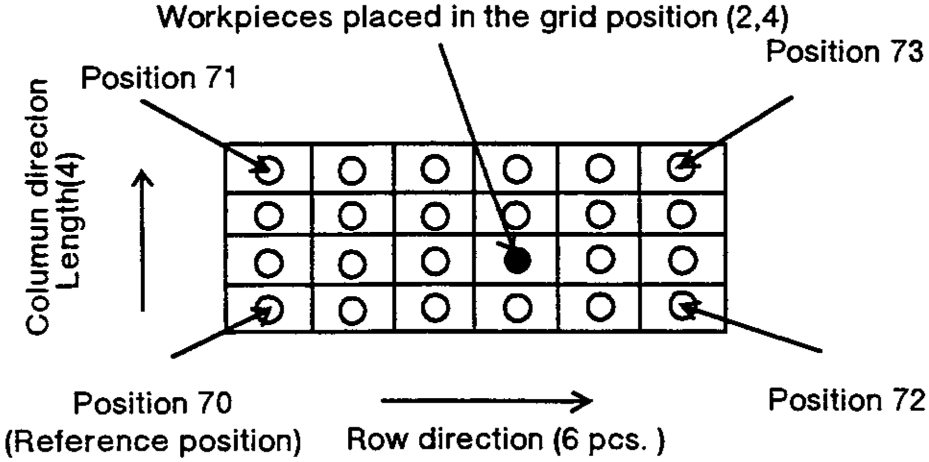
(4) The open or close state of the hand at the target grid point is the same as that in the pallet reference position.

(5) When executing the PT command, the tool length of the hand to be used must be properly defined by the TL command. The robot must be taught through the pallet positions (four corners) using the predefined correct tool length.

(6) Pay attention when you use pallet nine because the counters (91,92) of pallet nine are common counters among programs and other program may use these counters.

【 Sample program 】 (Movemaster command)

Suppose there is a pallet on which a total of 24 workpieces are arranged, 4 in the column direction and 6 in the row direction. Then have the system compute the coordinates of the workpiece placed in the grid position (2,4) , i.e. the second grid in the column direction and the fourth grid in the row direction, and get the robot hand to reach that position. Assume that pallet 7 is used.



※ You must teach the positions at four corners (70,71,72,73) in advance.

```
10 TL   200       ; Sets the tool length equivalent to the hand using.
20 PA   7,4,6     ; Sets the pallet number and the grid points of column and row.
30 SC   71,2      ; Sets the number of grid point in column direction.
40 SC   72,4      ; Sets the number of grid point in row direction.
50 PT   7         ; Allows the coordinates of the target grid point to be set to position
                    7.
60 MO   7         ; Moves to position 7.
70 ED             ; Ends program.
```

PW (Pulse Wait)

【 Function 】

Waits for in-position of servomotor about all joints till it becomes within the specified value.

【 Input Format 】

```
PW ‹positioning pulse›
```

【Term】

‹Positioning pulse ›   Specify the judgment pulse number of in-position.
                          1 ≦ positioning pulse ≦ 10000

【 Explanation 】

(1) Waits for in-position of servomotor about all joints till it becomes within the specified value.

(2) If you need the positioning of high accuracy when chucking a workpiece at a position, set small value for positioning pulse. If you need the positioning of low accuracy, set large value. If you set small value to the positioning pulse, the robot waits for positioning having the same effect as the TI command is executed.

(3) If you set small value to the positioning pulse when the handling workpiece is relatively heavy or the robot is moving at high speed, it may take a longer time to position than usual.

(4) Initial value of positioning pulse is 10000 pulse.

(5) When the setting value exceeds the above limit, alarm occurs.

【 Sample program 】 (Movemaster command)

```
10 MO  1        ; Moves to position 1.
20 PW  10       ; Waits for the positioning pulse becoming within 10 pulses.
30 GC           ; Closes hand.
40 ED           ; Ends program.
```

PX (Position Exchange)

[Function]

Exchanges the coordinates of the specified position for those of another specified position.

[Input Format]

```
PX ‹position number (a) ›, ‹position number (a) ›
```

[Term]

‹Position number›   Specify the position number exchanging.
1 ≦ position number (a) , (b) ≦ 999

[Explanation]

(1) After the PX command is executed, the coordinates of position (a) are exchanged for those of position (b)

(2) The open or close state of the hand at position (a) is also exchanged for that at position (b) .

(3) Alarm occurs if positions (a) and (b) have not been predefined.

[Sample program] (Movemaster command)

```
10 HE   2                ; Define the current position to position 2
20 MJ   20,30,10,0,0,0   ; Drives each joint by the specified amount.
30 GO                    ; Opens the hand.
40 HE   3                ; Define the current position to position 3.
50 PX   2,3              ; Exchanges the coordinates value of position 2 for those of
                         position 3.
60 ED                    ; Ends program.
```

QN (Question Number)

【 Function 】

Reads the program name or the program information.

【 Input Format 】

```
        QN    [‹program name›]
```

【Term】

‹Program name›   Specify the robot program name to be read. (Max. 8 characters)
Possible letter used : Digit (0-9)
Character (A - Z)
Symbol (! @# ,etc.)
Impossible letter used : ＊＋,. ／： ；＝？ [¥]' "
Special specification : When you specified only numeric value, the program name is handled for number.
Need to enclose program name with " " in the case of character used.

【 Explanation 】

(1) Outputs the selected program name or the selected program information through the RS-232C port. If you omitted the program name, the selected program name is turned over and if you specified it, the information about the program is turned over.

(2) The output format is ASCII coded as follows;
- The program number format : "N" followed by "Program name"
- The program information format : Used number of steps, used number of positions, used number of counters

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

【 Sample program 】 (BASIC)

```
 10 OPEN " COM1 : E83" AS #1             ; Opens the RS-232C communication file
                                          from the personal computer in BASIC.
  30 PRINT  #1," QN"                     ; Transmits the QN command.
  30 LINE   INPUT #1,A$                  ; Saves the received data to A$.
  40 PRINT  "Selected program is " : A $ ; Displays the contents of A$ to the screen.
  60 END                                 ; Ends.

  RUN                                    ; Run the program.
  Selected program is  N10               ; Outputs the program name.
```

RC (Repeat Cycle)

【 Function 】

Repeats the loop specified by the NX command the specified number of times.

【 Input Format 】

```
        RC ‹number of repeated cycles›
```

【Term】

‹Number of repeated cycles›   Specify the number of times repeating.
1 ≦ number of repeated cycles (decimal) ≦ 32767
& 0001 ≦ number of repeated cycles (hexadecimal) ≦ & 7FFF

【 Explanation 】

(1) Used with the NX command to cause the loop specified by the NX command to be executed the specified number of times and causes the line number following NX to be subsequently executed.

(2) To incorporate another loop (between RC and NX) into the existing loop is called "nesting". Up to 9 nesting levels are possible.

【 Sample program 】 (Movemaster command)

```
10 MO  1          ; Moves to position 1.
20 RC  3          ; Repeats loop delimited by NX three times
30 MO  2          ; Moves to position 2. ⎫
40 MO  3          ; Moves to position 3. ⎬   Loop.
50 MO  4          ; Moves to position 4. ⎭
60 NX             ; Delimits the loop.
70 MO  5          ; Moves to position 5.
80 ED             ; Ends program.
```

RN ※ (Run)

【 Function 】
Executes the specified part of commands in a program.

【 Input Format 】

```
RN   [‹start line number› [, ‹end line number› [, program name]]]
```

【Term】

‹Start line number› Specify the line number beginning.
1 ≦ start line number ≦ 9999 (The top line for default)

‹End line number› Specify the line number ending.
1 ≦ end line number ≦ 9999 (The last line or ED command line for default)

‹Program name› Specify the program name. (Max. 8 characters)
Possible letter used : Digit (0-9)
Character (A - Z)
Symbol (! @ ＃ ,etc)
Impossible letter used : ＊＋,. ／：；＝？ [¥]' "
Special specification : When you specified only numeric value, the program name is handled for number. Need to enclose program name with " " in the case of character used.

【 Explanation 】

(1) Runs the program starting with the specified starting line and ending with the line one ahead the specified ending line.
(2) If the program is to continue, restart with the ending line.
(3) If the teaching box is connected, the line number being executed is shown on its display.
(4) You can select the program by appointing the ‹program name›. In this case, the specified program becomes the target program and the program starts from the specified line.
(5) If you describe the RN command with line number in a program, no operation is executed.
(6) When the RN command is executed, the contents of the counter remain intact (not initialized) .
(7) The following name is identified as the same.
An example :
• Handled as the same. 1,01,001,00000001 (Only numeric value)
• Handled as not the same. 1,1 A, A0 _ 001 (Includes characters)
(8) The letters that controller can indicate to the LED are 0-9, A ~ Z (simplified) .

【 Sample program 】 (BASIC)

```
10 OPEN" COM1 : E83" AS ＃1      ; Opens the RS-232C communication file from
                                 the personal computer in BASIC.
20 PRINT ＃1," RN 100, ,2"       ; Executes the program 2 from line 100.
30 END                           ; Ends program.
```

RS ※ (Reset)

【 Function 】
Resets the program and error condition.

【 Input Format 】

```
        RS   [‹reset number›]
```

【Term】
‹Reset number›   Specify the contents of reset.
Reset contents
0 : Cancels alarm and resets program. (Default)
1 : Makes all counters undefined condition.
2 : Resets the battery timer.
3 : Deletes all programs and all positions.
(The same as the NW command.)
4 : Resets the origin setting condition.

【 Explanation 】
(1) Resets alarm condition in alarm mode, switching servo from OFF to ON and causing the program to return to its beginning.
(2) If any of the axes has exceeded its software limit, the alarm cannot be reset.
(3) The outputs remain unchanged by resetting any alarms.

【 Sample program 】 (BASIC)

```
10 OPEN " COM1 ;E83 AS #1   ;Opens the RS-232C communication file from
                            personal computer in BASIC.
 20 PRINT #1," MO 1000"     ;Alarm occurs because of the wrong value.
 30 PRINT #1," RS"          ;Cancels the alarm.
 40 END                     ;Ends program.
```

RT (Return)

【 Function 】
Completes a subroutine and returns to the main program.

【 Input Format 】

```
        RT  [‹line number›]
```

【Term】

‹Line number›   Specify the line number to jump.
1 ≦ line number ≦ 9999 (If omitted, returns to the next line of the GS command.)

【 Explanation 】

(1) Completes the subroutine called by the "GS" command and returns to the main program.
(2) Alarm occurs if the corresponding "GS" command is not specified.
(3) If you specify the line number, the program jumps to the specifiedline number after returning to the main routine.

【 Sample program 】
See the GS command.

SC (Set Counter)

[Function]

A specified value is set in the specified counter or character string.

[Input Format]

```
SC ‹counter number/character string number›, [‹counter set value
                                              /character string set value›]
```

[Term]

‹Counter number› Specify the number of counter setting.
1 ≦ counter number ≦ 99

‹Character string number› Specify character string number in numerical value which "$" is added to the head.
$1 ≦ character string number ≦ $99

‹Counter set value› Specify the value of counter setting. (0 for default)
-32768 ≦ set value (decimal) ≦ 32767
& 8000 ≦ set value (hexadecimal) ≦ & 7FFF

‹Character string set value› Specify the character string to be set.
Usable characters : numerals (0 to 9) , alphabetic characters (A to Z) , symbols (! @ #, etc.)
Unusable characters : "
Number of characters : Within 120 characters including line number and SC command.

[Explanation]

‹When counter number is specified›

(1) All counters are factory-set to zero.

(2) Used to count the number of workpieces and job sequences and to set the number of grid points in the pallet.

(3) The contents of the counter can be changed, compared or read by the relevant command. (Refer to the IC ,INP, DC , CP , CR , CL , AN , OR , XO commands.)

(4) The counter set value remains unchanged when the RS, NW or ED command is executed.

(5) The contents of the counter are battery backed after the power is switched off.

‹When a character string number is specified›

(1) Enclose the set character string with double quotations ("") .
Example) When setting the character string ABC, set "ABC".

(2) If the set character string is omitted, the details of the character string number will be blank. Thus, the details of the character string number can be deleted.

(3) Operation, comparison and reading of the character string are possible with the related commands. (Refer to the CP, CR, CL, EQ, NE, LG, SM, INP commands.)

(4) The value of the set character string will not change even if the RS, NW or ED command is executed. The value will be held by the battery even when the power is turned OFF.

[Sample program] (Movemaster command)

```
10 SC   21,10      ; Set value 10 to counter 21.
20 IC   21         ; Add 1 to counter 21.
30 CP   21         ; Set value of counter 21 to the internal register.
40 DR              ; Outputs the value of the internal register through RS-232C port.
50 SC   $5,"OK"    ; Set character string "OK" in character string number 5
60 CP   $5         ; Set details of character string number 5 in the character string
                     register
70 EQ   $10,200    ; Jumps to line 200 if the data equals character string number 10.
  :
100 ED             ; Ends program.
```

SD (Speed Define)

[Function]

Defines the moving velocity, first order time constant, acceleration/deceleration time, and continuous path setting.

[Input Format]

```
        SD   ‹moving speed›  [, ‹first order time constant›
             [, ‹acceleration time›, ‹deceleration time›  [, ‹CNT setting›]]]
```

[Term]

‹Moving speed› Set moving speed at linear or circular interpolation.
0.01 ≦ moving speed ≦ 650.00 (mm/sec)

‹First order time constant› 1 ≦ first order time constant ≦ 300 (millisecond)

‹Acceleration time› Set acceleration time to the maximum speed.
0 ≦ acceleration time ≦ 2000 (millisecond)

‹Deceleration time› Set deceleration time from the maximum speed.
0 ≦ deceleration time ≦ 2000 (millisecond)

‹CNT setting› Specify the enable or disable state of continuous path mode.
0 : Disable 1 : Enable

[Explanation]

(1) The least input increment of the moving speed is 0.01 mm/second or 0.01 degree/second (e.g., specify 20.05 for 20.05 mm/second) .
The least input increment of the first order time constant is 1 millisecond.

(2) Allows the moving speed (or angular speed) of the tip of hand for linear or circular interpolation to be defined in smaller increments than the SP command.

(3) Setting a large value to the first order time constant makes the robot operation slower and smoother.

(4) The speed set by the SD command is effective until a new value is set by the SD or SP command. If you omit the first order time constant and the acceleration/deceleration time, the predefined values continue to be effective.

(5) During linear or circular interpolation, a certain moving speed of the SD command may cause alarm in excess of the maximum speed of the corresponding joint. In this case, set the speed to a lower value.

(6) At power on, the moving speed is initialized to SP 12, H (63.3 mm/second) .

(7) By enabling the CNT setting, the robot moves continuously without acceleration and deceleration until the SD or SP command disables the CNT setting. (Path motion) However, the robot accelerates and decelerates at a starting and at a stopping point as well as when a timer or an input command is executed during the path motion.

(8) The acceleration time is the maximum time for the robot to reach to the maximum speed. Accordingly, when the moving speed does not reach to the maximum speed, the actual acceleration time becomes smaller than the specified value. (The situation is the same for the deceleration time.)

(9) As the acceleration and deceleration distance required for movement are preset, when the specified speed and acceleration/deceleration are set, if the movement distance is small, the set speed may not be reached.

(10) If the acceleration/deceleration time is less than 200msec, an overspeed or overload alarm may occur. Depending on the load conditions, the life of the mechanical parts may be shortened, so keep the time at 200msec or more when possible.

【 Sample program 】 (Movemaster command)

```
10 SP   15      ; Set the moving speed to 15.
20 MS   1       ; Moves to position 1 by linear interpolation.   (SP 15)
30 SD   100     ; Set the moving speed to 100mm/sec.
40 MS   2       ; Moves to position 2 by linear interpolation.   (100 mm/sec)
50 MO   3       ; Moves to position 3 by joint interpolation.    (SP 15)
60 MS   4       ; Moves to position 4 by linear interpolation.   (100 mm/sec)
70 ED           ; Ends program.
```

SF (Shift)

【 Function 】

Adds each coordinate value of position (b) to each coordinate value of position (a) and defines it again as a new position.

【 Input Format 】

```
        SF ‹position number (a) ›, ‹position number (b) ›
```

【Term】

‹Position number› Specify the position number.
1 ≦ position number (a) , (b) ≦ 999

【 Explanation 】

(1) The hand open or close state of position (a) , as well as the structural flag (R/L, A/B, N/F) , is not affected by the SF command.

(2) Alarm occurs if the position (a) and/or (b) have not been predefined.

(3) Does not effect any robot motion.

【 Sample program 】 (Movemaster command)

```
10 PD  20,0,0,20,0,0,0 ; Set the location and the attitude of position 20.
20 HE  10              ; Set the current position to position 10.
30 SF  10,20           ; The position 10 is shifted only Z-coordinate 20 mm of position
                         20.
40 MO  10              ; Moves to position 10.
50 ED                  ; Ends program.
```

SM (If Smaller)

[Function]

This compares the value of the internal register with a specified value. If smaller, the program will jump. The character string register and the numbers of characters in a specified character string are compared. If the character string register is smaller, the program will jump.

[Input Format]

```
SM  ‹compared value/character string number›, ‹branching line number›
```

[Term]

‹Compared value› Specify the value compared with the internal register.
-32768 ≦ Compared value (decimal) ≦ 32767
& 8000 ≦ Compared value (hexadecimal) ≦ & 7FFF

‹Character string number› Specify character string number in numerical value which "$" is added to the head.
$1 ≦ character string number ≦ $99

‹Branching line number› Specify the line number to which the program jumps.
1 ≦ branching line number ≦ 9999

[Explanation]

‹When counter number is specified›

(1) Causes a jump to occur conditionally in accordance with the external input data or the internal counter value.

(2) If the internal register value is smaller than the compared value (i.e., when the condition is met) , the program jumps to the specified line. Otherwise (i.e., when the condition is not met) , the program continues in sequence. Alarm occurs at a jump if the specified line number does not exist.

(3) A value can be loaded into the internal register by executing the input command (See ID) for the external input data or by executing the compare counter command (See CP) for the counter data. Accordingly when you carry out conditional branching, need to execute either of the above commands beforehand.

(4) The compared value may be defined either in decimal or hexadecimal. A hexadecimal value must be headed by "&".

‹When character string number is specified›

(1) The conditions will jump depending on the data input from an external source or the number of characters in a specified character string.

(2) If the number of characters in the character string register is smaller than the number of characters in a specified character string (when the conditions are established) , the program will jump to the specified line number. If the number is larger (when conditions are not established) , the next line will be executed. If the specified line number is not registered, an alarm will occur when jumping.

(3) By executing an INP command, the data input from an external device will be set in the character string register. The details of the character string number will be set by executing a CP command. Thus, when executing condition jumping, one of these commands must be executed first.

【 Sample program 】 (Movemaster command)

```
10  ID            ; Fetches the data from the external input port.
20  SM   10,100   ; If the input data is smaller than 10, jumps to line 100.
30  MS   1        ; Moves to  position 1 by linear interpolation.
40  ED            ; Ends program.
100 MO   10       ; Moves to position 10.
140 OPN 1,1       ; Opens the RS-232C port.
150 INP  1, ,2    ; Reads the data of character string register from the RS-232C
                    port.
160 SM   $5,200   ; Jumps to line 200 if the data smaller than character string number
                    10.
     :
200 MO 2          ; Moves to position 2.
```

SP (Speed)

[Function]

Sets the operating speed, acceleration or deceleration time and the continuous path setting.

[Input Format]

```
SP <speed level> [, <H/L> [, <CNT setting>]]
```

[Term]

<Speed level> Set moving speed.
$0 \leqq$ speed level $\leqq 30$

<H/L> Set acceleration/deceleration level.
H : High acceleration/deceleration (Max. 0.2 second)
L : Low acceleration/deceleration (Max. 0.4 second)

<CNT setting> Specify the enable/disable state of the continuous path mode.
0 : Disable 1 : Enable

[Explanation]

(1) Sets the operating speed in 31 steps and acceleration/deceleration time upon start and stop in 2 levels.

(2) The speed level is predetermined as a ratio to the maximum rpm of each joint for joint interpolation and as a ratio to the maximum speed of the tip of hand (650 mm/second) for linear interpolation.

(3) The acceleration/deceleration time may be selected from among H or L. The acceleration time is the maximum time for the robot to reach to the maximum speed. Therefore, when the moving speed does not reach to the maximum speed, the actual acceleration time becomes smaller than the specified value.

(4) The acceleration/deceleration distance required for movement is predetermined according to the specified speed and the set speed may not be reached if the distance of travel is small.

(5) For linear interpolation, the tip of hand, determined by the tool command, is moved at constant speed. In this case, alarm may result from any of the joints exceeding its maximum speed. If the motion of the position angle (RV-E2/RV-E2M : A,B,C angle, RV-E3J/RV-E3JM : A,B angle) is greater than the motion of the distance (X, Y, Z) , the robot moves in reference to the position angular speed.
The SD command allows the speed to be defined in smaller increments.

(6) Once set, the speed and acceleration/deceleration time remain valid until new values are set. The default values are SP 12, H. The most recent acceleration/deceleration time remains valid when it is not specified.

(7) By enabling the CNT setting, the robot moves continuously without acceleration and deceleration until the SD or SP command disables the CNT setting. (Path motion) However, the robot accelerates and decelerates at a starting and at a stopping point as well as when a timer or a input command is executed during the path motion.

[Sample program] (Movemaster command)

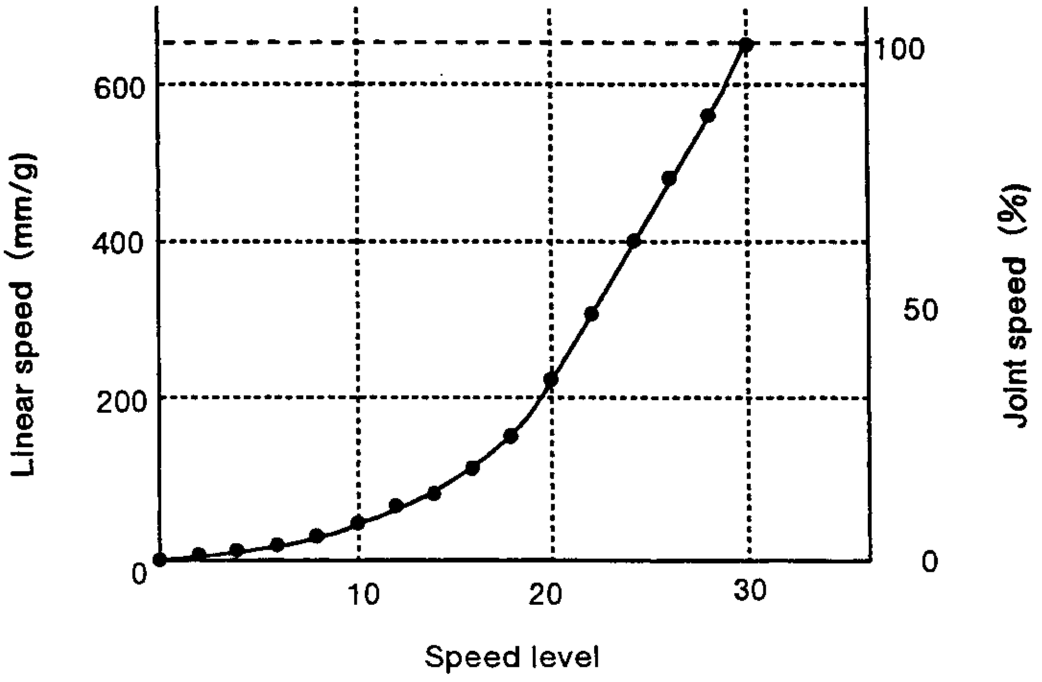
```
10 SP  8     ; Sets the moving speed to 8.
20 MO  5     ; Moves to position 5 by joint interpolation.
30 SP  10    ; Sets the moving speed to 10.
40 MS  7     ; Moves to position 7 by linear interpolation.
50 ED        ; Ends program.
```

The relation between the speed level and the moving speed.

Notice.) The robot moves in reference to the position angular speed in the case of the amount (RV-E2/RV-E2M : A, B, C turning angle, RV-E3J/RV-E3JM : A,B turning angle) ≧ the amount (X, Y, Z distance) in linear and circular interpolation. (The angular speed in degree/second is equivalent to the distance speed in mm/second divided by the value 2.12.)

▨ is the initial value when the power is turned on.

SP	Joint interpolation (%)	Linear interpolation (mm/second)	SP	Joint interpolation (%)	Linear interpolation (mm/second)
0	0.1	0.2	16	19.0	123.7
1	0.4	2.7	17	22.2	144.5
2	0.6	3.8	18	25.9	168.1
3	0.8	5.3	19	29.8	193.8
4	1.1	7.3	20	34.2	222.0
5	1.5	9.8	21	40.7	264.8
6	2.0	13.3	22	47.3	307.6
7	2.7	17.8	23	53.9	350.4
8	3.7	23.8	24	60.5	393.2
9	4.9	31.7	25	67.1	436.0
10	6.5	42.2	26	73.7	478.8
11	8.2	53.1	27	80.2	521.6
12	9.7	63.3	28	86.8	564.4
13	11.6	75.3	29	93.4	607.2
14	13.7	89.2	30	100.0	650.0
15	16.2	105.2			



STR ※ (Step Read)

【 Function 】

Reads the contents of the specified step number, or the stopping step number. (Using RS-232C)

【 Input Format 】

```
        STR   [‹step number›]
```

【Term】

‹Step number›   Specify the step number reading.
0 ≦ line number ≦ 9999

【 Explanation 】

(1) Outputs the contents of the specified step number, or the stopping step number through the RS-232C port.

(2) The output format is ASCII coded as follows;
‹In the case of teaching playback method›   : (colon) , program command
‹In the case of Movemaster command method› Line number, program command

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

(4) If you specify the undefined step number, the hexadecimal 0D is returned over.

(5) If the step number is not specified or zero is specified, the current stopping line number is read.

(6) In the above case, the command STR allows you to confirm the step number by a personal computer when alarm occurs

【 Sample program 】 (BASIC)

(1) RV-E2/RV-E2M

```
 10  OPEN " COM1 : E83"  AS #1      ; Opens the RS-232C communication file from
                                    the personal computer in BASIC.
 20 INPUT  "Reading step is " :  J $ ; Enters the step number that you want to read.
 30 PRINT #1," STR" +J$             ; Transmits "STR" + "step number" to the
                                    controller.
 40 LINE INPUT #1,A$                ; Saves the received data to A$.
 50 PRINT A$                        ; Displays the data on the screen.
 60 END                             ; Ends program.

 RUN                                ; Run the BASIC program.
 Reading step is  ? 2               ; Enters the step number.
: MPC 0,227.85,371.92,581.68,-60.71,102.83,30.85,R,A,N,C
                                    ; Outputs the contents of the step.
```

(2) RV-E3J/RV-E3JM

```
 10  OPEN " COM1 : E83"  AS #1      ; Opens the RS-232C communication file from
                                    the personal computer in BASIC.
 20 INPUT  "Reading step is " :  J $ ; Enters the step number that you want to read.
 30 PRINT #1," STR" +J$             ; Transmits "STR" + "step number" to the
                                    controller.
 40 LINE INPUT #1,A$                ; Saves the received data to A$.
 50 PRINT A$                        ; Displays the data on the screen.
 60 END                             ; Ends program.

 RUN                                ; Run the BASIC program.
 Reading step is  ? 2               ; Enters the step number.
: MPC 0,227.85,371.92,581.68,102.83,30.85,R,A,C
                                    ; Outputs the contents of the step.
```

TB (Test Bit)

[Function]

Causes a jump to occur in accordance with the specified bit value in the internal register.

[Input Format]

```
TB   [<+/->]  <bit number>, <branching line number>
```

[Term]

<+/-> Set the condition that compares bit.
+ : The bit is ON.
- : The bit is OFF.

<Bit number> Specify the bit number of the internal register.
0 ≦ bit number ≦ 15

<Branching line number> Specify the line number to which the program jump.
1 ≦ branching line number ≦ 9999

[Explanation]

(1) Causes a jump to occur conditionally in accordance with the external input data or the internal counter value.

(2) The program jumps to the specified line number if the specified bit in the internal register is on or off (i.e., when the condition is met) . Otherwise (i.e., when the condition is not met) , the program continues in sequence.

(3) A value can be loaded into the internal register by executing the input command (see ID) for the external input data or by executing the compare counter command (see CP) for the counter data. It is therefore necessary to execute either of the above commands beforehand so that a conditional jump can occur.

(4) Alarm occurs at a jump if the specified line number is not predefined

[Sample program] (Movemaster command)

```
10  ID              ; Fetches data from the external input port.
20  TB   +1,100     ; If the bit number 1 of input data is ON, then jumps to line number
                      100.
30  MS  1           ; Moves to position 1 by linear interpolation.
40  ED              ; Ends program.
 :
100 MO  10          ; Moves to position 10.
```

TBD (Test Bit Direct)

[Function]

Causes a jump to occur in accordance with the specified bit value in the external input.

[Input Format]

```
        TBD   [‹+/-›]  ‹input bit number›, ‹branching line number›
```

[Term]

‹+/-› Set the condition that compares bit.
+ : The bit is ON
- : The bit is OFF

‹Input bit number› Specify the bit number of general input.
0 ≦ input bit number ≦ 32767

‹Branching line number› Specify the line number to which the program jumps.
1 ≦ branching line number ≦ 9999

[Explanation]

(1) Causes a jump to occur conditionally in accordance with the external input data directly.

(2) The program jumps to the specified line number if the specified bit in the external input is on or off (i.e., when the condition is met) . Otherwise (i.e., when the condition is not met) , the program continues in sequence

(3) It is not necessary to execute the input command (ID) beforehand, and the internal register remains intact after the execution of TBD command.

(4) Alarm occurs if the specified line number is not predefined.

[Sample program] (Movemaster command)

```
10  TBD +19,100    ; If the bit 19 of the external input is ON, then jumps to line number
                     100.
20  MS  1          ; Moves to position 1 by linear interpolation.
30  ED             ; Ends program.
 :
100 MO  10         ; Moves to position 10.
```

TI (Timer)

【 Function 】
Halts the motion for the specified length of time.

【 Input Format 】

```
TI ‹timer counter›
```

【Term】

‹Timer counter› Set the period of timer.
0 ≦ timer counter ≦ 32767 (0.1 sec)

【 Explanation 】

(1) Causes the robot to halt its motion for the specified counter value × 0.1 second. (Max. 3276.7 seconds)
(2) Used to introduce a time delay before and after the hand is opened or closed for gripping a workpiece.
(3) The default value is zero.

【 Sample program 】 (Movemaster command)

```
10 MO  1,O        ; Moves to position 1.
20 TI  5          ; Wait for 0.5 second.
30 GC             ; Closes hand.
40 TI  10         ; Wait for 1.0 second.
50 MO  2          ; Moves to position 2.
60 ED             ; Ends program.
```

TL (Tool)

[Function]

Establishes the distance between the hand mounting surface and the tip of hand.

[Input Format]

```
        TL   [‹tool length›]
```

[Term]

‹Tool length›   Set the distance from the hand mounting surface to the tip of hand.
0 ≦ tool length ≦ 300.00 (mm)   (0 for default)

[Explanation]

(1) The least input increment of the tool length is 0.01 mm (e.g., specify 200.05 for 200.05 mm) .

(2) Once established, the tool length remains valid until a new value is set (battery backed when the power is switched off) . When the tool length has been changed, the current position is also changed accordingly, which, however, does not involve any robotic motion. (Initial tool length is 123 mm.)

(3) Since the point defined by the TL command is the basis for calculation of the current position, XYZ jogging and commands involving the XYZ coordinates system, the accurate tool length must be established according to the tool being used.

(4) Whenever a program is to be run, the same tool length as that established during teaching must be set at the beginning of the program.

[Sample program] (Movemaster command)

```
10 TL   120          ; Sets the tool length to 120 mm.
20 HE   1            ; Define the current position to position 1.
30 TL   100          ; Changes the tool length to 100 mm.
40 MO   1            ; Moves to position 1 advancing 20 mm in the tool direction.
50 ED                ; Ends program.
```

VR (Version Read)

【 Function 】
Reads the software version of the system ROM. (Using RS-232C)

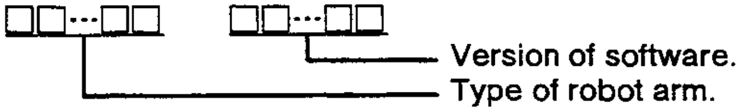
【 Input Format 】

```
VR
```

【 Explanation 】

(1) Outputs the software version of the system ROM mounted in the controller through the RS-232C port.

(2) The output format is ASCII coded. (Example : "RV-E2 Ver. A1")
The denotation of the software version is the following.



(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

【 Sample program 】 (BASIC)

```
10 OPEN" COM1:E83" AS #1              ;Opens the RS-232C communication file
                                      from the personal computer in BASIC.
20 PRINT #1," VR"                     ;Transmits the VR command.
30 LINE INPUT #1,A$                   ;Saves the received data to A$.
50 PRINT "Software version is " : A$; ;Displays the data on the screen.
60 END                                ;Ends program.

RUN                                   ;Run the program.
Software version is  RV-E2  Ver. A1   ;Outputs the version name.
```

WH (Where)

【 Function 】

Reads the coordinates of the current position and the open or close state of the hand. (Using RS-232C)

【 Input Format 】

```
        WH
```

【 Explanation 】

(1) Causes the coordinates of the current position of the tip of hand, as determined by the tool length (see the TL command) , and the hand open or close state to be output through the RS-232C port.

(2) The output format is ASCII coded as indicated below. The least output increment is 0.01 mm or 0.01 degree. (e.g., 20.01 for 20.01 mm)

Output format :

RV-E2/RV-E2M : X, Y, Z coordinate value, A, B, C turning angle, R/L, A/B, N/F, O/C (structural flag)

RV-E3J/RV-E3JM : X, Y, Z coordinate value, A, B turning angle, R/L, A/B, O/C (structural flag)

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

【 Sample program 】 (BASIC)

(1) RV-E2/RV-E2M

```
10 OPEN " COM1 :E83" AS #1           ; Opens the RS-232C communication file from
                                       the personal computer in BASIC.
20 PRINT #1," WH"                    ; Transmits the command "WH".
40 LINE INPUT #1,A$                  ; Saves the received data to A$.
50 PRINT "Current coordinates are " : A $   ; Displays the contents of A$ on the
                                       screen.
60 END                               ; Ends.

RUN                                  ; Run the program.
Current coordinates are +10.00,      ; Displays the value of the current position.
+380.00,300.00,-70.00,50.00,+40.00, R, A, N, C
```

(2) RV-E3J/RV-E3JM

```
10 OPEN " COM1 :E83" AS #1           ; Opens the RS-232C communication file from
                                       the personal computer in BASIC.
20 PRINT #1," WH"                    ; Transmits the command "WH".
40 LINE INPUT #1,A$                  ; Saves the received data to A$.
50 PRINT "Current coordinates are " : A $   ; Displays the contents of A$ on the
                                       screen.
60 END                               ; Ends.

RUN                                  ; Run the program.
Current coordinates are +10.00,      ; Displays the value of the current position.
+380.00,300.00,+50.00,+40.00, R, A, C
```

WT (What Tool)

[Function]

Reads the tool length currently being established. (Using RS-232C)

[Input Format]

```
      WT
```

[Explanation]

(1) Causes the tool length currently being established (by the TL command) to be output through the RS-232C port.

(2) The data is output in ASCII coded decimal. The least output increment is 0.01 mm (e.g., 105.07 is displayed for 105.07 mm) .

(3) Because the terminator of the output data is carriage return (Hex.0D) , it is necessary to handle serial data strings up to hexadecimal 0D in receiving a message by a personal computer. "LINE INPUT #" statement is equivalent to this in BASIC.

(4) All robotic motions are based on the established tool length. If a wrong tool length has been defined, the robot may interfere with a surrounding object. When the tool length is unknown, therefore, check the tool length using the WT command before starting the robot.

[Sample program] (BASIC)

```
10 OPEN " COM1 : E83"  AS #1     ; Opens the RS-232C communication file from
                                   the personal computer in BASIC.
 20 PRINT #1," WT"                ; Transmits the command "WT".
 40 LINE INPUT #1,A$              ; Saves the received data to A $.
 50 PRINT " TOOL="  ; A$          ; Displays the contents of A$ on the screen.
 60 END                           ; Ends.

 RUN                              ; Run the program.
 TOOL=105.7                       ; Outputs the tool length.
```

XO (Exclusive Or)

【 Function 】
EXCLUSIVE ORs the specified data and the internal register data.

【 Input Format 】

```
        XO ‹operation data›
```

【Term】
‹Operation data› Specify the data to be operated.
-32768 ≦ operation data (decimal) ≦ 32767
& 8000 ≦ operation data (hexadecimal) ≦ & 7FFF

【 Explanation 】

(1) Specify the data to be operated in decimal or hexadecimal. Any hexadecimal value must be headed by "&".

(2) The operation result is stored into the internal register and can be changed, compared or read by relevant commands. (See the EQ, NE, LG, SM, CL, DR, AN, OR commands)

(3) Execution of the XO command after the input command (ID) allows the required bits of the parallel input data fetched from the external device to be flipped to their opposite settings.

【 Sample program 】 (Movemaster command)

```
10  ID               ; Fetches data from the external input port.
20  AN   &000F       ; Fetches only lower 4 bits.
30  XO   &000F       ; Flips data of 4 lower bits to their opposite settings.
40  CL   21          ; Sets above data to counter 21.
50  EQ   10,200      ; If the above data equals 10, then jumps to line 200.
60  ED               ; Ends program.
 :
200 MO   99          ; Moves to position 99.
```

' (Comment)

【 Function 】

Allows the programmer to write a comment.

【 Input Format 】

' [‹string consisting of up to 120 alphanumeric characters
including line number and ' (apostrophe) ›]

【 Explanation 】

(1) You can describe up to 120 alphanumeric characters including line number and ' (apostrophe) .

(2) Use it to describe the name and date on the generated program or to mark a subroutine. Comments are helpful to check back the program as the LR (Line Read) command is used.

(3) The system ignores comments as it processes its commands.

(4) If the number of characters exceeds the limit, the whole excess is ignored.

【 Sample program 】 (Movemaster command)

```
10´ ****************
20´ Sample Program
30´ Date : 93-10-01                  ; Specify the contents of program,
40´ Programmed by Mitsubishi          the date of implementation, and
50´ ****************                  thename of programmer, etc.
60 NT
```

Chapter 4 : Design and Engineering

This chapter is a collection of useful technical information on the system design that will help you when you set up the robot. Also, for related information that is described separately, refer to the index (below) .

4.1 External connections to controller

4.1.1 Specifications for external connections to controller

In this section we'll provide explanations and guidance related to the various equipment connections to the controller. For details, about connections, refer to the Specifications manual or the instruction manuals (User's Manual and Reference Manual) . The topics can be found as follows :

(1) Machine cable connections ....Instruction manual (User's Manual)
Chapter 3 section 3.4

(2) Power cable connections ....Instruction manual (User's Manual)
Chapter 3 section 3.5

(3) Ground ....Instruction manual (User's Manual)
Chapter 3 section 3.6

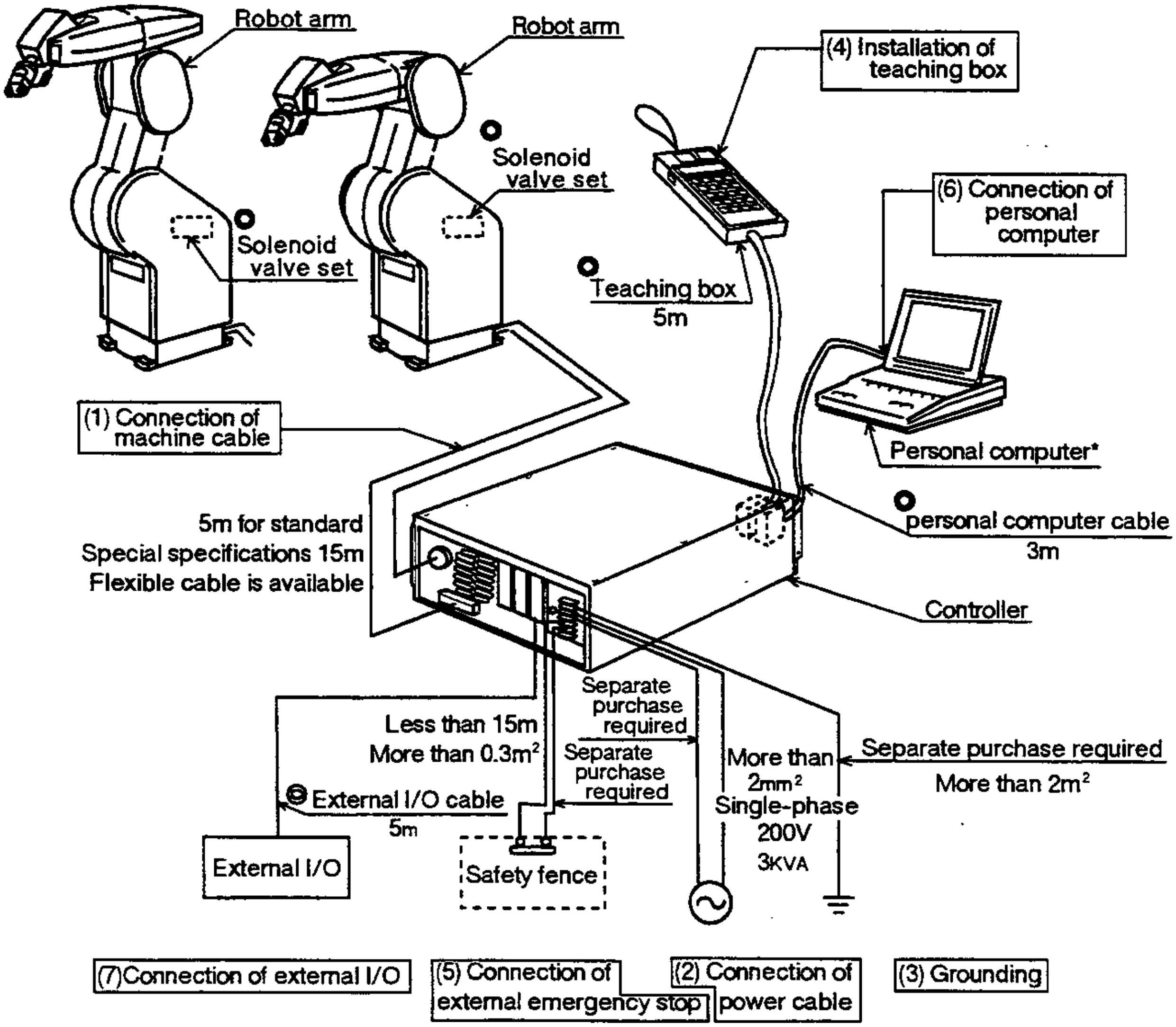
(4) Installing the teaching box ....Instruction manual (User's Manual)
Chapter 3 section 3.7

(5) External emergency stop connections .....Specifications Manual
Chapter 6 section 6.1

(6) Personal computer connections ....Instruction manual (Reference Manual)
Chapter 4 section 4.3

(7) External I/O connections ....Instruction manual (Reference Manual)
Chapter 4 section 4.1

The numbers (1) to (8) in the graph correspond to the numbers in the previous pages.


Robot arm
Robot arm
◎ Solenoid valve set
◎ Solenoid valve set
(4) Installation of teaching box
(6) Connection of personal computer
◎ Teaching box 5m
Personal computer*
◎ personal computer cable 3m
Controller
(1) Connection of machine cable
5m for standard
Special specifications 15m
Flexible cable is available
Less than 15m
More than 0.3m²
Separate purchase required
Separate purchase required
More than 2mm²
Single-phase 200V
3KVA
Separate purchase required
More than 2m²
◎ External I/O cable 5m
External I/O
Safety fence
(7)Connection of external I/O
(5) Connection of external emergency stop
(2) Connection of power cable
(3) Grounding

Note1: Two concentric circles (◎) mean it's optional equipment, an asterisk (*) means that separate purchase is necessary.

Fig. 4.1 External connections to controller

4.1.2 Interfacing with external I/O equipment

(1) Connection method

The connections between the controller and external equipment are done by connecting the connector for I/O interface card on the back of the controller to the external I/O device with the optional external I/O cable. One parallel I/O interface card is standard equipment for the controller, but by adding more, you can have up to 3. You need to supply the input/output cables and power source (DC12 to 24V)

(2) The number of inputs/outputs for the parallel I/O interface

Each interface card has a total of 20 inputs and 16 outputs as fixed points, and it is possible to allot general-purpose points as well as specific purpose points by setting the parameters for the various bits.

It is also possible to select functions for specialized I/O from the following tables 4.5 and 4.6. so that you can have the best I/O allotment for the given use.

Here, shown below, are the factory settings for the standard card (the first one) and the optional expansion cards (2nd and 3rd cards) .

Table 4.1 Specifications for the standard parallel I/O interface

Type	General-purpose	Specialized	Notes
Input	17 points(It contains setting value of input 4+4)	3 points(STA,STP,RST)	Total 20 points (Start, Stop, Reset)
Output	13 points	3 points(RUN,WAI,ERR)	Total 16 points (Executing,Waiting,Alarm)

Table 4.2 Expansion card NO. 1 parallel I/O interface specifications

Type	General-purpose	Specialized	Notes
Input	20 points	0 points	Total 20 points (All are for general-purpose input)
Output	16 points	0 points	Total 16 points(All are for general-purpose output)

Table 4.3 Expansion card NO. 2 parallel I/O interface specifications

Type	General-purpose	Specialized	Notes
Input	20 points	0 points	Total 20 points (All are for general-purpose input)
Output	16 points	0 points	Total 16 points(All are for general-purpose output)

4.1.3 Assignment of specialized I/O signals for parallel I/O interfaces

(1) Factory set commands

The specialized inputs and the specialized outputs can be used by writing the commands in tables 4.5 and 4.6 in bit format to the parameters (Input : IN1 to IN3 and output : OT1 to OT3) . For information on how to read and write parameters, refer to 2.6 "Other functions" under " (11) Parameter settings" in this manual, and then do so.

Shown below are the setting commands at the time of factory shipment for the various cards.

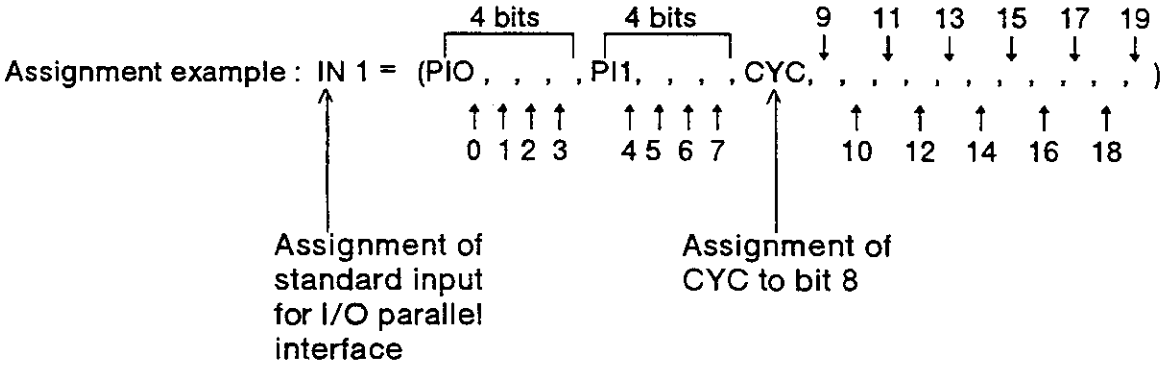
Table 4.4 Setting commands for the parallel I/O interface

	I/O card	I/O	Parameter	Command at time of factory shipment
				0 4 8 12 16 19 bits
	Standard	Input	IN1	PIO, , , ,PI1, , , , , , , , , , , , ,STA ,STP,RST
		Output	OT1	, , , , , , , , , , , , ,RUN,WAI,ERR
Optional	Expansion 1st board	Input	IN2	, , , , , , , , , , , , , , , , , , ,
		Output	OT2	, , , , , , , , , , , , , , , ,
	Expansion 2st board	Input	IN3	, , , , , , , , , , , , , , , , , , ,
		Output	OT3	, , , , , , , , , , , , , , , ,

(2) Assignment of the I/O value

The I/O value commands (Input : PI0 to PI3, output PO0 to PO3) communicates the program's numerical data in binary format, and each command uses 4 continuous bits. (Example : 1 command can input/output 0 to 15 numerical values, 2 commands 0 to 255.)

Accordingly, put 4 commas (,) after the numerical I/O command. If you omit the commas, the object bits pile up and the robot cannot properly handle input/output.


4 bits
4 bits
9 11 13 15 17 19
Assignment example : IN 1 = (PIO , , , , PI1, , , , CYC, , , , , , , , , , , ,)
0 1 2 3
4 5 6 7
10 12 14 16 18
Assignment of standard input for I/O parallel interface
Assignment of CYC to bit 8

⚠ Caution A signal that has been assigned as a specialized input can be used as a general-purpose input when the program is in progress, but for safety reasons, don't mix non-numerical inputs with general purpose inputs. If you try to do so, an alarm will result.

4.1.4 Commands and functions that can be written to parameters

In the table shown below, you'll find an explanation of commands, along with their functions, that can be written to the I/O setting parameters (IN1 to IN3 and OT1 to OT3) of the parallel I/O. To find the read and change operation, refer to section 2.6 "Other Functions (11) Setting the parameters" in this Manual.

Table 4.5 Input command names and functions

Name	Command	Function	Level
Start	Note 3: STA	Executes a program and starts the robot.	PE
Stop	Note 2: STP	Interrupts a program and stops the robot	HL
Program reset	Note 3: RST	Resets the interrupted status. The alarm status is also reset at the same time. If the servo power is OFF, it is turned ON.	PE
Alarm reset	Note 3: ERS	Resets alarms, but does not reset the program. If the servo power is OFF, it does not turn it ON.	PE
Servo ON/OFF	Note 3: SVO	Turns servo ON/OFF. When the signal is ON, the servo can be switched ON/OFF. (The brake works synchronously.)	PE
Brake ON/OFF	Note 3: BRK	Controls brakes when servo is OFF. The brakes are disengaged when the signal is ON.(Be careful of the falling arm.)	HL
Continuous/cycle	Note 3: CYC	Changes the continuous/cycle operation mode. It toggles when the signal is ON.	PE
Manual control mode	TMD	Controls the maximum speed during program execution to set speed of the parameters JGJ, JGP for the designated jog.	PE
Enable requirement	Note 3: ORQ	Toggles between enable/disable for the control of external signal control. It toggles when the signal is ON. When enabled start up and resetting operations will not be possible.	PE
General output reset	Note 3: ORS	Turns OFF all general-purpose output signals.	PE
Program number designation	Note 3: PGN	Reads data set at the numerical input signal as the program number.	PE
Program number output requirement	PGR	Outputs a program number of a program in progress from the numerical output signal.	PE
Line number designation	Note 3: LLN	Reads data set at the numerical input signal as the line number for starting the program. (Starting in the middle of a program is possible.)	PE
Line number output requirement	LLR	Outputs the line number currently being executed from the numerical value output signal.	PE
Override designation	OVR	Reads data set at the numerical value input signal as the override speed for operating the program.	PE
Override output requirement	ORR	Outputs override speed of program in progress from the numerical value output signal.	PE
Numerical value input signal (0) (1) (2) (3)	PI0 PI1 PI2 PI3	Used for inputting numerical data for various purposes, including designating a program number, designating line number, and designating an override. (Each are 4 bits)	HL HL HL HL

Note 1 : The definition of "level" is as shown below: If a signal has not been connected, its status will be OFF.

HL (High active level) When an external signal is ON, the designated function will be enabled.
When the external signal is OFF, it will be disabled.

PE (Positive edge) When an external signal changes from OFF to ON, the designated function becomes enabled. After that, even if the external signal is turned OFF, the designated function will remain in its original enabled state. A pulse rate of 0.1 seconds or longer is required.

Note 2 : At time of factory shipment, when the external signal is ON, it has an A connection function that maintains the stopped status, but, by changing the value of the parameter INB from 0 to 1, it is possible to change it to a B connection function.

Note 3 : Depending on the particular status of the robot, such as whether the robot is operating, or whether a stop signal has been entered, these signals may not function even if the signal in question is input.
Below you'll find robot conditions for when signals will not function and their names.

Signal name	(Command)	Enable/disable for the signal mentioned on the left that depends on the robot's condition
Program reset	(RST)	Will not function when in operation (run condition)
Brake ON/OFF	(BRK)	
General-purpose output reset	(ORS)	
Program number designation	(PGN)	
Line number designation	(LLN)	
Start	(STA)	Functions only when the external I/O device is enabled (in ATV condition) However, it will always be enabled when the servo is OFF based on the servo ON/OFF (SVO).
Program reset	(RST)	
Alarm reset	(ERS)	
Servo ON/OFF	(SVO)	
Brake ON/OFF	(BRK)	
Continuous/cycle	(CYC)	
General-purpose output reset	(ORS)	
Program number designation	(PGN)	
Line number designation	(LLN)	
Override designation	(OVR)	
Start	(STA)	Will not function when a stop (STP) input signal has been received. However, it will always be enabled when the servo is OFF based on the servo ON/OFF (SVO).
Servo ON/OFF	(SVO)	

Table 4.6 Output command names and functions

Name	Command	Function	Level
Executing	RUN	Outputs the fact that the program is executing	HL
Waiting	WAI	Outputs the fact that the program has temporarily stopped.	HL
Sounding alarm	ERR	Outputs the fact that an alarm has not been generated.	HL
Device enable	ATV	Outputs the fact that an external I/O wields operating control. Corresponds to the input command ORQ.	HL
Servo ON	SVA	Outputs the fact that the Servo is ON. Corresponds to the input command SVO.	HL
Continuous cycle status	CYS	Outputs the fact that the program is in the continuous mode. Corresponds to the input command CYC.	HL
Manual mode status	TMS	Outputs fact that the maximum speed during operation will be restricted in the manual mode. Corresponds to the input command TMD.	HL
Within user-defined area	UAR	Outputs the fact that the current position of the robot is within the user-defined area based on the parameter UAR.	HL
Ready	RDY	Outputs the fact that external input signals can be received when the controller power is ON. (It is output until the power is turned OFF)	HL
Numerical output signal (0)	PO0	Used for outputting numerical data for various purposes, including designating a program number, designating line number, and designating an override. (Each are 4 bits) (All bits have not been assigned at time of factory shipment.)	HL
Numerical output signal (1)	PO1		HL
Numerical output signal (2)	PO2		HL
Numerical output signal (3)	PO3		HL

Note 1 : The definition of ”level” is as follows :

HL (High active level) When a designated external condition exists, the output signals will be ON
Once the signals have been output, until the designated condition changes, their status will remain unchanged. .

4.1.5 Connection table for parallel I/O interface

(1) Standard parallel I/O interface

The table shown below contains assignment of the connector pin numbers of the standard card and their various functions. Here is shown the distinction of the signal lines when you connect the optional external I/O cables. For details on the connector pin array and the external I/O cable, refer to section 4.2 "Option (22) , (23) " in the Specifications Manual.

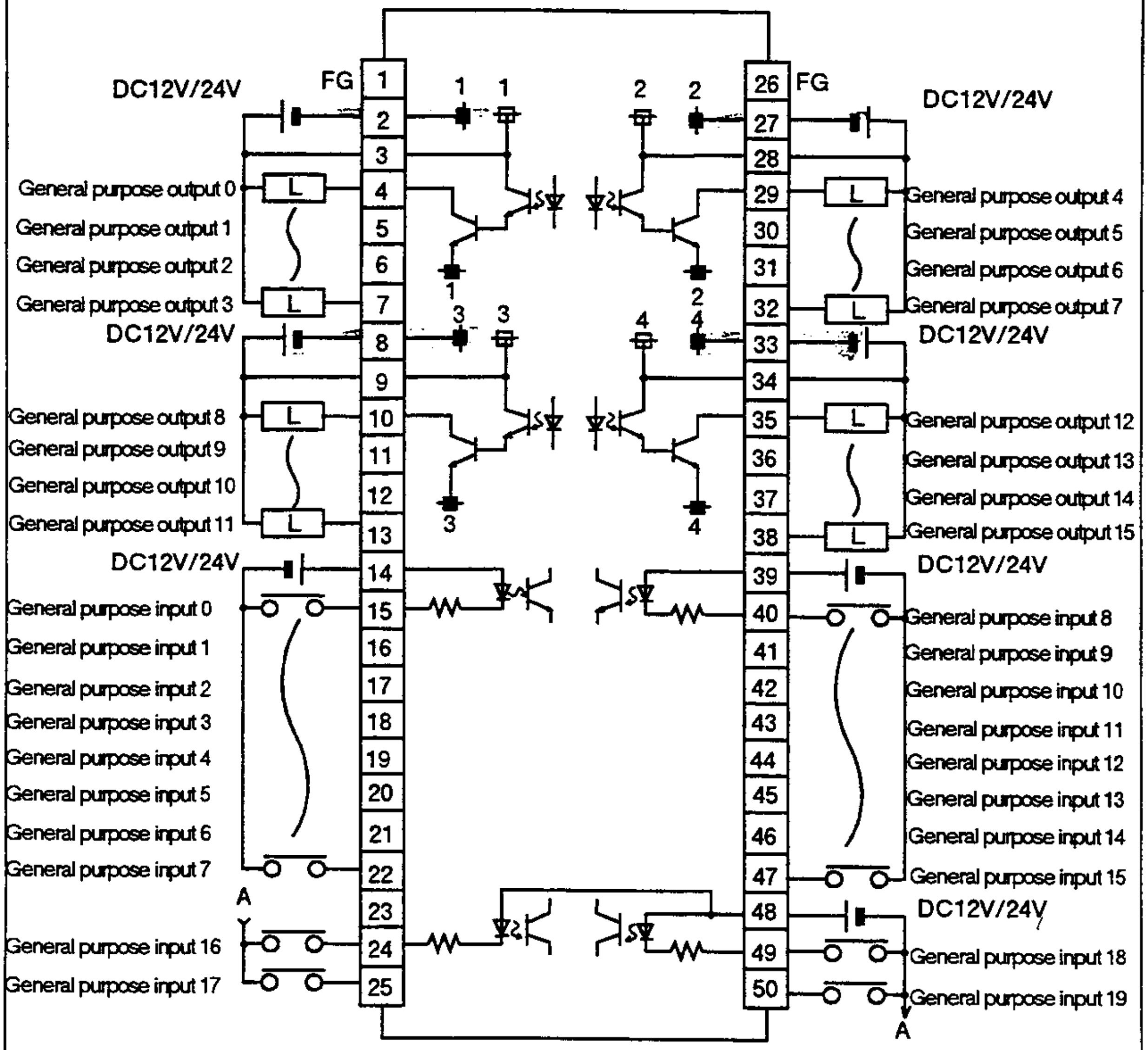
Table 4.7 Directory of assignments for pin numbers and signals for the standard parallel I/O interface

Pin number	Color of external I/O cable	Function name		Pin number	Color of external I/O cable	Function name	
		General purpose	Specialized/ power source/ common			General purpose	Specialized/ power source/ common
1	White/black A		FG	26	White/black B		FG
2	Yellow/black A		OV :4-7 pin use	27	Yellow/black B		OV : 29-32 pin use
3	Blue/black A		12V/24V : 4-7 pin use	28	Blue/black B		12V/24V : 29-32 pin use
4	Green/black A	General-purpose output 100		29	Green/black B	General-purpose output 104	
5	Orange/black A	General-purpose output 101		30	Orange/black A	General-purpose output 105	
6	Pink/black A	General-purpose output 102		31	Pink/black B	General-purpose output 106	
7	Gray/black A	General-purpose output 103		32	Gray/black B	General-purpose output 107	
8	Red/black A		OV :10-13 pin use	33	Red/black B		OV : 35-38 pin use
9	Purple/black A		12V/24V : 10-13 pin use	34	Purple/black B		12V/24V: 35-38 pin use
10	Brown/black A	General-purpose output 108		35	Brown/black B	General-purpose output 112	
11	White/black C	General-purpose output 109		36	White/black D	General-purpose output 113	RUN (RUN)
12	Yellow/black C	General-purpose output 110		37	Yellow/black D	General-purpose output 114	WAIT (WAI)
13	Blue/black C	General-purpose output 111		38	Blue/black D	General-purpose output 115	ERROR (ERR)
14	Green/black C		COM0 : 15-22 pin use	39	Green/black D		COM1 : 40-47 pin use
15	Orange/black C	General-purpose output 100		40	Orange/black D	General-purpose output 108	
16	Pink/black C	General-purpose output 101		41	Pink/black D	General-purpose output 109	
17	Gray/black C	General-purpose output 102		42	Gray/black D	General-purpose output 110	
18	Red/black C	General-purpose output 103		43	Red/black D	General-purpose output 111	
19	Purple/black C	General-purpose output 104		44	Purple/black D	General-purpose output 112	
20	Brown/black C	General-purpose output 105		45	Brown/black D	General-purpose output 113	
21	White/red A	General-purpose output 106		46	White/red B	General-purpose output 114	
22	Yellow/red A	General-purpose output 107		47	Yellow/red B	General-purpose output 115	
23	Blue/red A		N.C.	48	Blue/red B		COM2:24,25,49,50 use
24	Green/red A	General-purpose output 116		49	Green/red B	General-purpose output 118	STOP (STP)
25	Orange/red A	General-purpose output 117	STARTING (STA)	50	Orange/red B	General-purpose output 119	RESET (RST)

Do to a misprint tables 4.7, 4.8, & 4.9 at Pins 15 thru 25 and pins 40 thru 50 are marked output & should be INPUTS

= 0 V (GROUND)
= +12/24VDC 80mA

The assignment of the connections to the parallel I/O interface board is shown in Fig. 4.2. (2E-31IO)

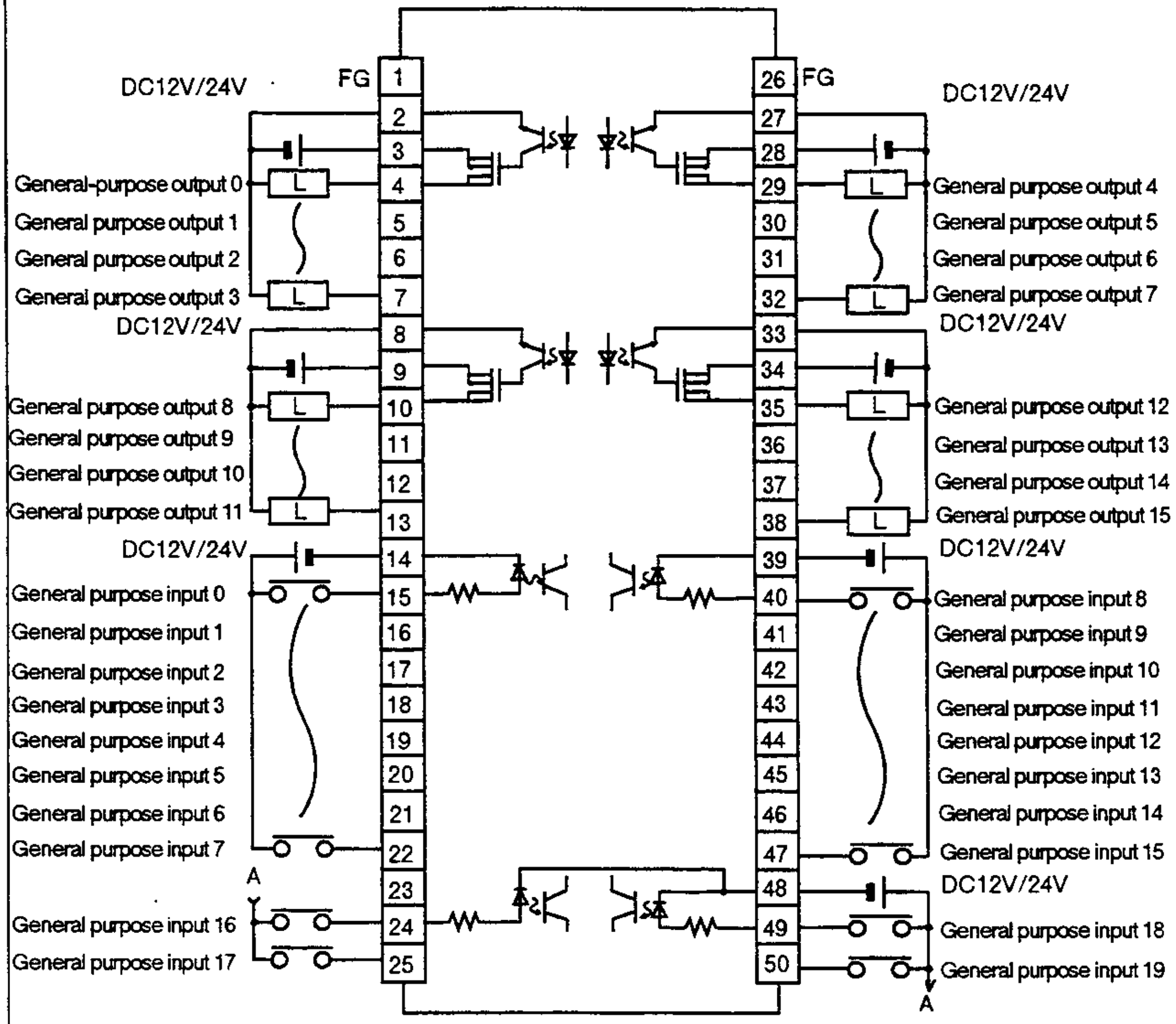


Note 1 : You need to supply your own power source DC12V/24V.

	‹Standard›		‹Expansion›
• The I/O number for the first addition is 100 added to the standard I/O number.	1	→	101
• The I/O number for the second addition is 200 added to the standard I/O number.	1	→	201.

Fig. 4.2 Connection scheme for parallel I/O interface board (U.S.A.,Japan)

The assignment of the connections to the parallel I/O interface board is shown in Fig. 4.2E. (2E-31IOE)



Note 1 : You need to supply your own power source DC12V/24V.

	‹Standard›		‹Expansion›
• The I/O number for the first addition is 100 added to the standard I/O number.	1	→	101
• The I/O number for the second addition is 200 added to the standard I/O number.	1	→	201.

Fig. 4.2E Connection scheme for parallel I/O interface board (EUROPE)

(2) The first additional parallel I/O interface

Shown below is the pin assignment for the connector pin numbers of the first additional card and their functions.

Here is shown the distinction of the signal lines when you connect the optional external I/O cables. For details on the connector pin array and the external I/O cable, refer to section 4.2 "Option (22) , (23) " in the Specifications Manual.

Table 4.8 Directory of assignments for pin numbers and signals for the first additional parallel I/O interface

Pin num ber	Color of external I/O cable	Function name		Pin num ber	Color of external I/O cable	Function name	
		General purpose	Specialized/ power source/ common			General purpose	Specialized/ power source/ common
1	White/black A		FG	26	White/black B		FG
2	Yellow/black A		OV : 4-7 pin use	27	Yellow/black B		OV : 29-32 pin use
3	Blue/black A		12V/24V : 4-7 pin use	28	Blue/black B		12V/24V : 29-32 pin use
4	Green/black A	General-purpose output 100		29	Green/black B	General-purpose output 104	
5	Orange/black A	General-purpose output 101		30	Orange/black A	General-purpose output 105	
6	Pink/black A	General-purpose output 102		31	Pink/black B	General-purpose output 106	
7	Gray/black A	General-purpose output 103		32	Gray/black B	General-purpose output 107	
8	Red/black A		OV : 10-13 pin use	33	Red/black B		OV : 35-38 pin use
9	Purple/black A		12V/24V : 10-13 pin use	34	Purple/black B		12V/24V: 35-38 pin use
10	Brown/black A	General-purpose output 108		35	Brown/black B	General-purpose output 112	
11	White/black C	General-purpose output 109		36	White/black D	General-purpose output 113	
12	Yellow/black C	General-purpose output 110		37	Yellow/black D	General-purpose output 114	
13	Blue/black C	General-purpose output 111		38	Blue/black D	General-purpose output 115	
14	Green/black C		COM0 : 15-22 pin use	39	Green/black D		COM1 : 40-47 pin use
15	Orange/black C	General-purpose output 100		40	Orange/black D	General-purpose output 108	
16	Pink/black C	General-purpose output 101		41	Pink/black D	General-purpose output 109	
17	Gray/black C	General-purpose output 102		42	Gray/black D	General-purpose output 110	
18	Red/black C	General-purpose output 103		43	Red/black D	General-purpose output 111	
19	Purple/black C	General-purpose output 104		44	Purple/black D	General-purpose output 112	
20	Brown/black C	General-purpose output 105		45	Brown/black D	General-purpose output 113	
21	White/red A	General-purpose output 106		46	White/red B	General-purpose output 114	
22	Yellow/red A	General-purpose output 107		47	Yellow/red B	General-purpose output 115	
23	Blue/red A		N.C.	48	Blue/red B		COM2:24,25,49,50 use
24	Green/red A	General-purpose output 116		49	Green/red B	General-purpose output 118	
25	Orange/red A	General-purpose output 117		50	Orange/red B	General-purpose output 119	

- The I/O number for the first addition is 100 added to the standard I/O number. (1 → 101)

(3) The second additional parallel I/O interface

Shown below is the pin assignment for the connector pin numbers of the second additional card and their functions.

Here is shown the distinction of the signal lines when you connect the optional external I/O cables. For details on the connector pin array and the external I/O cable, refer to section 4.2 "Option (22) , (23) " in the Specifications Manual.

Table 4.9 Directory of assignments for pin numbers and signals for the second additional parallel I/O interface

Pin number	Color of external I/O cable	Function name		Pin number	Color of external I/O cable	Function name	
		General purpose	Specialized/ power source/ common			General purpose	Specialized/ power source/ common
1	White/black A		FG	26	White/black B		FG
2	Yellow/black A		OV : 4-7 pin use	27	Yellow/black B		OV : 29-32 pin use
3	Blue/black A		12V/24V : 4-7 pin use	28	Blue/black B		12V/24V : 29-32 pin use
4	Green/black A	General-purpose output 200		29	Green/black B	General-purpose output 204	
5	Orange/black A	General-purpose output 201		30	Orange/black A	General-purpose output 205	
6	Pink/black A	General-purpose output 202		31	Pink/black B	General-purpose output 206	
7	Gray/black A	General-purpose output 203		32	Gray/black B	General-purpose output 207	
8	Red/black A		OV : 10-13 pin use	33	Red/black B		OV : 35-38 pin use
9	Purple/black A		12V/24V : 10-13 pin use	34	Purple/black B		12V/24V: 35-38 pin use
10	Brown/black A	General-purpose output 208		35	Brown/black B	General-purpose output 212	
11	White/black C	General-purpose output 209		36	White/black D	General-purpose output 213	
12	Yellow/black C	General-purpose output 210		37	Yellow/black D	General-purpose output 214	
13	Blue/black C	General-purpose output 211		38	Blue/black D	General-purpose output 215	/
14	Green/black C		COM0 : 15-22 pin use	39	Green/black D		COM1 : 40-47 pin use
15	Orange/black C	General-purpose output 200		40	Orange/black D	General-purpose output 208	
16	Pink/black C	General-purpose output 201		41	Pink/black D	General-purpose output 209	
17	Gray/black C	General-purpose output 202		42	Gray/black D	General-purpose output 210	
18	Red/black C	General-purpose output 203		43	Red/black D	General-purpose output 211	
19	Purple/black C	General-purpose output 204		44	Purple/black D	General-purpose output 212	
20	Brown/black C	General-purpose output 205		45	Brown/black D	General-purpose output 213	
21	White/red A	General-purpose output 206		46	White/red B	General-purpose output 214	
22	Yellow/red A	General-purpose output 207		47	Yellow/red B	General-purpose output 215	
23	Blue/red A		N.C.	48	Blue/red B		COM2:24,25,49,50 use
24	Green/red A	General-purpose output 216		49	Green/red B	General-purpose output 218	
25	Orange/red A	General-purpose output 217		50	Orange/red B	General-purpose output 219	

- The I/O number for the second addition is 200 added to the standard I/O number. (1 → 201)

4.1.6 Precautions for connections with external devices

⚠ Caution
- Do not exceed the designated limit for the power source of the external device.
- Be careful not to make a mistake about the polarity of the power source.

⚠ Caution
- Use a transistor's open collector signal or a non-voltage connection signal for the input signal.

⚠ Caution
- When using a coil load, such as a relay or solenoid, connect a surge absorbing diode, parallel to the load. (Be careful of the diode's polarity.)

⚠ Caution
- When the LED is lit, connect a protective resistor that is equivalent to the rated current, in serial.
- Connect the parallel register to get the drain current to be 20% as much as the rated current, because an incandescent lamp gives rush current 10 times as much as the rated current.

⚠ Caution
- When connecting the cables to peripheral equipment with the external I/O connector cables, do the installation as far away as possible from noise sources. Otherwise, it can cause a malfunction.

⚠ Caution
- Set the load connected to the output signal so that it does not exceed the maximum current of the output.

⚠ Caution
- Be careful so that you don't short-circuit the I/O cables with the power source.

  (It can destroy the output transistors that are inside.)

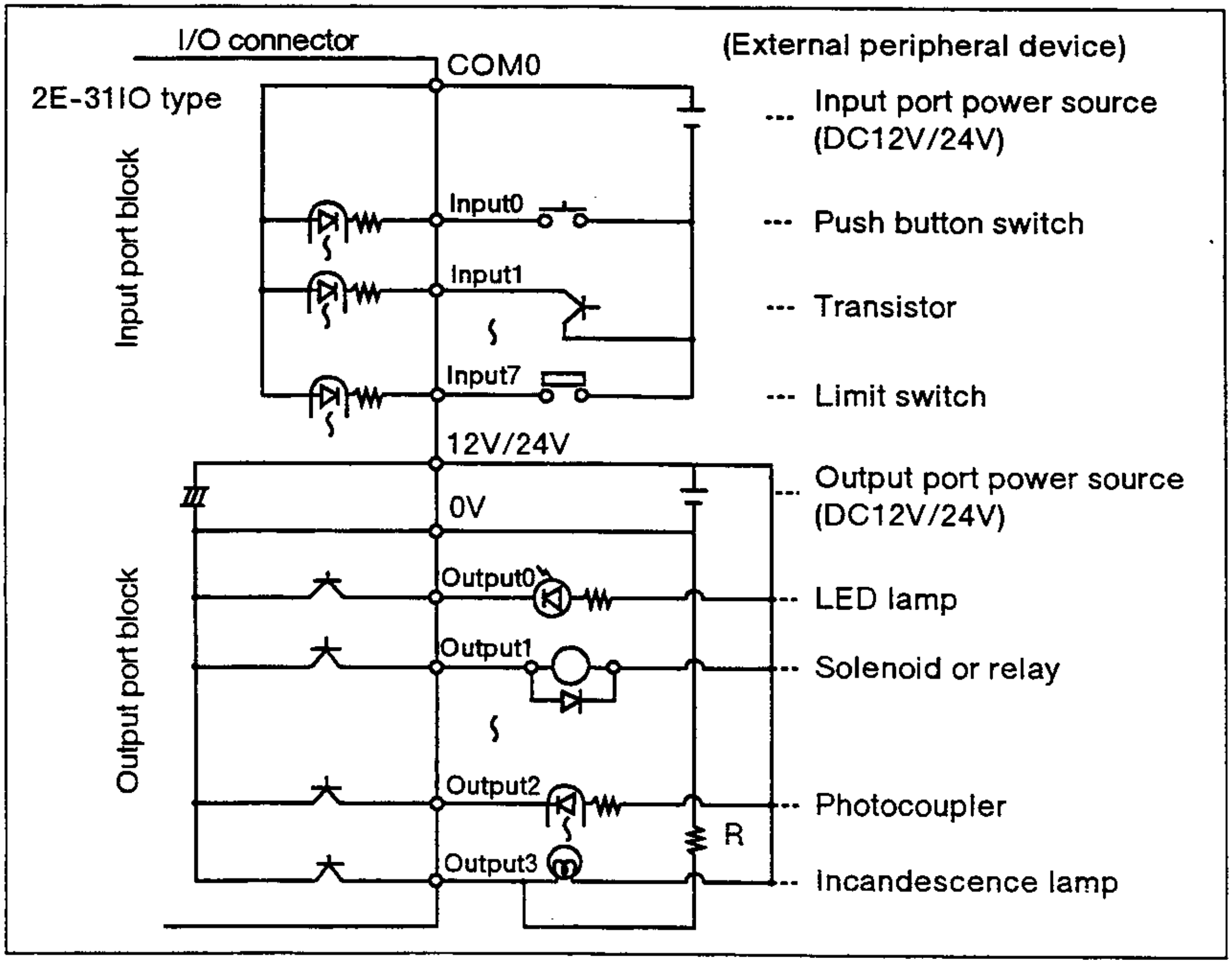


Fig. 4.3 Example of connection scheme for I/O circuitry (U.S.A.,Japan)

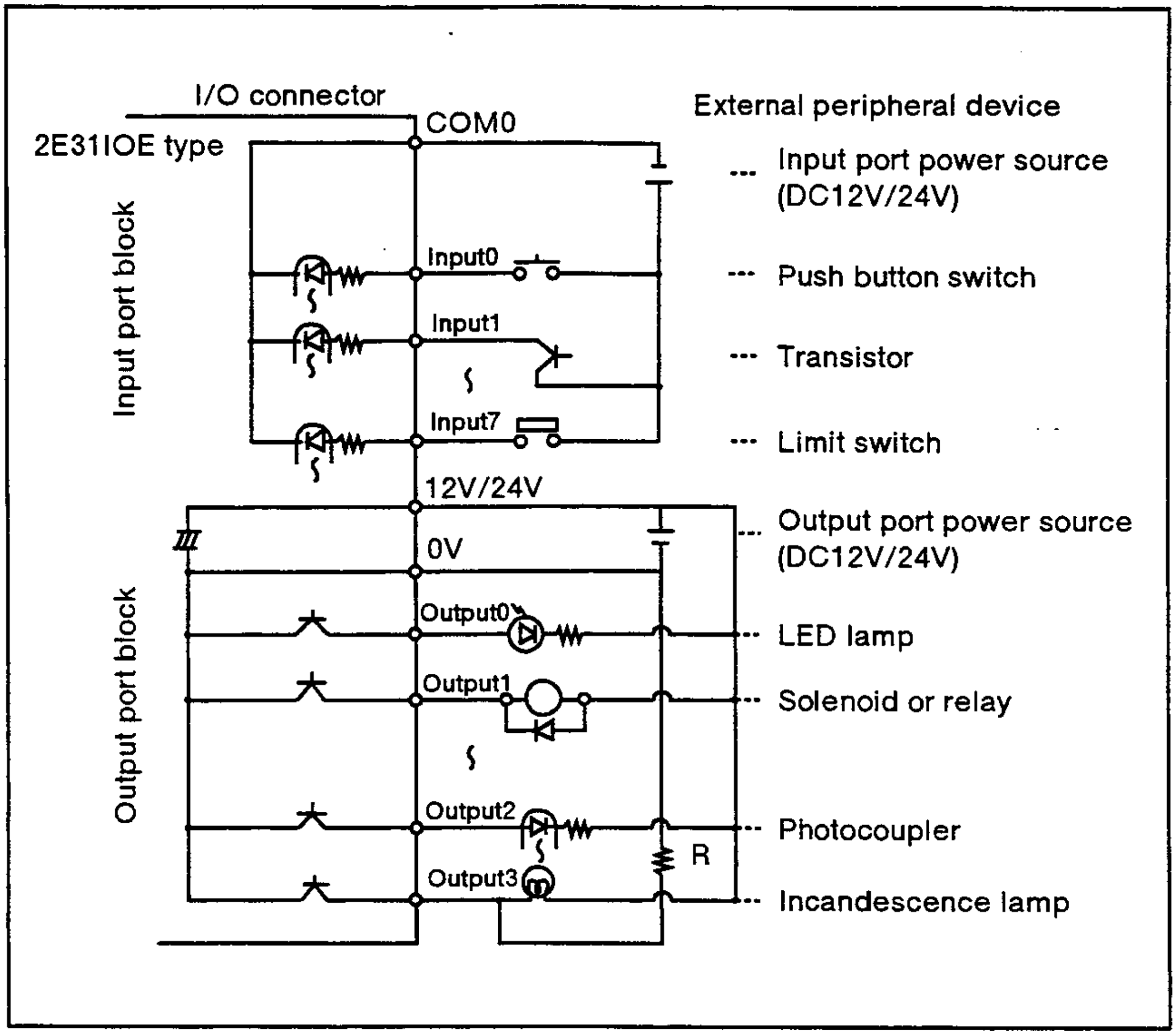
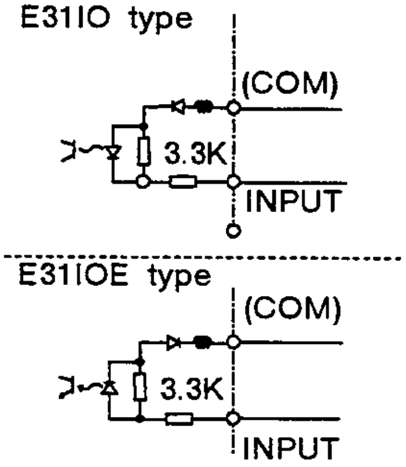


Fig. 4.3E Connection example to I/O circuitry (Europe)

(1) Electrical specifications for the input circuit

The electrical specifications for the input circuit is shown in Table 4.10

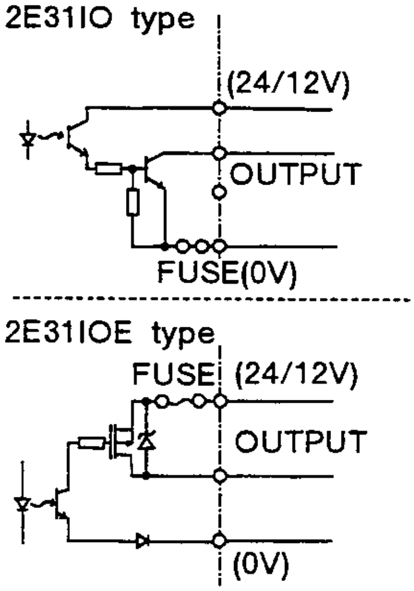
Table 4.10 Electrical specifications for the input circuit

Item		Specification	Internal circuit
Type		DC input	 E31IO type (COM) 3.3K INPUT E31IOE type (COM) 3.3K INPUT
No. of input points		20	
Insulation method		Photocoupler insulation	
Rated input voltage		DC12V/DC24V	
Rated input current		Approx. 3mA/Approx. 7mA	
Operating voltage range		DC10.2 to 26.4V (Ripple factor less than 5%)	
Voltage ON/ Current ON		DC8V or more/ 2mA or more	
Voltage OFF/ Current OFF		DC4V or less/ 1mA or less	
Input resistance		Approx. 3.3k ohms	
Response time	OFF-ON	10ms or less (DC24V)	
	ON-OFF	10ms or less (DC24V)	
Common method		8 points 1 common (Partly 4 points 1 common)	
External line connection method		Connector	

(2) Electrical specifications for the output circuit

The electrical specifications for the output circuit are shown in Table 4.11.

Table 4.11 Electrical specifications for the output circuit

Item		Specification	Internal circuit
Type		Transister output	 2E31IO type (24/12V) OUTPUT FUSE(0V) 2E31IOE type FUSE (24/12V) OUTPUT (0V)
No. of output points		16	
Insulation method		Photocoupler insulation	
Rated load voltage		DC12V/DC24V	
Usage load voltage range		DC10.2~30V (Peak voltage DC30V)	
Max. load current		0.1A / point (100%)	
Leakage current during OFF		0.1mA or less	
Max.voltage drop during ON		DC0.9V (TYP.)	
Response time	OFF-ON	2ms or less (Hardware response time)	
	ON-OFF	2ms or less (Resistor load)	
Fuse rating		Fuse 3.2A (one per common) cannot be replaced	
Common method		4 points 1 common	
External line connection method		Connector	
External supply power	Voltage	DC12/24V(DC10.2~30)	
	Current	60 mA (per TYP.DC24V1 common)	

※ The power supply (DC24V) for the input/output circuit must be prepared by the user.

(3) Selecting programs from external devices

Below you will find an explanation for a program selection method based on specialized input attached to an external input. There are two methods, so you can choose either one. For the parameter changing methods, refer to section 2.6 "Other functions" under " (11) Setting parameters" in this manual.

(A) Method using STA signal and PGN signal

1. Set the value of the program selection and start parameter PST to "0."
2. In one of the parallel I/O input setting parameters (IN1 to IN3) , assign the numerical value input signal (such as PI0) and the specialized input command "STA" and "PGN".
3. Set the program number to be selected by the external device at the various bits for the numerical value input signal.
4. Input the signal into PGN's bit. The program number that was set by the startup signal will be read into the robot.
5. Input the "STA" signal. The program you selected will start.

- ◇◆◇ Tip ◇◆◇

In the above method, when the program ends, and you want to use the same program again, input only the STA signal. (The numerical value input signal is unnecessary.) For information on the timing chart, refer to section 4.1.7.

(B) Method using the STA signal

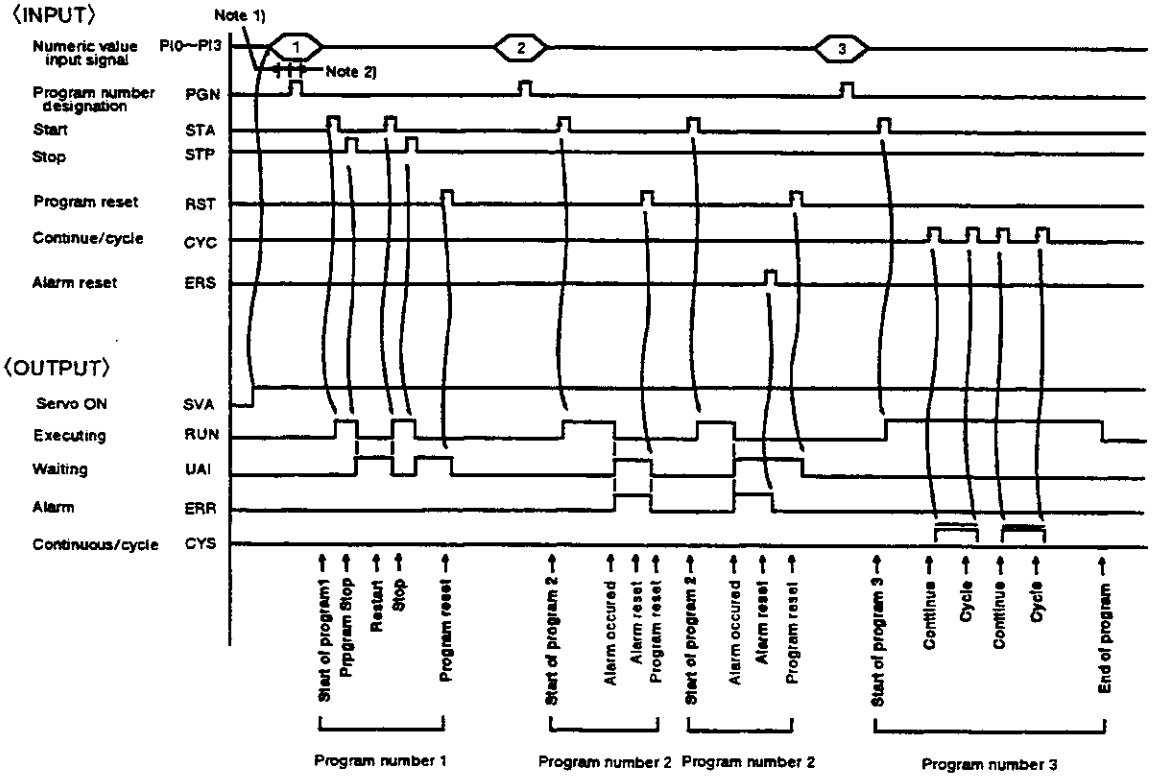
1. Set the value of the program selection and start parameter PST to "1."
2. In one of the parallel I/O input setting parameters (IN1 to IN3) , assign the numerical value input signal (such as PI0) and the specialized input command STA.
3. Set the program number to be selected by the external device at the various bits for the numerical value input signal.
4. Input the STA signal. The program number that was set by the startup signal will be read into the robot, and the program selected will start simultaneously.

⚠ Caution   In the above methods, it is necessary to set the program number at the numerical value input signal each time you start.
If you input only the STA signal, depending on the numerical value input signal, the wrong program can be selected.

4.1.7 Timing chart for external operations

Examples of timing charts for external operations are shown in Fig. 4.4 to Fig. 4.6

(1) Timing chart for external operations, Example (1):



Note 1 : After the setting of the numeric value input signal, you should input a PGN signal after 0.1 seconds.

Note 2 : The pulse width of the PE signal (positive edge) of the PGN signal and others should be 0.1seconds or more.

Fig.4.4 Timing chart for external operations, example 1

(2) Timing chart for external operations, example 2

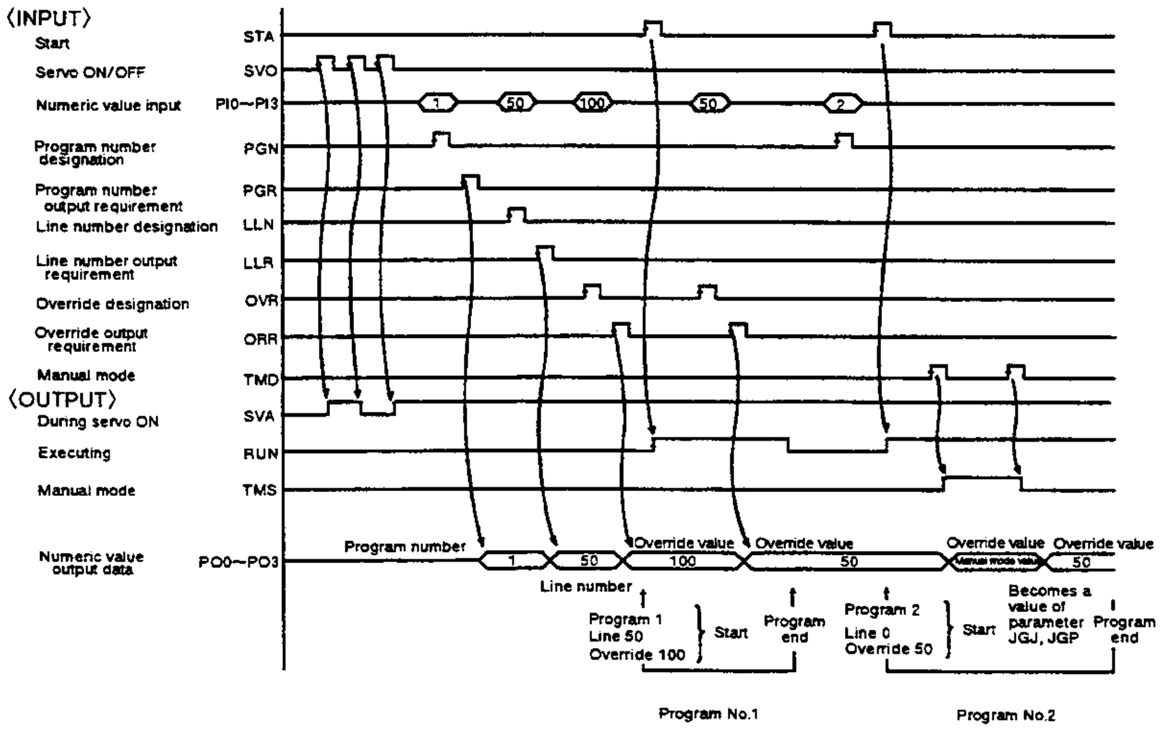


Fig. 4.5 Time chart for external operations, example 2

(3) Time chart for external operations, example 3

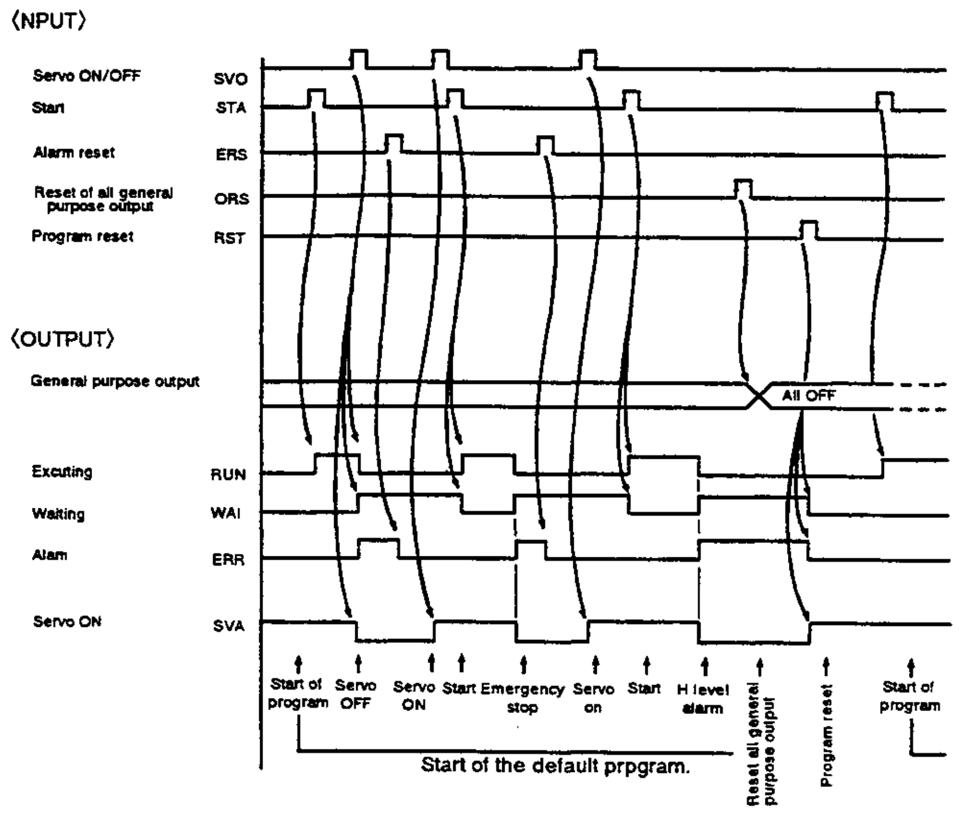


Fig. 4.6 Time chart for external operations, example 3

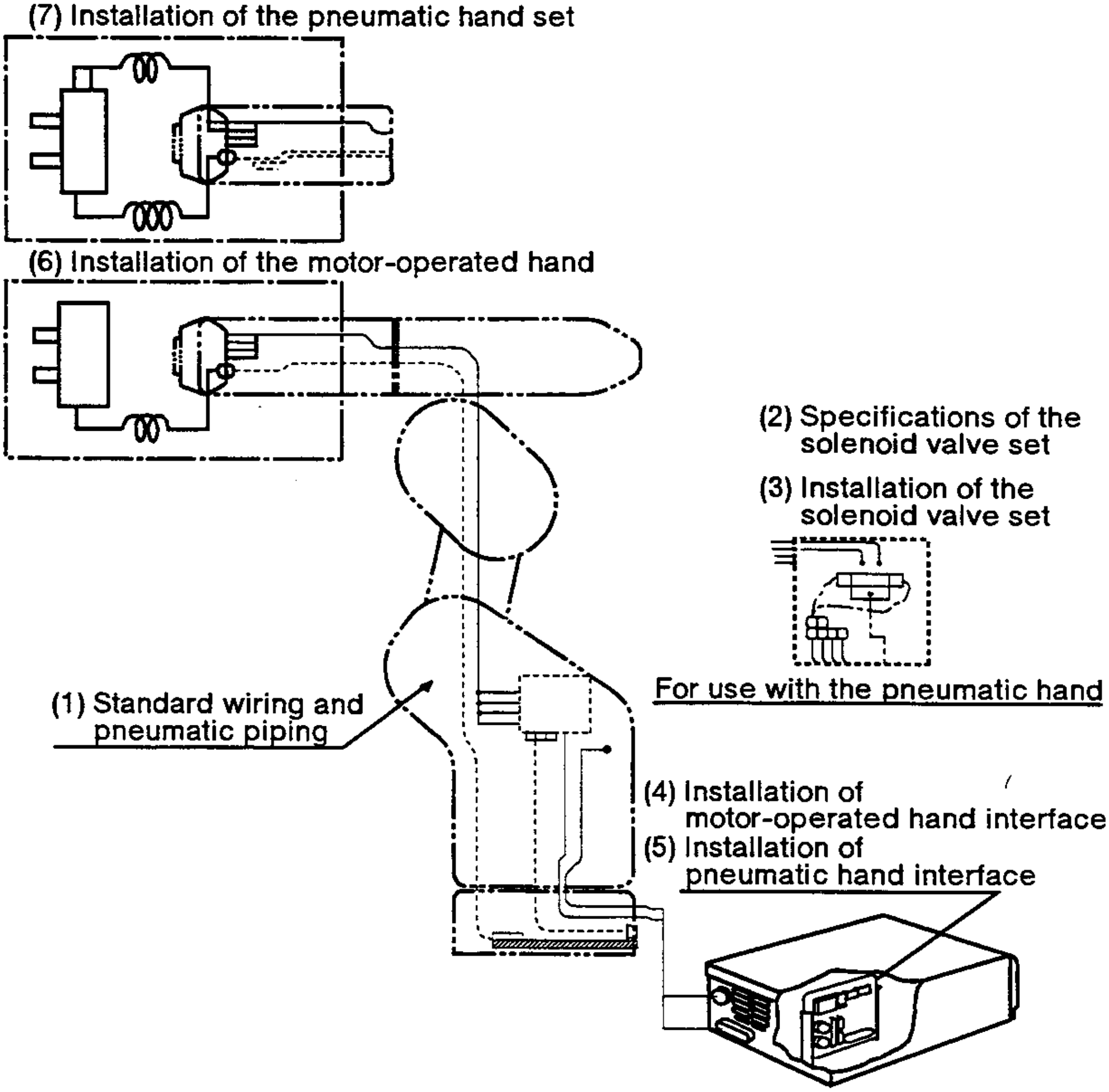
4.2 Tooling

4.2.1 Specifications for tooling connections

Details for making connections can be found in the Specifications Manual and in the instruction manuals (User's Manual and Reference Manual) , so please make the connections according to the written instructions. The topics can be found as follows:

(1)Standard air wiring and piping .... Specifications Chapter 3 Section 3.1.4
(2)Solenoid valve set specifications........... Specifications Chapter 4 Section 4.2 (3)
(3)Solenoid valve set installation .... Instruction Manual (Reference manual/ This manual) Chapter 1 Section 1.1.3
(4)Motor-operated hand interface installation......Instruction Manual (User's manual) Chapter 3 Section 3.8.1
(5)Pneumatic hand interface installation ... Instruction Manual (Reference manual/ This manual) Chapter 1 Section 1.1.4
(6)Motor-operated hand installation........... Instruction Manual (User's manual) Chapter 3 Section 3.8
(7)Pneumatic hand set installation .... Instruction Manual (Reference manual/ This manual) Chapter 1 Section 1.1.2

The numbers (1) to (7) found in the diagram correspond to the number headings in the previous pages.


(7) Installation of the pneumatic hand set
(6) Installation of the motor-operated hand
(2) Specifications of the solenoid valve set
(3) Installation of the solenoid valve set
For use with the pneumatic hand
(1) Standard wiring and pneumatic piping
(4) Installation of motor-operated hand interface
(5) Installation of pneumatic hand interface

Note : This was written for the RV-E2 robot. The specifications are the same for the RV-E2M and for the RV-E3J /RV-E3JM

Fig. 4.7 Connections for wiring and piping

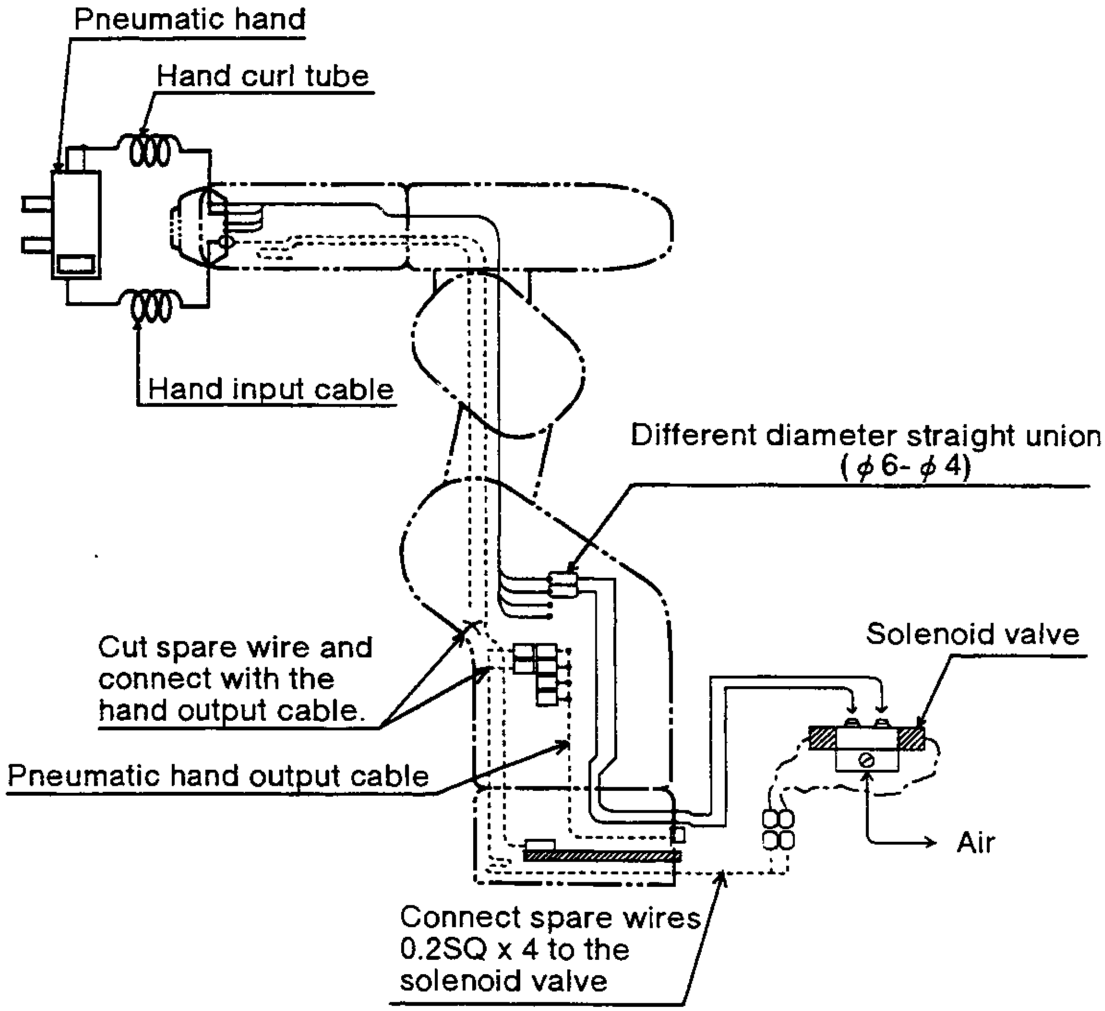
4.2.2 Example of tooling plan

In regards to a tooling plan, we'll explain a connection scheme involving "the method of installing an external solenoid valve." Here, we'll explain using a single hand, but even if you have a double hand, the basic procedure is the same. However, if you have 3 hands, everything will come under (2) Parallel I/O controls and external piping for the robot.

(1) Method of installing a solenoid valve externally (Using hand I/O for control)

We'll show you an example of installing a solenoid valve for pneumatic-hand control, such as one that may be different from the optional solenoid valve, and one that you may purchase separately yourself.

Here, we'll show you the way to connect a hand I/O to the hand output signal line. In this case, we'll use teaching to make it possible to open and close the hand. Vacuum based solenoid valves (ejectors) and other similar units will have the same wiring.

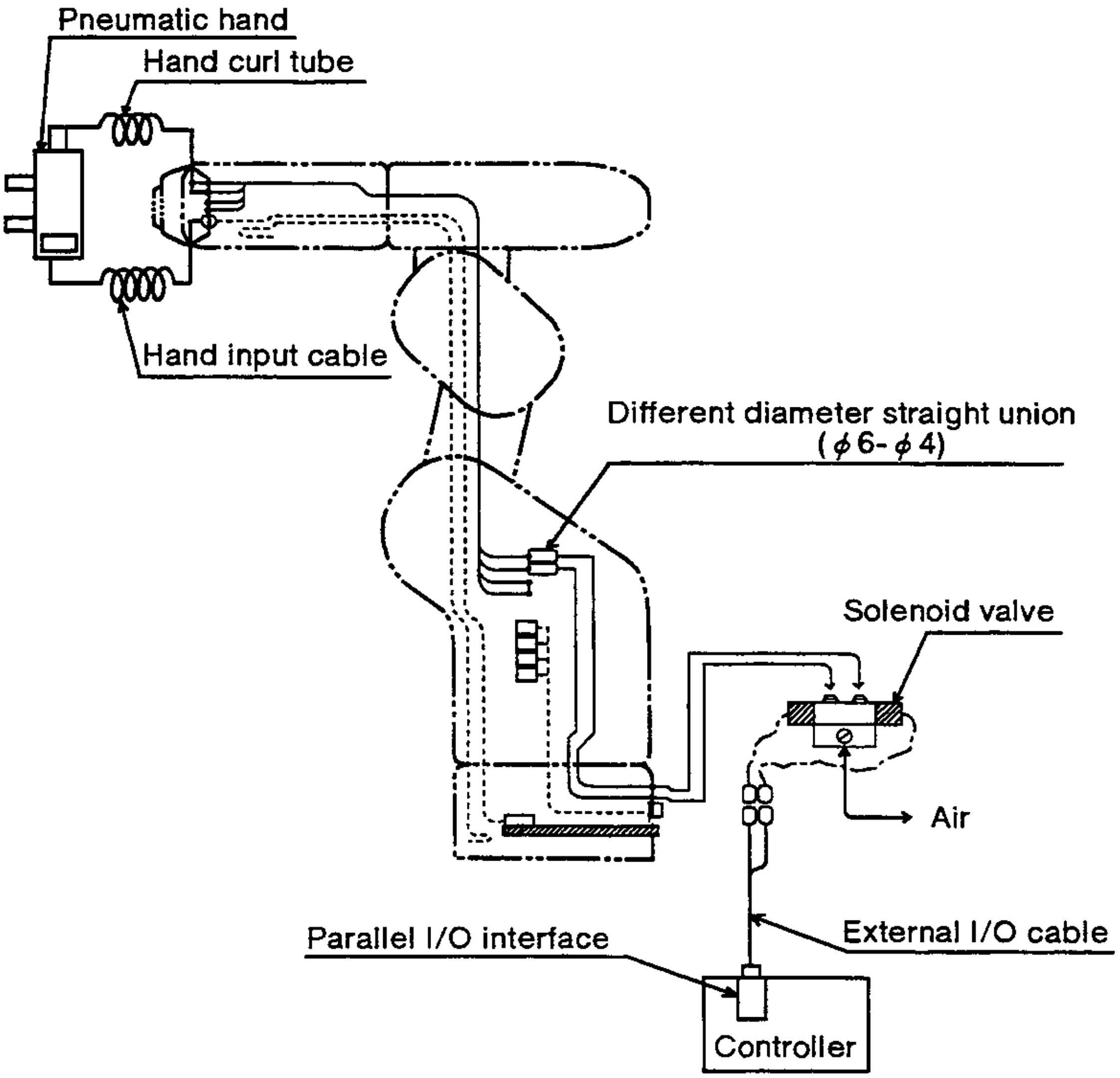


Note : This was written for the RV-E2 robot. The specifications are the same for the RV-E2M and for the RV-E3J /RV-E3JM

Fig. 4.8 Method of installing a solenoid valve externally (Using hand I/O for control)

⚠ Caution If you want to use 2 or more solenoid valves, put the air tube outside the robot cover.

(2) Method of installing a solenoid valve externally (using parallel I/O interface for control)
Here, we'll show you another method of installing a solenoid valve externally, a method of installing the had output control signal wire to the parallel I/O interface. In this case, you will not be able to open and close the hand with the teaching box. You should use the same wiring for a vacuum-type solenoid valve (ejector) or similar unit.



Note : This was written for the RV-E2 robot. The specifications are the same for the RV-E2M and for the RV-E3J /RV-E3JM

Fig. 4.9 Method of installing a solenoid valve externally (Using parallel I/O to control)

⚠ Caution If you want to use 2 or more solenoid valves, put the air tube outside the robot cover.

4.3 Connections to a personal computer (RS-232C)

4.3.1 Connector pin array for the RS-232C

The relationship between signal names and pin numbers for the RS232C connector pins is shown below.

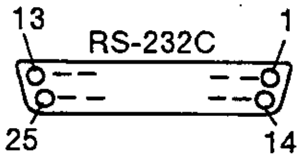


Fig. 4.10 Pin array (RS-232C)

Table 4.12 Pin numbers and signal names (for the RS-232C)

Pin number	Signal name	Pin number	Signal name	Pin number	Signal name
1	FG	10	N.C	19	N.C
2	$\overline{\text{SD}}$ ($\overline{\text{TXD}}$)	11	N.C	20	ER (DTR)
3	$\overline{\text{RD}}$ ($\overline{\text{RXD}}$)	12	N.C	21	N.C
4	RS (RTS)	13	N.C	22	N.C
5	CS (CTS)	14	N.C	23	N.C
6	DR (DSR)	15	N.C	24	N.C
7	SG	16	N.C	25	N.C
8	N.C	17	N.C	-	-
9	N.C	18	N.C	-	-

Adapter-connector : JAEDB-25P (prong side) N.C. stands for no connection possible.

4.3.2 Function of each signal cable

Table 4.13 Signal name and function

Signal name	Input/Output	Function
FG	-	This is a frame gland. Connects to the controller's G terminal
$\overline{\text{SD}}$ ($\overline{\text{TXD}}$)	Output	Transmission data from the controller to the personal computer
$\overline{\text{RD}}$ ($\overline{\text{RXD}}$)	Input	Receiving data from the personal computer to the controller
RS (RTS)	Output	Transmission request signal for the personal computer
CS (CTS)	Input	Transmission permission signal for the controller
DR (DSR)	Input	Personal computer's transmission ready signal.
SG	-	Ground for the signal line.
ER (DTR)	Output	Controller's transmission ready signal

4.3.3 Setting the RS-232C

When you use the RS-232C interface, it is necessary to set the below-mentioned communications with the personal computer. These settings are for making the same settings on the controller as for the personal computer. Unless they both have the same settings, communications can't work properly. Below are the communications specifications :

Table 4.14 Communications specifications

Item	Description
Baud rate	9600 bps
Data bit length	8 bits
Parity check	Even parity
Stop bit length	2 bits
New line code	CR

※ The settings on the left can also be made by changing the CM0 parameter. (Refer to 6.2 "Parameter directory" in this manual.) For the parameter changing methods, refer to section 2.6 "Other functions" under" (11) Setting parameters" in this manual.

Set the above-mentioned settings in your personal computer.

4.3.4 Time charts for the various signal lines

The standards for the RS-232C interface are essentially based on such items as electrical specifications, connector types, and pin numbers.

The way of using various signal lines and communications protocols are so varied depending on the type of equipment. Accordingly, even though you have made the correct signal lines connected to the personal computer, you still may not be able to operate with it.

Accordingly, in regards to making this connection, you should develop a full understanding of the functions of the signal lines that you will use between the controller and the personal computer before you make the connections. Please note that all data transmission between the controller and personal computer will be done in ASCII code.

(1) The timing of sending data transmissions from the personal computer to the robot.

‹Robot side›

ER (DTR) , and RS (RTS) should both be at the "H" level and be waiting for data input. When the end code is input, ER (DTR) and RS (RTS) both change to level L, and execute the command received.

Once the command has been executed, both ER (DTR) and RS (RTS) return to level H.

The end code is hexadecimal "OD" (CR : carriage return) , and or hexadecimal "OD" + "OA" (LF : line feed) .

‹Personal computer side›

When DR (DSR) is at level H, you can transmit characters. When DR (DSR) is at level L, if you transmit characters, an error will result on the robot side.

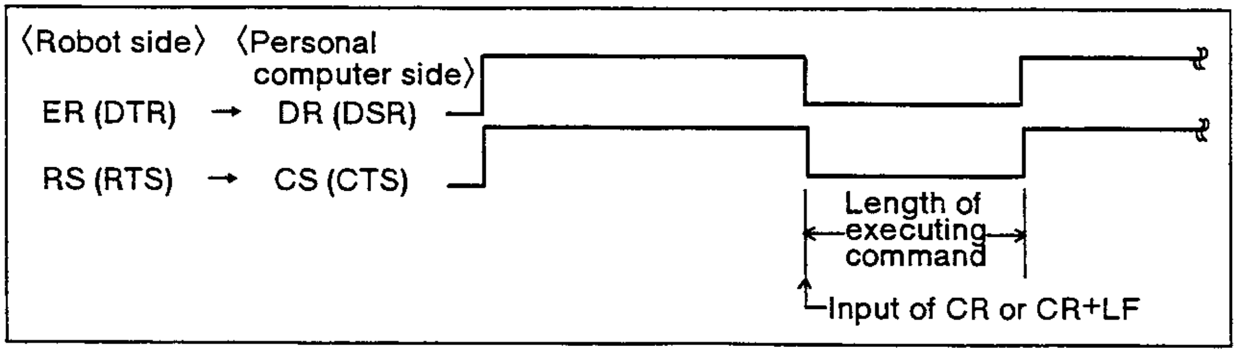


Fig. 4.11 Timing of data transmissions (from personal computer and robot)

(2) Time of data transmissions from the robot to the personal computer

‹Robot side›

After changing ER (DTR) to level H, you can begin to send data transmissions, and when the end code is transmitted, ER (DTR) 's level is changed back to level L. You can set the end code at parameter CM0. Refer to section 6.2 "Parameter directory" in this manual.

‹Personal computer side›

The request signal is sent after the ER (DTR) and RS (RTS) are both at level H, and the personal computer waits for a transmission from the robot side.

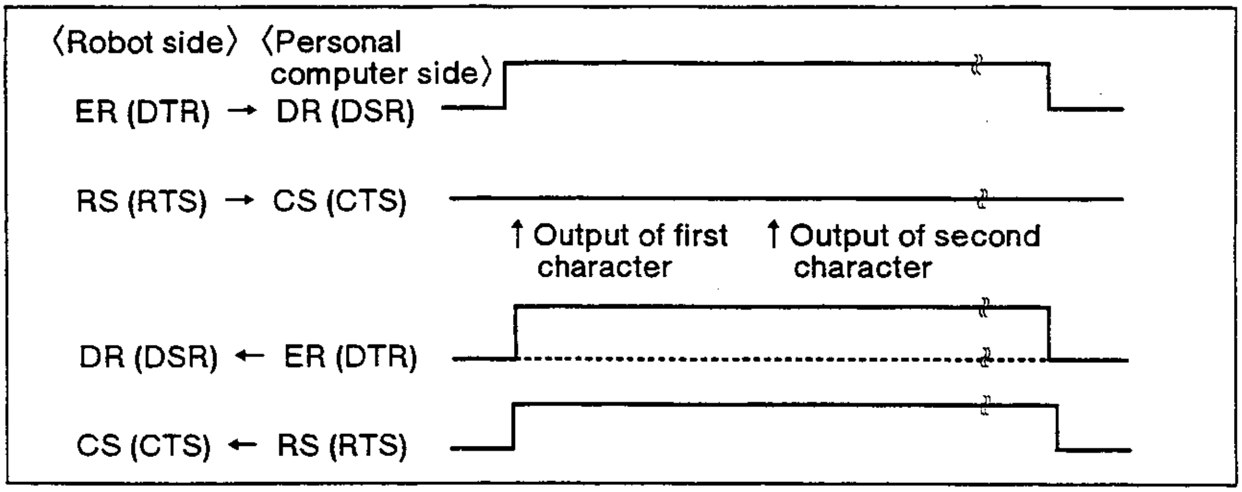


Fig. 4.12 Timing of data transmission (from robot to personal computer)

—◇◆◇ Tip ◇◆◇

1. When the personal computer transmits data to the robot, depending on the personal computer, the personal computer may ignore the ON/OFF signal line for the CS or DR, and transmit data continuously, resulting in an error on the robot side. In this case, make an adjustment on the personal computer, such as by using a timer to send data. On the other hand, when the robot transmits data to the personal computer, the robot will ignore the ON/OFF signal line of CS and ER on the personal computer, and transmit data.
2. When the RS-232C read command is used repeatedly at the program on the robot side, if the processing speed of the personal computer is slow, a communications error can result on the side of the personal computer. (Line buffer overflow error) . In this case, use a timer or some other technique so that you can expand the execution interval of the RS-232C read commands.
3. Th robot cannot receive new commands when it is executing direct commands, such as when it moves by the MO command. Transmit a new command only after it finishes a command completely.
   (By sending an ER command, you can confirm the existence of an alarm and the fact that a transmission has been completed.)
4. If the wrong command is transmitted by the RS-232C while the program is in progress, an alarm will be generated. In this case, reset the robot.

4.3.5 The RS-232C connector cable

If you are connecting the personal computer together with the robot with the RS-232C interface, you should adapt the cable you use for the RS-232C so that it can fit the connector on the front side of the controller. For information on wiring specifications of the connector cables, refer to section 4.2 "Options" under " (25) Personal computer cables" in the Specifications Manual. It will be necessary to satisfy the above-mentioned conditions related to timing, for all of the signals.

4.3.6 Example of a connection to a personal computer with the RS-232C

We'll show below an example of how to make a connection to the PC-9801N (NEC)

(1) Connecting the RS232C cable

Using the option cable built for the RS232C (Model type : RS-PC-CBL) , connect the connector on the personal computer with the controller. (Do this with the power source OFF for both the controller and personal computer.)

(2) Setting for the personal computer

1. Setting the main unit's switches

Set the dip switehes on the personal computer according to Table 4.15.

Table 4.15 Setting the dip switches

Switch	Condition	Notes
Switch 2	OFF (Up)	Changes the transmission mode of the RS-232C to asynchronous.
Switch 3	OFF (Up)	
Switch 5	ON (Down)	Prevents the memory switches from being initialized (changed) when the power source is ON.

After setting the dip switches, turn ON the power source for the personal computer. Depending on the personal computer, it may be possible to enter these settings by key entry on the personal computer.

2. Setting the software

Depending on the BASIC version you use, you can find the setting method in Table 4.16.

Table 4.16 How to set the software

Type of BASIC	Setting method
DISK-BASIC	(1) Start up N88DISK-BASIC. (2) Set the baud rate for the memory switches according to the procedures shown below: 1. Start up SWITCH.N88, which is attached to DISK-BASIC. (Input RUN "SWITCH.N88" and press RETURN Key). 2. In the Menu screen, select "RS-232C (default settings)" and press RETURN Key. 3. In the screen for RS-232C (default settings),set the baud rate to 9600 . 4. Press ESC to return to the original menu. 5. Select "END" and press RETURN Key.
DOS-BASIC	(1) Start up MS-DOS. (2) Using the switch commands, set the baud rate to 9600 for the memory switches. (For operating instructions, refer to the MS-DOS manual.)

(3) Confirming the connections

Using N88-BASIC, try inputting and executing the following program:

```
10  OPEN" COM 1 : E83" AS #1 ............Set the communication settings
20  PRINT #1, "ER"            ............Execute the ER command.
30  LINE INPUT #1, A$         ............Read the alarm status
40  PRINT "ALARM" ; A$        ............Display any alarm on the screen
50  END                       ............End the BASIC program
RUN                           ............Execute the program
ALARM0
OK
```

If you find that "0" follows after "ALARM" on the personal computer screen, then communications will be possible. ("1" or "2" will appear if an alarm is generated.)

Chapter 5 : Maintenance and inspection

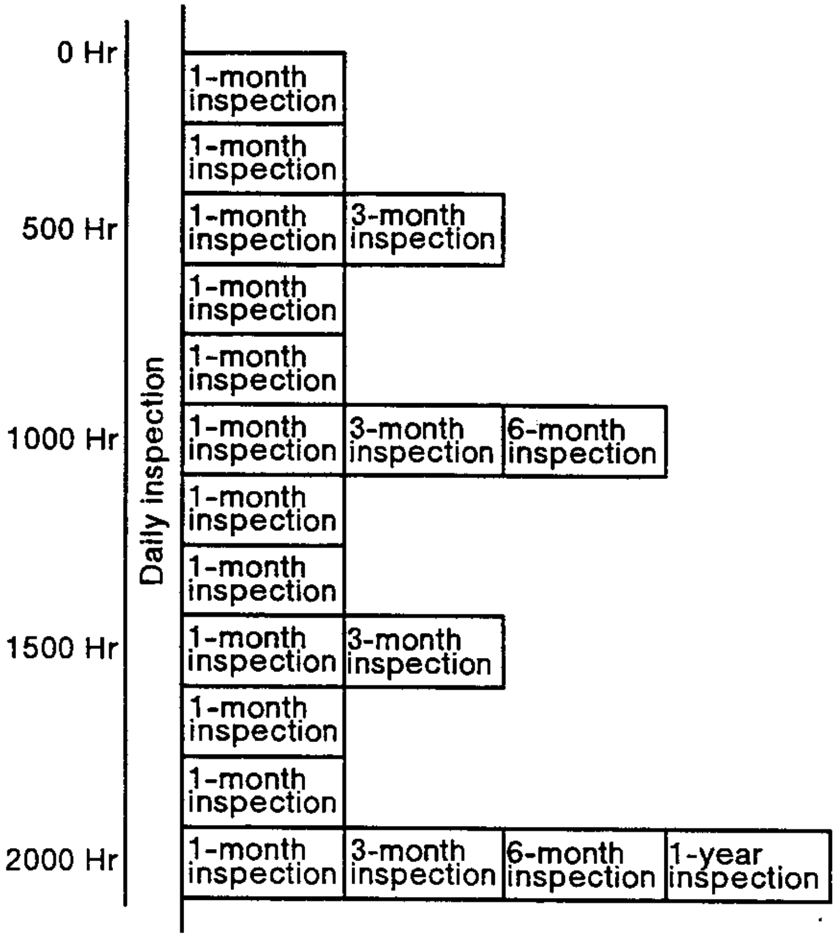
In this chapter we'll show you some maintenance and inspection duties that will help let you use the robot for a long time without any trouble. We'll also explain the types of expendable parts.

5.1 Maintenance period

Maintenance and inspection fall into two categories: daily and periodic. You should conduct maintenance and inspections diligently to promote safety and to prevent breakdowns before they happen so that you can enjoy a long product life.

(1) Inspection schedule

In addition to the monthly inspections, there is another additional inspection that must be made every 3 months (assuming operating hours total about 500 hours) , as shown below.



〈Calculations for inspection schedule〉

In case of a single shift

8hr/day x 20 days/month x 3 months = approx. 500hr.

10hr/day x 20 days/month x 3 months = approx. 600hr.

In case of a double shift

15 hr/day x 20 days/month x 3 months = approx. 1000 hr.

Note :

According to the schedule on the left, when using the double shift, you should make the inspections at half the regular intervals.

Fig. 5.1 Inspection schedule

5.2 Inspection items

5.2.1 Daily inspection items

Conduct the daily inspections in the order shown in Table 5.1

Table 5.1 Daily inspection items (Description)

Step	Inspection item (Description)	Remedial action
Before the power source is turned ON. (Inspect the following items before the power source is turned ON.)		
1	Are the installation bolts on the robot loose? (visually inspect)	Tighten the bolt securely.
2	Are the set screws on the cover loose? (visually inspect)	Securely tighten the screws.
3	Are the installation bolts on the hand loose? (visually inspect)	Tighten the bolt securely.
4	Is the power cable securely connected? (visually inspect)	Securely connect it.
5	Is the machine cable between the robot and controller securely connected? (visually inspect)	Securely connect it.
6	Are there any cracks or residue on the robot or controller cover? (visually inspect)	Exchange for new parts and take emergency measures.
7	Is the main robot unit leaking grease? (visually inspect)	Clean the robot and re-supply it with grease.
8	Are there any problems with the pneumatic system? Check for air leaks, drain residue, torn hoses, and whether the air source is normal. (visually inspect)	Take measures to repair drains or leaks. (Replace parts)
After the power source is turned ON (Watch the robot when you turn it ON.)		
1	Does the robot make strange movements or strange sounds when you turn it ON?	Refer to troubleshooting measures.
During operation (Use one of your programs to move the robot)		
1	Confirm that there are not gaps at the movement points. If there is a gap, confirm the following items: 1. Are the fixing bolts loose? 2. Are the bolts holding the hand loose? 3. Are there any gaps in the jig type positions other than the robot's? 4. If the gap does not heal, refer to 6.6 "Troubleshooting" in this manual and then take the appropriate measures.	Refer to troubleshooting measures.
2	Does the robot move strangely or make strange sounds? (visually inspect)	Refer to troubleshooting measures.

5.2.2 Periodic inspection

Conduct the periodic inspections following the procedures in Table 5.2

Table 5.2 Periodic inspection items

Step	Inspection item (Description)	Remedial action
Monthly inspection		
1	Are there any loose screws or bolts on the main robot unit?	Securely fasten the screws and bolts.
2	Are the connector's fixed screws or the terminal block's screws loose?	Securely fasten the screws.
3	Remove the various covers and confirm that there is no stain, damage, or friction on the various cables.	Determine the cause and remove it. If there is any big damage to the cables, contact the service desk at Mitsubishi.
3-month inspection		
1	Are there any problems with the tension of the timing belt?	Adjust it so that the tension is not too loose nor too tight. (See page 5.3.3)
6-month inspection		
1	Confirm the abrasion of the timing gear teeth.	When gear teeth get badly damaged, replace them.
1-year inspection items		
1	Supply grease for the harmonic reduction gears for each axis.	Refer to page 5.3.4 for information about greasing.
2	Replace the backup batteries in the main robot unit and inside the controller.	Refer to page 5.3.5 when you do the replacing.

5.3 Maintenance and inspection procedures

In the following we'll explain the actual procedures for conducting the periodic maintenance and inspection. Study the descriptions carefully and follow the instructions when you carry out the tasks. You can arrange to have these tasks done for you by contacting the Mitsubishi service desk, but you will be charged for these services. (Never disassemble the parts except for the parts described below.)
If you decide to implement the maintenance and inspection program yourself, you can find information about the maintenance parts in section 5.4 "Maintenance parts" in this manual. If you need some parts, you can contact the dealer where you purchased this product, or you can contact your local Mitsubishi service center.

⚠ Caution If you undertake these duties, there may be some items that result in origin slips of the machine system. You may need to do "revising the body position" or "re-teaching."

5.3.1 Structure of the main robot unit In

Figure 5.2.1 shows a structural outline drawing for the RV-E2 /RV-E2M and Figure 5.2.2 shows one for the RV-E3J/RV-E3JM. The various sections are as follows:

(1) Waist joint
The rotation of the waist moves according to 1) the W joint motor and 2) the harmonic reduction gear.
The W joint motor is equipped with a brake.

(2) Shoulder joint
The rotation of the shoulder moves according to 3) the S-joint motor located in the upper arm and 4) S-joint reduction gear.
The S-joint motor, 3) , is equipped with a brake in order to prevent the robot from falling when the power source is OFF.

(3) Elbow joint
The elbow joint moves according to 5) the E-joint motor located in the upper arm, and 6) the E-joint reduction gear.
The E-joint motor, 5) , is equipped with a brake.

(4) Wrist twist joint (For the RV-E2/RV-E2M only)
The wrist twist joint rotates according to 7) the T-joint motor located in the elbow block and 8) the T-joint reduction gear.

(5) Wrist pitch joint area
The rotation of 9) the P-joint motor located in the forearm is transfered to 11) the P-joint reduction gear through 10) the P-joint timing belt, resulting the wrist housing.

(6) Wrist roll joint
The rotation of the wrist roll moves according to 12) the R-joint motor in the wrist housing and to 13) the joint reduction gear.

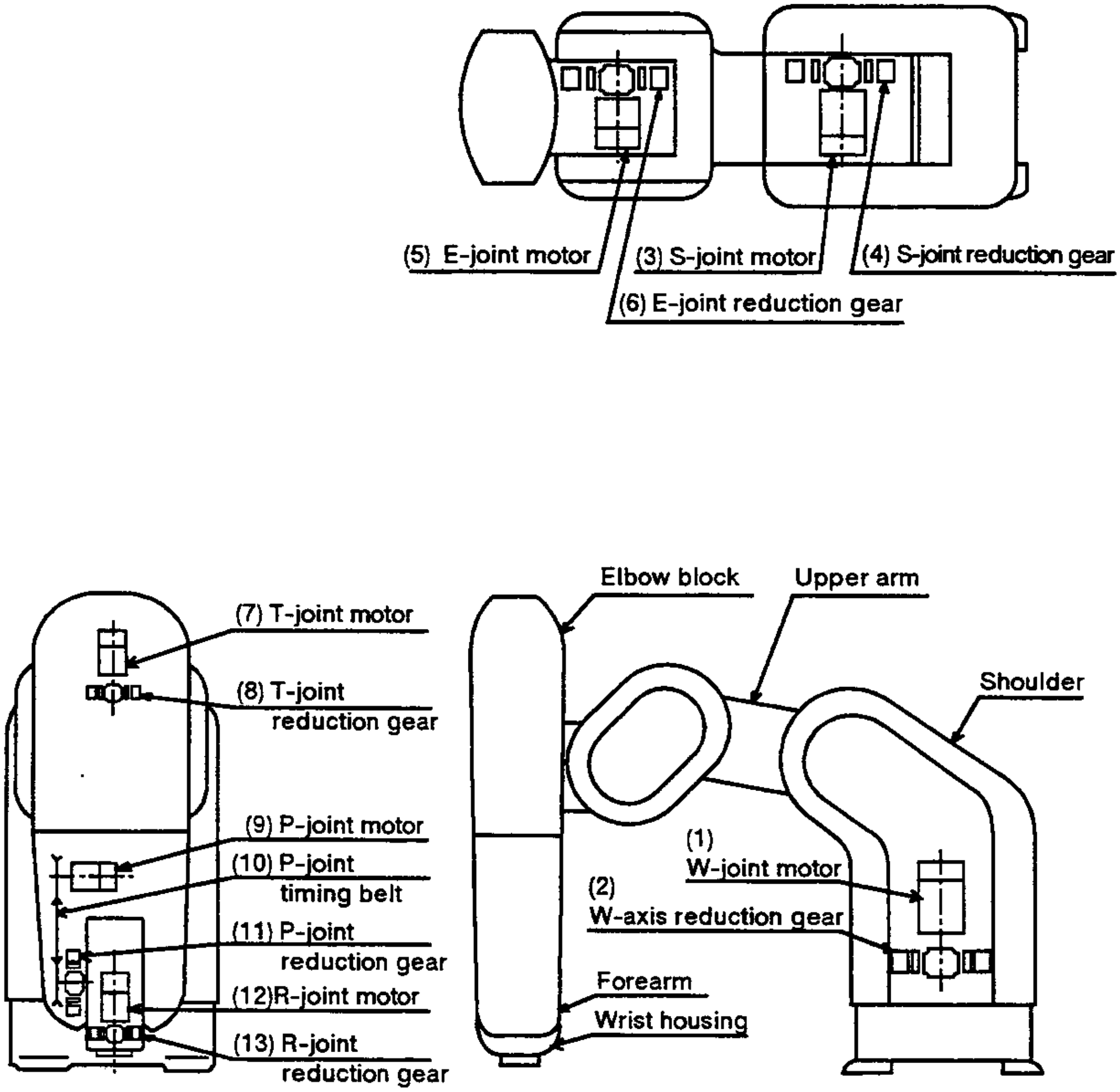


Fig. 5.2.1 Main robot unit's outline structural diagram (RV-E2/RV-E2M)

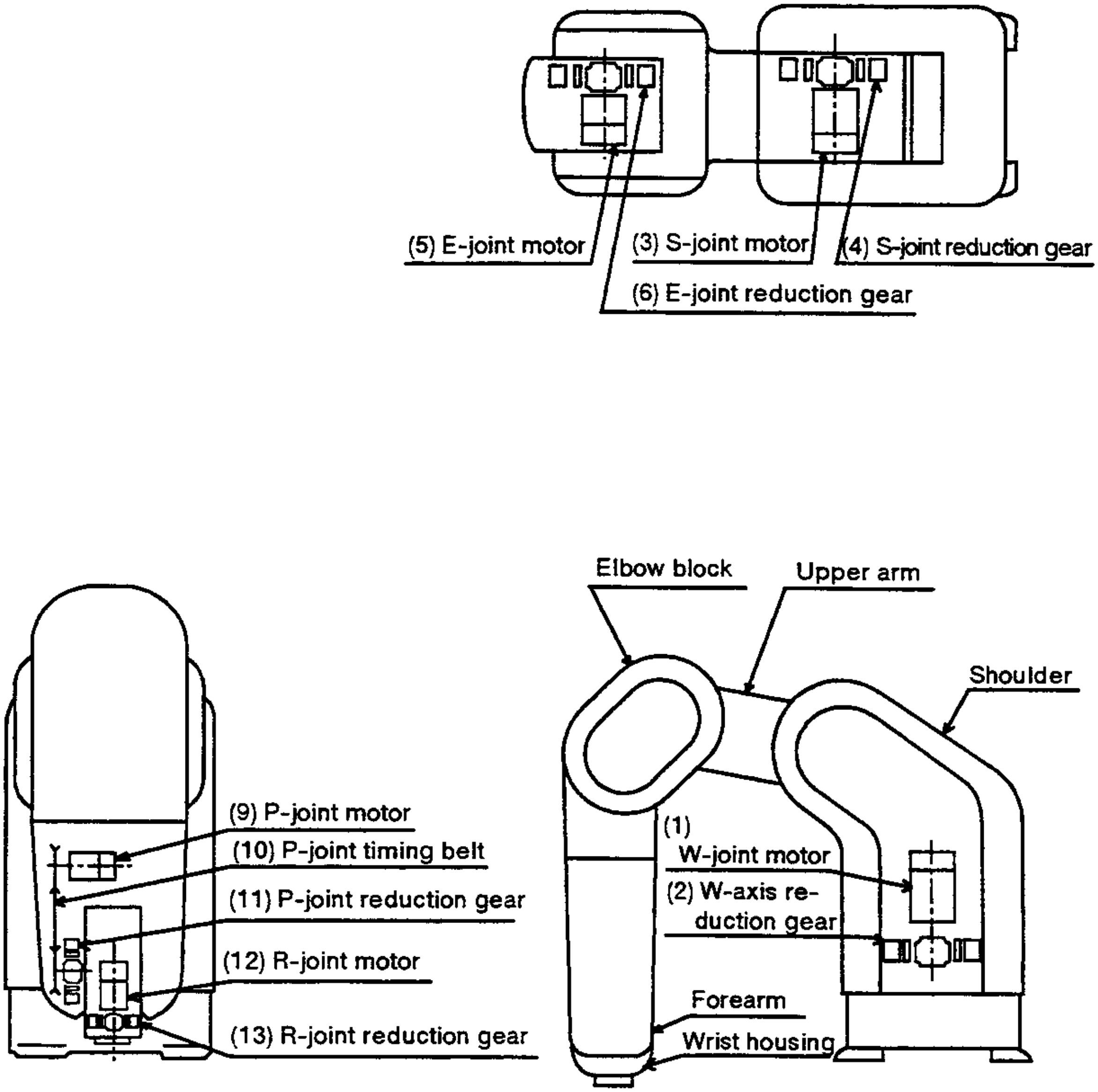


Fig. 5.2.2 Main robot unit's outline structural diagram (RV-E3J/RV-E3JM)

5.3.2 Installation/removal of cover

(1)If the robot is the RV-E2/RV-E2M, refer to Fig.5.3.1, and if the robot is RV-E3J/RV-E3JM, refer to Fig. 5.3.2, and remove the cover.

(2)The names of the covers are shown in Table 5.3, and the list of cover installation screws is shown in Table 5.4.

(3)Some covers may be difficult to remove depending on the posture of the robot. In this case, change the robot posture with jogging operation, and then remove the cover.

(4)Use the removal procedure inreverse to install the cover after maintenance and inspection.

Table 5.3 Cover name

Product number	Cover name	Quantity	Notes
1	Shoulder cover (L)	1	
2	Shoulder cover (R)	1	
3	Shoulder cover (F)	1	
4	Shoulder cover (B)	1	
5	NO. 1 Arm cover (R)	1	
6	NO. 1 Arm cover (L)	1	
7	Elbow cover (U)	1	
8	For the RV-E2/RV-E2M Elbow cover (B)	1	
	For the RV-E3J /RV-E3JM Elbow cover (L)	1	
9	Elbow cover (F)	1	Applies only to RV-E2/ RV-E2M
10	NO. 2 Arm cover (U)	1	/
11	NO. 2 Arm cover (L)	1	
12	Wrist cover (U)	1	
13	Wrist cover (L)	1	
14	Wrist cover (S)	2	

Table 5.4 List of cover installation screws

Symbol	Installation screw name	Qty.	Remarks
a	Socket bolt M3X8 (nickel plated)	27	
b	Truss screw M3X6 (nickel plated)	10	
c	Truss scrrew M3X8 (nickel plated)	8	
d	Socket bolt M3X6 (nickel plated)	2	Applies only to RV-E2/ RV-E2M

The product numbers and symbols used in Table 5.3 and Table 5.4 correspond to those used in Fig. 5.3.1and Fig. 5.3.2.

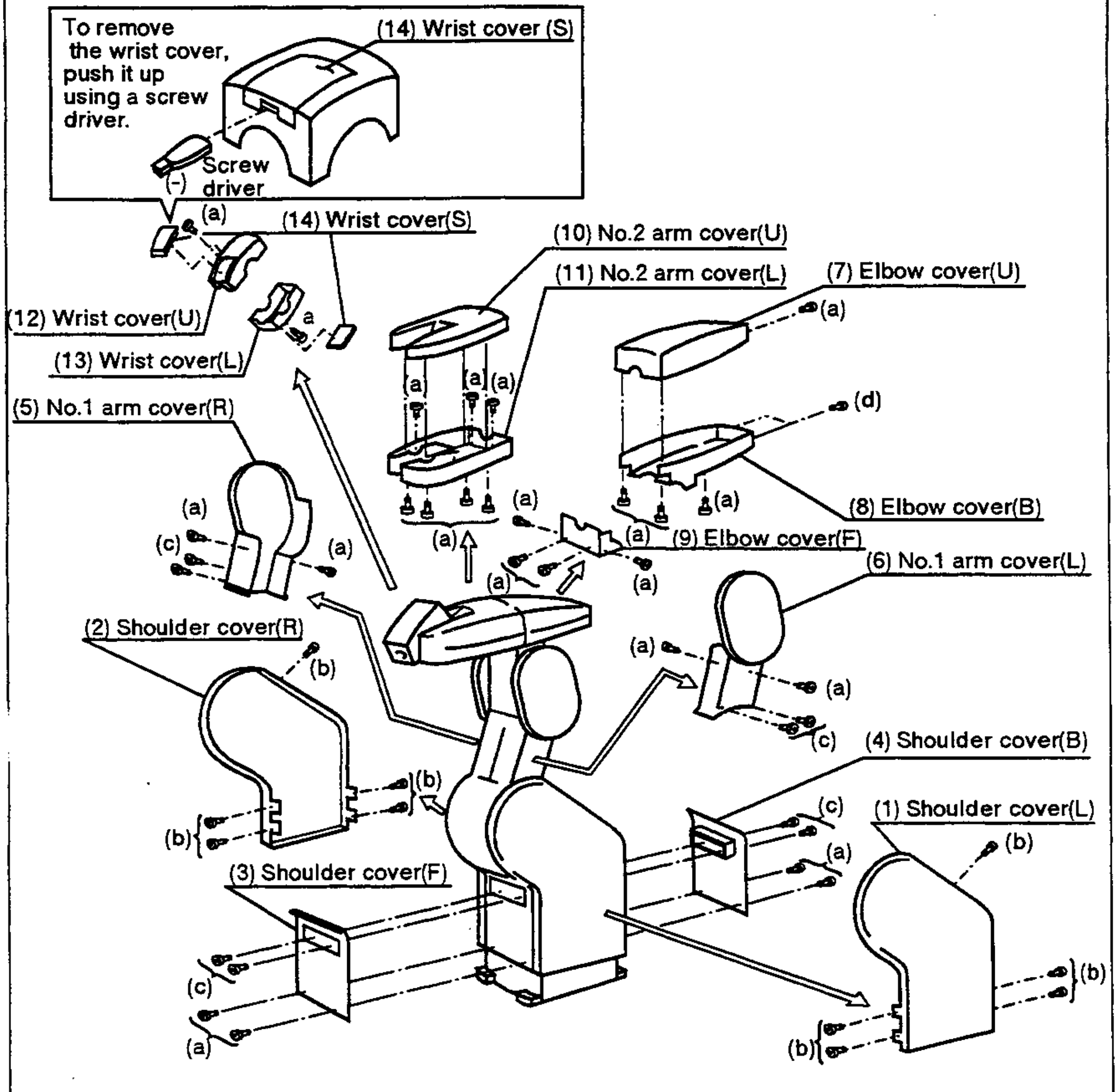


Fig. 5.3.1 How to disassemble the covers (RV-E2/RV-E2M)

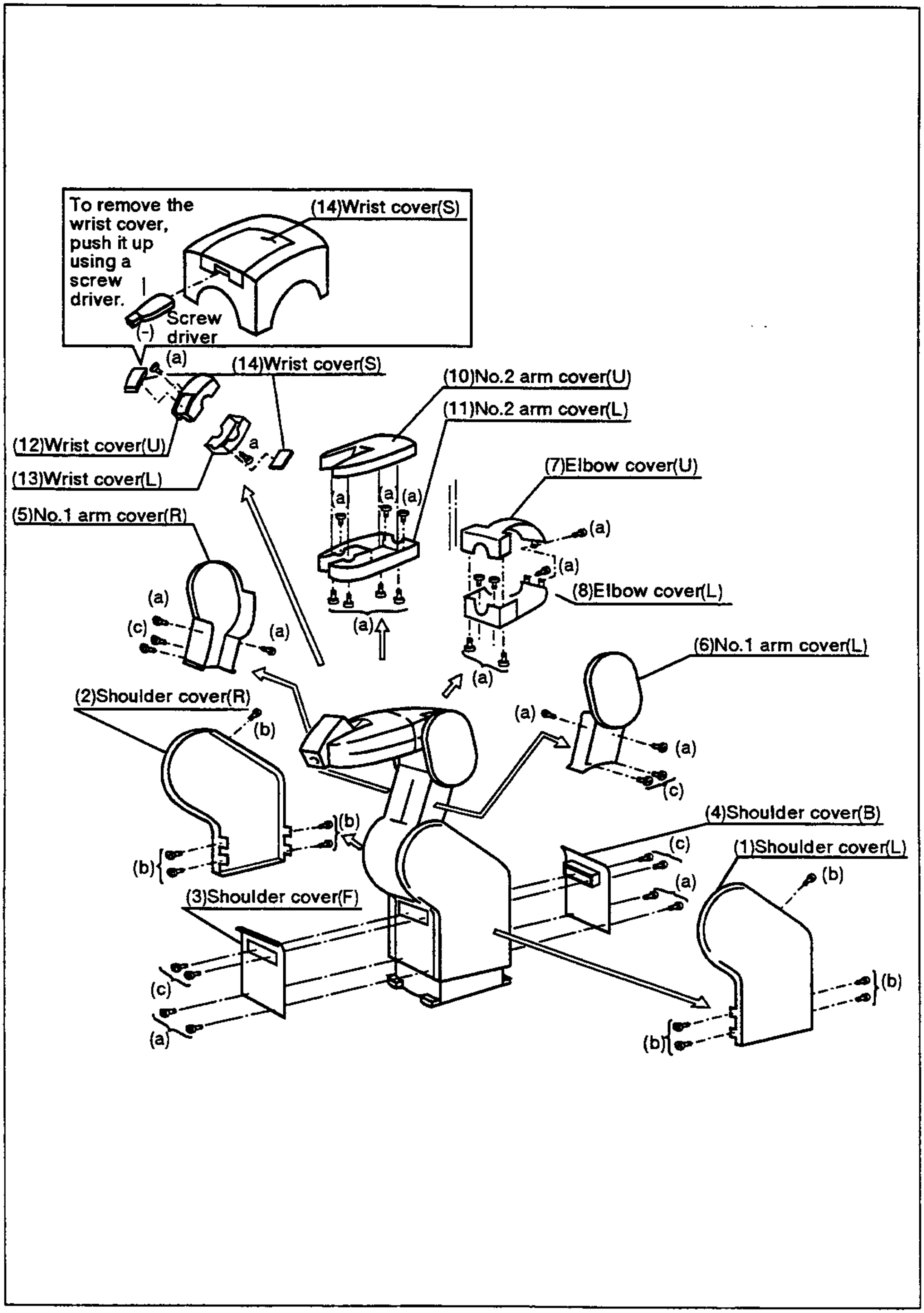


Fig. 5.3.2 How to disassemble the covers (RV-E3J/RV-E3JM)

5.3.3 Inspecting, adjusting, and replacing the timing belts

The robot uses a timing belt to drive the pitch joint. The timing belt is superior to gears or chains in that it requires no lubrication and makes little noise. When tensile control or tension adjusting is unsuitable, that could cause a shorter product life and generate noise.

We carry out an aging operation to remove initial elongation and we also adjust the belt tensile before shipment. But tensile confirmation is necessary in the routine inspections, because elongation occurs over a long period of time, in some cases.

Replacing the timing belt is necessary in the following cases:

*Replacing the timing belt

It's had to judge the product life of the timing belt because it depends on the running environment.

(1) Replace the timing belts under these cases:

1. It gets cracked on the base of a tooth or the back of the belt tooth.
2. It swells out as a result of stuck oil.
3. It looses its width (approx. half a tooth width)
4. It jumps the gear tooth by abrasion.
5. It breaks.

⚠ Caution Initial abrasion of the timing belt is unavoidable. It is not an abnormality if you find abrasion powder in the cover after about 300 hours of operation. Change the belt, even if you can wipe off the abrasion powder.

(2) Inspecting, adjusting, and replacing the timer belt for the wrist pitch joint drive.

A. Inspection method

1. Figure 5.4 shows how to inspect and adjust a timing belt for wrist pitch joint drive.
2. Remove the no.2 arm covers U and L as shown in the Figures 5.3 "How to disassemble the covers."
3. Visually confirm that situation shown in the former section actually happened.
4. Push the middle o f the belt with the finger lightly. (About 0.7Kgf) as shown in Fig. 5.5 and confirm that it sags 2mm.

B. Adjustment method

1. Fig. 5.4 shows how to inspect and adjust a timing belt for wrist pitch joint drive.
2. Loosen the 4 motor installation screws
3. Move the motor to the direction of the arrow in Fig. 5.4 to adjust the tension by confirming the belt. The motor casing has long hole to adjust it.
4. Move the motor to arrow "a" to fasten the belt. Move the motor in the direction of the arrow "b" to loosen the belt.
5. Don't loosen the belt too much so that it comes off the timing pulleys, 4, 5.
6. Fasten the 4 motor installation screws tightly after making the adjustments. If they are not securely fixed, they can come loose from vibration.

C. Replacement method

1. Fig. 5.4 shows how to replace a timing belt.
2. Turn down the wrist pitch joint with the teaching box. (Turn it to the direction of gravity)
3. Hold the pulleys when replacing the belt. It can cause a location gap where the location relationship of pulley (4) and pulley (5) slips off.
4. Mark the timing belt (2) and the timing pulleys (4) and (5) to keep their location.
5. Put off the motor installation screws to remove the timing belt.
6. Copy the mark to a new timing belt by stretching the belt.
7. Assemble a new timing belt into the timing pulleys to set the marks.
8. Adjust the tension of the belt. The former items (3) to (6) explain how to adjust it.
9. There is the possibility of a location gap being created after replacing the belt. Confirm that a location gap has not occurred.
10. Adjust the robot when it gets a location gap. Refer to section 3.9 "Setting the origin" in the User's Manual or 'Setting the origin" in this manual to reset the zero point position.

⚠ Caution When you replace the belt, a machine-caused location gap of the origin can be generated. In this case, you will need to reenter the position data.

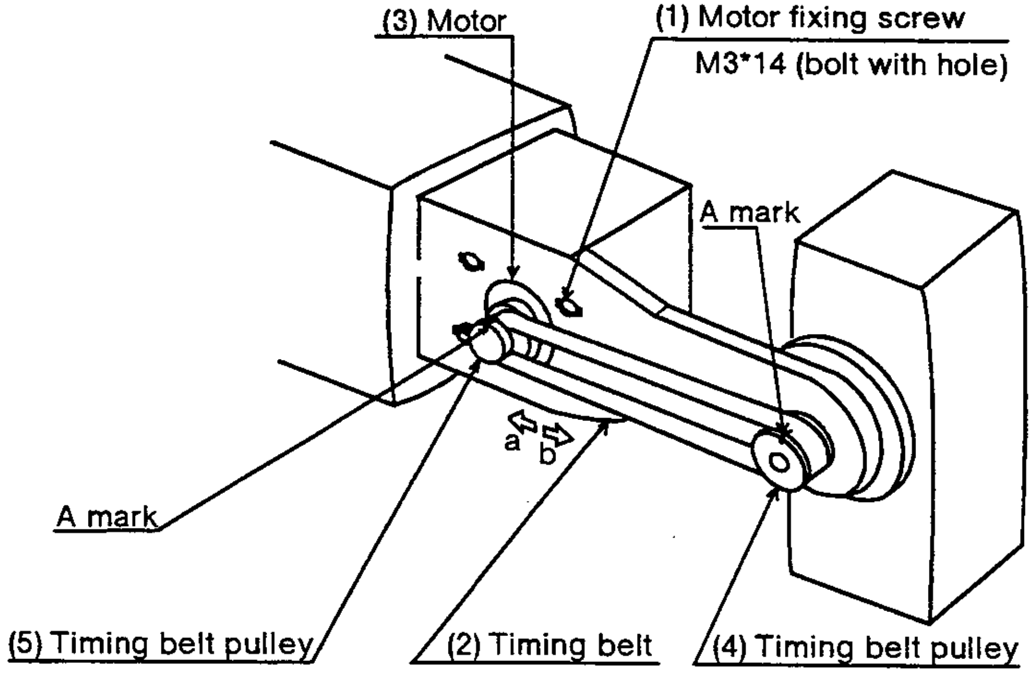
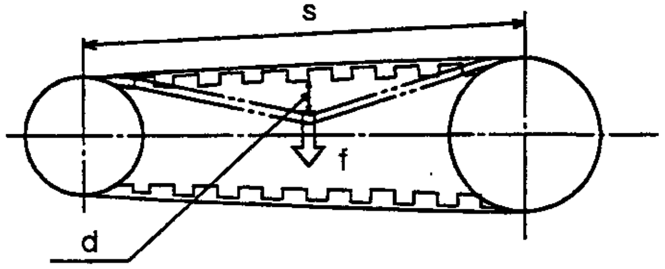


Fig. 5.4 Inspecting, adjusting, and replacing the timing belt for the wrist pitch joint drive.

(3) Tension of the timing belt

A timing belt needs suitable tension to keep good power transmission and its durability. When the tension of the belt is weak, the sides of the belt vibrate. When the tension of the belt is strong, the tension side of the belt vibrates with a sharp sound. Fig. 5.5 and Table 5.5 illustrate the how to adjust the tension of the belt. Adjust the tension of the belt by the force f and the deflection of the span, s.



f : Tension of belt kgf
s : Length of span mm
d : Deflection of belt mm

Fig. 5.5 Force and deflection

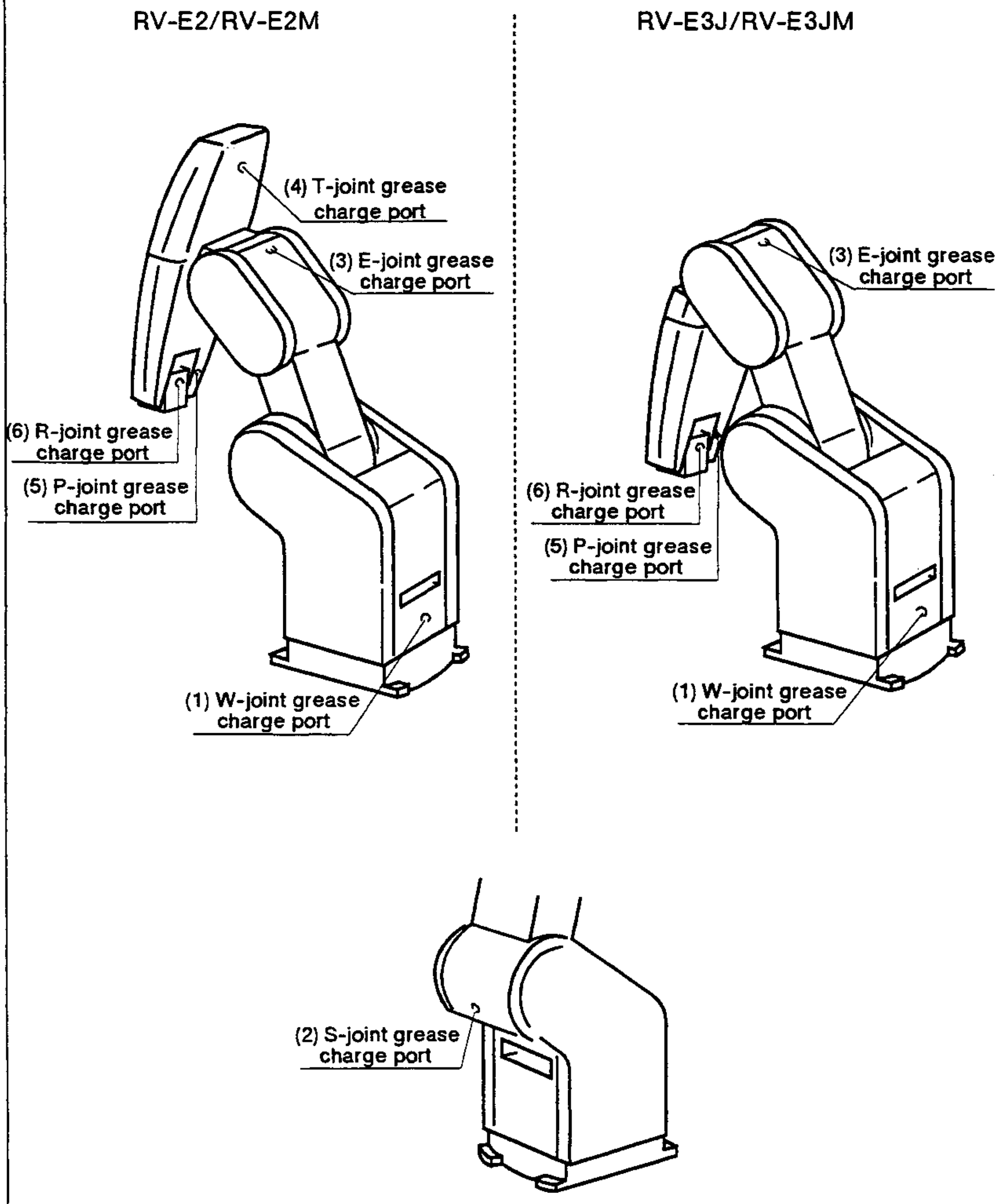
Table 5.5 Tension of the belt

Joint	Belt type	Span s: mm	Deflection d: mm	Force f: kgf
P Joint	S2M-320-U	120	2	0.7

5.3.4 Grease charge

(1) Grease charge place and grease charge specification

Fig. 5.6 shows the location for the grease nipple. And Table 5.6 shows grease charge specifications for each place. When you want to charge grease, follow section 5.3.2 "Disassembling covers" to remove the necessary covers.


RV-E2/RV-E2M
RV-E3J/RV-E3JM
(4) T-joint grease charge port
(3) E-joint grease charge port
(6) R-joint grease charge port
(5) P-joint grease charge port
(1) W-joint grease charge port
(3) E-joint grease charge port
(6) R-joint grease charge port
(5) P-joint grease charge port
(1) W-joint grease charge port
(2) S-joint grease charge port

Common for RV-E2/RV-E2M, RV-E3J/RV-E3JM

Fig. 5.6 Grease charge ports

Table 5.6 Grease charge specification

Num ber	Grease charge port	Grease type	Feed lubricating oil Volume at time of shipment	Hours of supply Grease charge interval	Removal cover
1	Waist joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 40g	※ 2000Hr 17g	Shoulder cover (B)
2	Shoulder joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 40g	※ 2000Hr 17g	
3	Elbow joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 40g	※ 2000Hr 17g	
4	Wrist twist joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 6.5Kg	※ 2000Hr 6.2g	
5	Wrist pitch joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 6.5Kg	※ 2000Hr 1.6g	
6	Wrist roll joint Reduction gear	Grease nipple WA-610	Grease Gear grease SK-1A 6.5Kg	※ 2000Hr 3.5g	Wrist cover (S)

Note : The RV-E3J/RV-E3JM in number 4 have no wrist twist joint.

- ◇◆◇ Tip ◇◆◇

1. The brand names in Table 5.6 are the ones used for the robot at the time of factory shipment.
2. The "Grease charge interval" is the total time running full speed.
   You can extend the grease charge interval by operating it under continuous operation or at a slow fixed speed.
   (The mark (※) refers to one year inspections. 8 hours x 20 days x 12 months = 2000 hours.)
3. The "Hours of supply" can vary depending on operating conditions, so please make sure you don't run out.
4. The number in the upper list corresponds to the numbers given in Fig. 5.6.

(2) Grease charge approach

1. Pose the robot as shown in Fig. 5.6
2. Refer to section 5.3.2 "Disassembling covers" in the Reference Manual to find how to take off the covers of the waist and wrist roll axes.
3. Inject grease into the grease nipple with the grease gun. and enter the amount of grease listed in Table 5.6.
4. Reassemble the covers by following the steps in reverse for when you disassembled the covers.

5.3.5 Replacing the backup battery

Since in the E-series, there is a position detector equipped, a position memory with a battery backup is needed in case of power failures..

The controller also uses a battery backup for such things as program memory. The batteries are supplied at the factory at the time they are shipped, but since they are expendable parts, you need to replace them at periodic intervals.

A general yardstick for battery backup is one year, but depending on how you use the robot, the actual time can vary. When remaining battery life is low, a message reading "As soon as battery alarm occur." is displayed as an alarm (alarm number 2300) . When this alarm has occurred, change the batteries for both the robot and controller as soon as possible. Both the controller and the robot use lithium batteries (Type : A6BAT) .

Below, you'll find an explanation of how to change the batteries.

(1) Replacing the batteries on the main robot unit

Fig. 5.7 shows how to replace the batteries for the main robot unit.
Since the RV-E2. RV-E3J don't have a battery cover, 4, it is not necessary to remove one. Also, there are four batteries for the RV-E3J/RV-E3JM.

1. Confirm that the cable between the cable and robot is securely attached.
2. Turn ON the controller power.
   When we replace the batteries, we'll be relying on the power supply from the controller to protect the position data. Consequently, if the cable is not securely attached, or if the controller power is OFF, the position data can lost.
3. Press the Emergency stop button to bring the robot into an emergency stop condition. This is a safety measure, so don't forget to do it.
4. Remove the fixing screws 2 and 3 as shown in Fig. 5.7, and remove the shoulder cover (B) , 1, from the robot.
5. Remove the installation bolt, 5, in section A of the main robot unit, and remove the battery cover, 4.
6. There is a battery cover 4 in the battery holder. Take the old batteries out of the holder and remove the lead connectors.
7. Place the new batteries into the holder and reconnect the lead connector. You should replace all batteries at the same time.
8. Go back to step 4 and 5 and replace the battery cover, 4, and shoulder cover B by following the instructions in reverse order.

⚠ Caution If the connector comes loose. the battery can't work properly, and the encoder's position data could be lost. You will lose the encoder's position data if you don't supply power from the controller to the robot when you are replacing batteries, or if you keep the controller's power OFF.

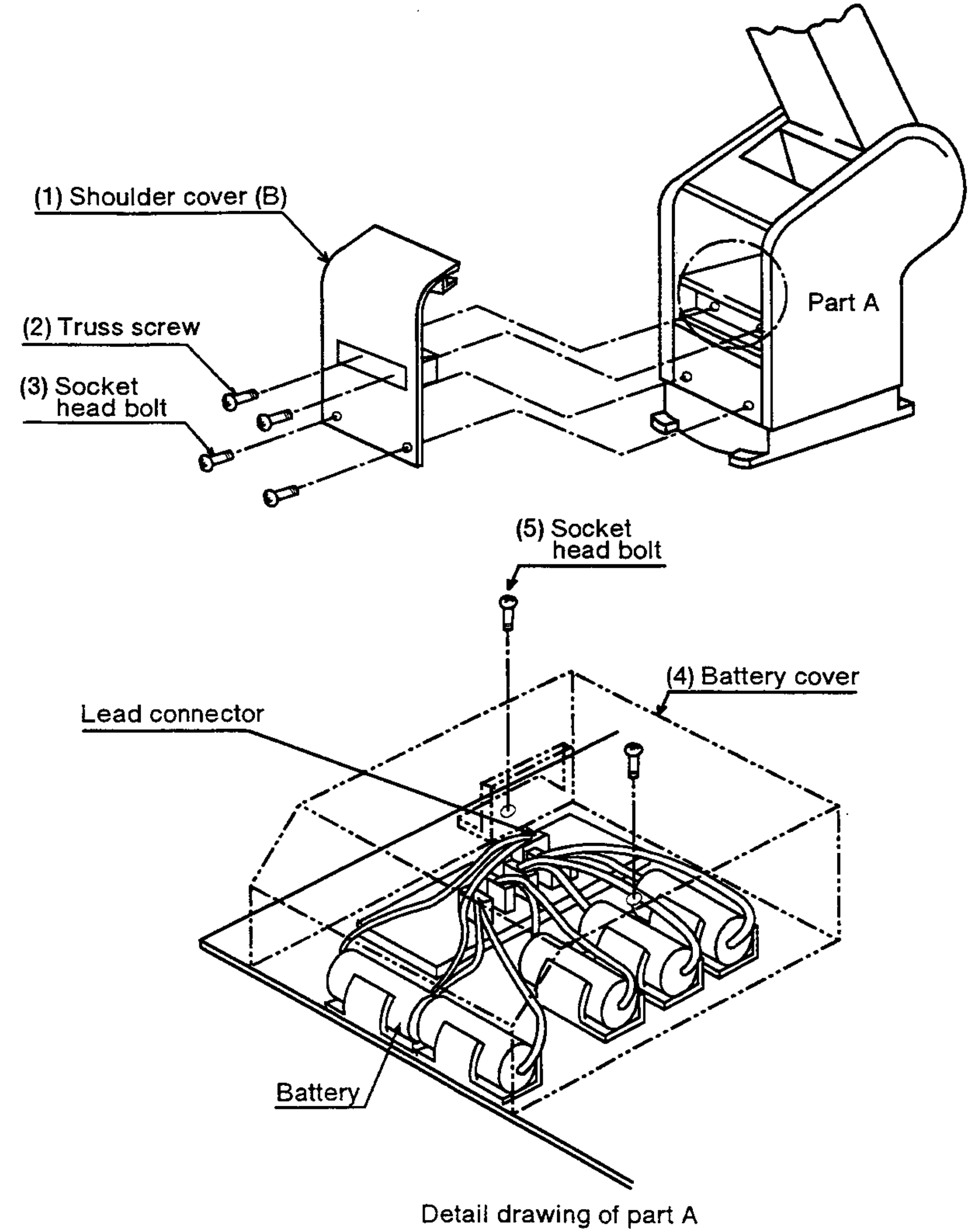


Fig. 5.7 How to replace batteries of the main robot unit

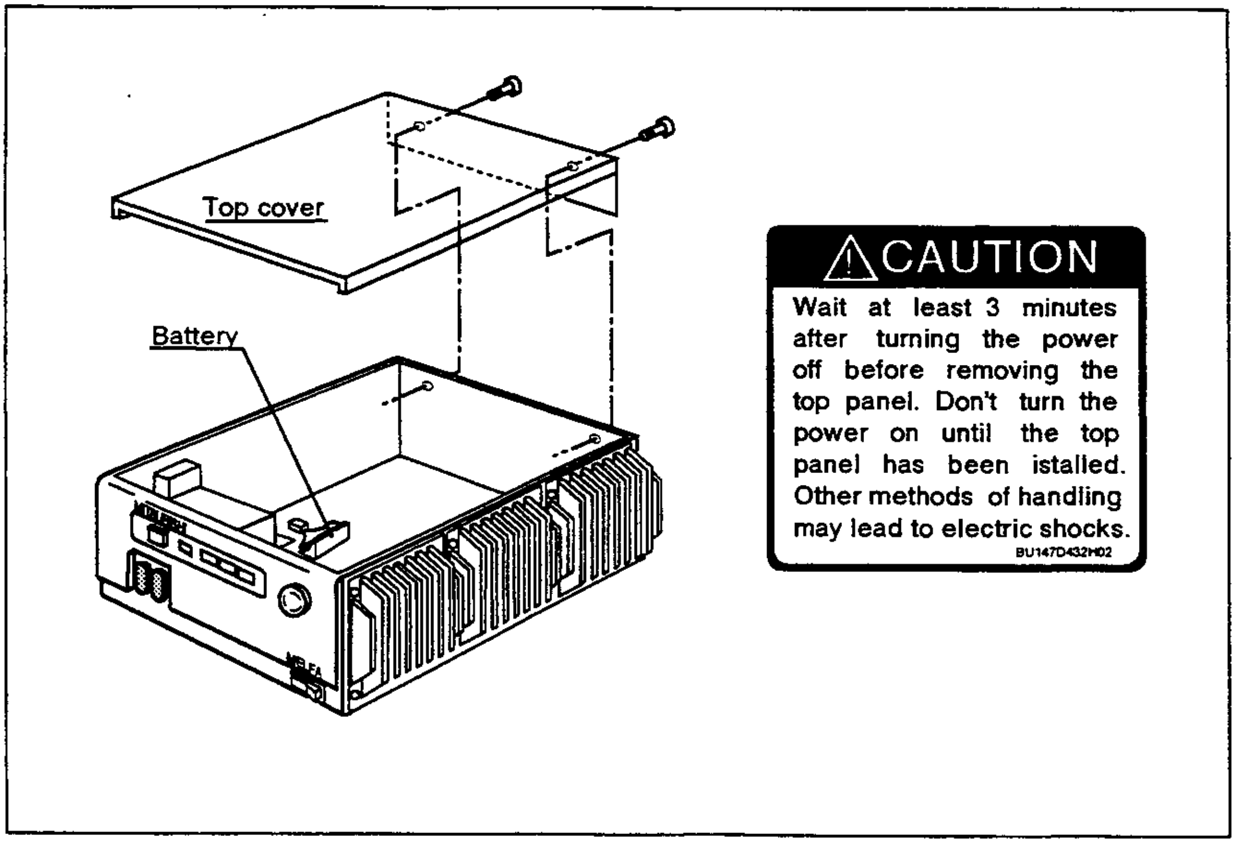
(2) Replacing the batteries for the controller

1. Turn the controller power ON once. (For about 1 minute)
2. Turn OFF the controller power as well as the power source. After at least 3 minutes have elapsed, you can begin to remove the fixing screws and remove the top cover.
3. Remove the old batteries inside the controller.
4. Place new batteries where the old ones used to be so that they fit securely.
5. Pick the cable soket of the new battery and connect it to the connector (CON1BT) on the card so that the red color lead comes to the + (plus) side.
   This entire process should be completed within 15 minutes from the time you remove the batteries.
6. When you are finished, replace the top cover and tighten the cover.

⚠ Caution If you continue to work despite the fact that a battery reserve time alarm has been generated, you will get a backup failure alarm. If you generate a backup failure alarm, the contents of the memory can't be guaranteed, so you should back up important programs and position data beforehand, using personal computer software or other software to save the data onto floppy disks for storage.

(3) Alarm reset

1. Once you have finished replacing the batteries for both the robot and the controller, don't forget to reset the backup failure alarm.
2. The reset operation should be done using the RS command (RS2) in direct execution, from the teaching box or from the personal computer.
   When using direct execution, the follow the written instructions for "creating a program" under section 2.3.2 Programming with the Movemaster command method in this manual, and input " [R] [S] [2] " to execute.
   Refer also to Table 2.14 on the same page, 2.3.2.



⚠CAUTION

Wait at least 3 minutes after turning the power off before removing the top panel. Don't turn the power on until the top panel has been istalled. Other methods of handling may lead to electric shocks.

BU147D432H02

Fig.5.8 How to remove the batteries from the controller

5.3.6 Replacing the fuse

When the controller has blown its fuse, follow the steps below to replace it.

(1) Turn the power button on the bottom right corner of the controller OFF
(2) Turn OFF the power source that supplies the controller.
(3) Turn the fuse holder, 1, on the back of the controller, to the left, and pull out the fuse holder with the fuse. At this time, confirm that the fuse has been blown.
(4) Remove the cause of the blown fuse.
Example : You supplied 200V to the controller, which is specified for only 100V.
(5) Take out the fuse from the holder.
(6) Insert a new fuse into the holder.
(7) Replace the new fuse by following the steps in reverse order in step (3) for the fuse holder, and screw it in securely.

⚠ Caution If it seems that the fuse will blow again, contact your nearest service center.

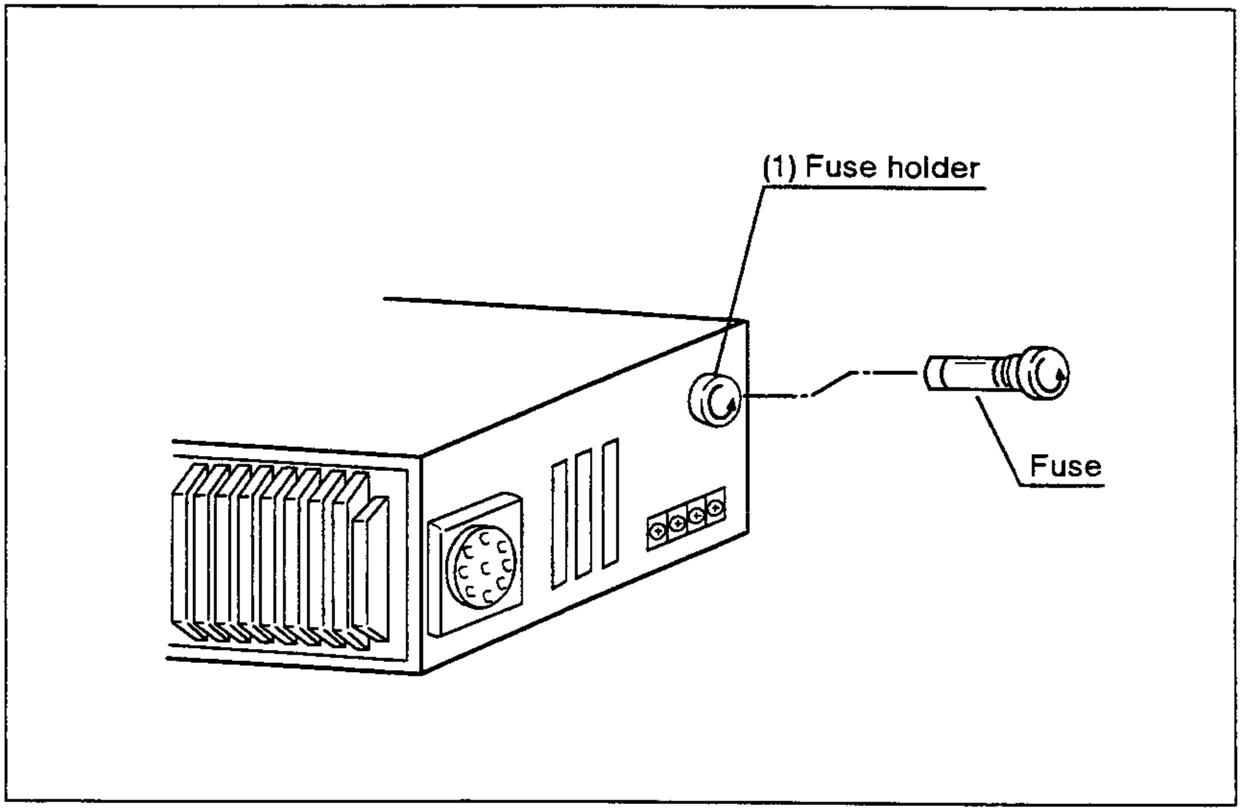


Fig. 5.9 How to replace the fuse

5.4 Maintenance parts

The expendable parts that need to be replaced periodically for this product, are listed in Table 5.7 and Table 5.9. Also, we'll show you spare parts that you will probably eventually need for making repairs in Table 5.8 and Table 5.10. When you purchase parts, buy from recommended manufacturers or from Mitsubishi's service department. Note that the parts designated by Mitsubishi differ from the maker's standard parts.Always confirm the part name, and the robot and controller manufacturing Nos., and purchase the parts from the Mitsubishi Service Center.

(1)Robot consumable parts

Table5.7 Robot consumable part list

Number	Part name	Model type	Supplier (Manufacturer)	Location of use	Quantity
1	Timing belt	S2M-320-U	Mitsuboshi belt	Wrist pitch	1
2	Grease	Gear grease SK-1A	Japan Harmonic Systems	Gear reduction units at various axes	slightly
3	Lithium batteries	A6BAT	Mitsubishi Electric	For the RV-E2/RV-E2M, the shoulder	5
				For the RV-E3J/RV-E3JM, the shoulder	4

(2) Robot spare parts

Table5.8 Robot spare parts list

Number	Part name	Model type (Note)	Supplier (Manufacturer)	Location of use	Quantity
1	AC servo motor		Mitsubishi Electric	Waist, shoulder, elbow	3
2	AC servo motor		Mitsubishi Electric	For the RV-E2/RV-E2M, the twist, pitch, and roll	3
				For the RV-E3J/RV-E3JM, the pitch and roll	2
3	Joint reduction gear		Mitsubishi Electric	Waist, shoulder	2
4	Joint reduction gear		Mitsubishi Electric	Elbow	1
5	Joint reduction gear		Mitsubishi Electric	For the RV-E2/RV-E2M, the twist, and pitch	2
				For the RV-E3J/RV-E3JM, the pitch	1
6	Joint reduction gear		Mitsubishi Electric	Wrist roll	1

Note : Confirm the robot manufacturing No., and contact the Mitsubishi Service Center.

(3) Controller consumable parts

Table5.9 Controller consumable part list

Number	Part name	Model type (Note)	Supplier (Manufacturer)	Location of use	Quantity
1	Lithium batteries	A6BAT	Mitsubishi Electric	Controller, top of CPU card	1

(4) Controller spare parts

Table5.10 Controller spare parts list

Number	Part name	Model type (Note)	Supplier (Manufacturer)	Location of use	Quantity
1	Fuse	MF60NR-8A-05	Toyo Fuse	Controller, back side, inside fuse holder	1

Chapter6 : Appendix.

6.1 Command list.

Position motion control commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
1	Change figure	RV-E2/ RV-E2M CF a[,[R/L] [,[A/B] [,[N/F]]]]	Changes the pose of the robot at position (a).	yes	1 ≦ a ≦ 999 R means Right, L means Left. A means Above, B means Below. N means Non flip, F means Flip.	3-5
		RV-E3J/ RV-E3JM CF a[,[R/L] [,[A/B]]]	Changes the pose of the robot at position (a).	yes	1 ≦ a ≦ 999 R means Right, L means Left. A means Above, B means Below.	3-5
2	Draw joint	RV-E2/ RV-E2M DJ a,b	Moves the joint (a) by the specifed amount (b).	yes	1 ≦ a ≦ 6 Joint interpolation.	3-13
		RV-E3J/ RV-E3JM DJ a,b	Moves the joint (a) by the specifed amount (b).	yes	1 ≦ a ≦ 5 Joint interpolation.	3-13
3	Decrement position	DP	Moves to the previous position in number from current position.	yes	Joint interpolation.	3-15
4	Draw straight	DS [x],[y],[z]	Moves by the specified distance from current position.	yes	Linear interpolation.	3-17
5	Draw	DW [x],[y],[z]	Moves by the specified distance from current position.	yes	Joint interpolation.	3-18
6	Here	HE a	Memorizes current position as the position number (a).	yes	0≦ a ≦ 999 (In the case of 0, the robot memorizes the data as the user defined origin.)	3-34
7	Home	HO [a]	Memorizes current position as the origin.	yes	a=0: Mechanical stopper method a=1: Calibration jig method a=2: User defined origine method	3-36
8	Increment position	IP	Moves to the next position in number from current position.	yes	Joint interpolation.	3-41
9	Joint roll change	JRC <[+]1/-1>	Overwrites the current position by adding +/- 360 to the current joint position of the R-axis	yes	+1 adds 360 -1 subtracts 360	3-42
10	Move approach	MA a1,a2 [,[O/C]]	Moves to a position away from position (a1) by the distance of position (a2).	yes	1 ≦ a1,a2 ≦ 999 Joint interpolation. O means opening a hand. C means closing a hand.	3-46
11	Move continuous	MC a1,a2 [,[O/C]]	Moves from the position (a1) to the position (a2) with linear interpolation.	yes	1 ≦ a1,a2 ≦ 999 | A1-a2 | ≦ 99 O means opening a hand. C means closing a hand.	3-47
12	Move joint	RV-E2/ RV-E2M MJ [w],[s],[e] ,[t],[p],[r]	Turns the joint by the specified angle from the current position.	yes	Joint interpolation.	3-48
		RV-E3J/ RV-E3JM MJ [w],[s],[e] ,[p],[r]	Turns the joint by the specified angle from the current position.	yes	Joint interpolation.	3-48
14	Move	MO a [,[a2]]	Moves to the position (a) .	yes	Joint interpolation.	3-49

Position motion control commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
14	Move position	RV-E2/ RV-E2M MP [x],[y],[z], [a],[b],[c] [,[R/L] [,[A/B] [,[N/F]]]	Moves to the position (x,y,z,a,b,c).	yes	Joint interpolation. R means Right, L means Left. A means A bove, B means Below. N means Non Flip, F means Flip.	3-50
		RV-E3J/ RV-E3JM MP [x],[y],[z], [a],[b] [,[R/L] [,[A/B]]	Moves to the position (x,y,z,a,b).	yes	Joint interpolation. R means Right, L means Left. A means Above, B means Below.	3-50
15	Move play back	RV-E2/ RV-E2M MPB [d],[e] ,[f],[g] ,[h],[i] [,[j] ,[x],[y],[z] ,[a],[b],[c] [,[R/L] [,[A/B] [,[N/F]]] [,[O/C]]	Moves to the specified position with the speed (d), timer(e), output state (f)(g), input state (h)(i), interpolation mode (j).	yes	0 ≦ b ≦32767 0 ≦ e ≦255 0 ≦ f.g,h,i ≦&7FFF J =0(joint) 1(linear) 2(arc) R : Right, L : Left. A : Above, B : Below N: Non Flip F: Flip. O :Opening a hand C :Closing a hand	3-51
		RV-E3J/ RV-E3JM MPB [d],[e] ,[f],[g] ,[h],[i] [,[j] ,[x],[y],[z] ,[a],[b] [,[R/L] [,[A/B]] [,[O/C]]	Moves to the specified position with the speed (d), timer(e), output state (f)(g), input state (h)(i), interpolation mode (j).	yes	0 ≦ b ≦32767 0 ≦ e ≦255 0 ≦ f.g,h,i ≦&7FFF J =0(joint) 1(linear) 2(arc) R : Right, L : Left. A : Above, B : Below O :Opening a hand C :Closing a hand	3-51
16	Move play back continuous	RV-E2/ RV-E2M MPC [d] ,[x],[y],[z] ,[a],[b],[c] [,[R/L] [,[A/B] [,[N/F]]] [,[O/C]]	Moves to the specified position with the specified interpolation method.	yes	D =0(joint) 1(linear) 2(arc) R : Right, L : Left. A : Above, B : Below. N: Non Flip F: Flip. O :Opening a hand C :Closing a hand	3-53
		RV-E3J/ RV-E3JM MPC [d] ,[x],[y],[z] ,[a],[b] [,[R/L] [,[A/B]] [,[O/C]]	Moves to the specified position with the specified interpolation method.	yes	D =0(joint) 1(linear) 2(arc) R : Right, L : Left. A : Above, B : Below. O :Opening a hand C :Closing a hand	3-53
17	Move R	MR a1,a2,a3 [,[O/C]]	Moves on the arc that position (a1)(a2)(a3) determine with circular interpolation.	yes	1 ≦ a1,a2,a3≦999 O :Opening a hand C :Closing a hand	3-54
18	Move RA	MRA a [,[O/C]]	The robot moves on the arc that the previous and the next MRA determine with circular interpolation.	yes	1 ≦ a ≦999 O :Opening a hand C :Closing a hand	3-56
19	Move straight	MS a [,[O/C]]	Moves to the position (a) with linear interpolation.	yes	1 ≦ a ≦999 O :Opening a hand C :Closing a hand	3-57
20	Move tool	MT a,[b] [,[O/C]]	Moves to a position away from position (a) by the distance (b) in the tool direction.(Unit:mm)	yes	1 ≦ a ≦999 Joint interpolation. O :Opening a hand C :Closing a hand	3-58
21	Move tool straight	MTS a,[b] [,[O/C]]	Moves to a position away from position (a) by the distance (b) in the tool direction.(Unit:mm)	yes	1 ≦ a ≦999 Linear interpolation. O :Opening a hand C :Closing a hand	3-59
22	Nest	NT	Moves to the user specified origin position.	yes		3-63
23	Origin	OG	Moves to the user specified origin position.	yes	Joint interpolation.	3-69
24	Override	OVR a	Establishes program over-ride.	yes	1 ≦ a ≦200	3-72

Position motion control commands.

Nɑ	Nomenclature	Input Format	Function	Program	Note	Page
25	Pllet Assign	PA i,j,k	Defines the number of grid points (j,k) in the column and row direction for pallet (i).	yes	1≦i≦9 1≦j,k≦32767	3-73
26	Position clear	PC a1[,a2]]	Clears all position data from position (a1) to (a2).	no	a1≦a2 1≦a1,a2≦999	3-74
27	Position define	RV-E2/ RV-E2M PD a,[x],[y] ,[z],[a] ,[b],[c] [,[R/L] [,[A/B] [,[N/F]]] [,[O/C]]	Substitutes (x,y,z,a,b,c) to position (a).	yes	1 ≦ a ≦ 999 R:Right, L:Left A:Above, B:Below N:Nonflip, F:Flip O:open a hand C:closing a hand	3-75
		RV-E3J/ RV-E3JM PD a,[x],[y] ,[z],[a] ,[b] [,[R/L] [,[A/B]]] [,[O/C]]	Substitutes (x,y,z,a,b) to position (a).	yes	1 ≦ a ≦ 999 R:Right, L:Left A:Above, B:Below O:hand open, C:hand close	3-75
28	Position loading	PL a1,a2	Substitutes coordinates value of position (a2) to position (a1).	yes	1 ≦ a1,a2 ≦ 999	3-77
29	Pallet	PT a	Calculates the coordi-nates of a grid point on pallet (a) and identifies the coordinates as position (a).	yes	1 ≦ a ≦ 999	3-83
30	Pulse wait	PW a	Waits for all axes being positioned within the specified pulses (a).	yes	1 ≦ a ≦ 10000	3-85
31	Position exchange	PX a1,a2	Exchanges the coordinates of position (a1) for those of position (a2).	yes	1 ≦ a1,a2 ≦ 999	3-86
32	Speed define	SD a [,b [,c,d[,e]]]	Defines speed (a), the first order deley time const (b), acceleration time (c), deceleration time (d) and CNT setting (e) in linear interpolation and circular interpolation.	yes	0.1 ≦ a ≦ 650.0 1 ≦ b ≦ 300 0≦ c,b ≦ 2000 e = 0 (disable), over 1(enable).	3-94
33	Shift	SF a1,a2	Shifts the coordinates of position (a1) by the coordinates of position (a2).	yes	1 ≦ a1,a2 ≦ 999	3-96
35	Speed	SP a [,H/L [,b]]	Defines speed (a), acceleration and decele-ration, CNT setting (b).	yes	0 ≦ a ≦ 30 H:Quick motion, L:Smooth motion. B = 0 (disable), over 1(enable).	3-99
36	Timer	TI a	Halts motion for time (a). (Unit :0.1 second)	yes	0 ≦ a ≦ 32767	3-104
37	Tool	TL [a]	Establishes distance (a) between hand flange surface and hand tip.	yes	Unit :mm.	3-105

Program control commands.

Nɑ	Nomenclature	Input Format	Function	Program	Note	Page
38	Counter Load	CL a	Loads internal register value/strings to the specified counter.	yes	1 ≦ a ≦ 99 $1≦ a ≦ $99	3-6
39	Compare counter	CP a	Loads value/strings in counter (a) to the internal register.	yes	1 ≦ a ≦ 99 $1≦ a ≦ $99	3-7
40	Disable Act	DA a	Disables interrupt by the bit (a) of external input terminal.	yes	0 ≦ a ≦ 32767	3-11
41	Decrement Count	DC a	Decrements counter (a) by 1.	yes	1 ≦ a ≦ 99	3-12

Program control commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
42	Delete Line	DL [a1][,[a2] [,[b1][,[b2]]]]	Deletes program lines from (a1) to (a2), or steps from (b1) to (b2).	no	a1 ≦ a2 b1 ≦ b2 1 ≦ a1,a2,b1,b2 ≦ 9999	3-14
43	Enable Act	EA [+/-] a,b [,[c]]	Enables interrupt by the bit (a) of external input terminal, specifies line number (b) to which the program jumps when interrupt occurs, and specifies jumping mode.	yes	0 ≦ a ≦ 32767 +: Bit on, -: Bit off 1 ≦ b ≦ 9999 c=0 or 1 0: Jump, 1: Call subroutine	3-19
44	End	ED	Ends the program.	yes		3-21
45	If Equal	EQ a(or &b) ,c	Causes a jump to line number (c) if internal register value/strings equals (a) (or &b).	yes	-32768≦ a ≦32767 &8000 ≦ b ≦&7fff $1≦ a ≦ $99 1 ≦ c ≦ 9999	3-22
46	Go Sub	GS [a][,[b]]	Call subroutine which starts from line number (a) of program (b)	yes	1 ≦ a ≦ 9999	3-31
47	Go To	GT a	Causes a jump to line number (a).	yes	1 ≦ a ≦ 9999	3-33
48	Halt	HLT	Halts the program.	yes		3-35
49	Increment count	IC a	It increments the counter (a).	yes	1 ≦ a ≦ 99	3-37
50	If Larger	LG a(or &b) ,c	Causes a jump to line number (c)if internal register value/strings is greater than (a) (or &b).	yes	-32768 ≦ a ≦+32767 &8000 ≦ b ≦&7fff $1≦ a ≦ $99 1 ≦ c ≦ 9999	3-43
51	If Not Equal	NE a(or &b) ,c	Causes a jump to line number (c)if internal register value/strings does not equal (a) (or &b).	yes	-32768 ≦ a ≦+32767 &8000 ≦ b ≦&7fff $1≦ a ≦ $99 1 ≦ c ≦ 9999	3-61
52	New	NW	Deletes all lines and positions of the selected program.	no		3-64
53	Next	NX	Specifies the range of a loop in a program executed by command RC.	yes		3-65
54	Repeat Cycle	RC a	Repeats the loop specified by command NX (a) times.	yes	1≦ a ≦32767	3-88
55	Run	RN [a1][,a2 [,b]]]	Executes program (b) from line (a1) to (a2). (a2 not included)	yes	1≦ a1,a2 ≦9999	3-89
56	Return	RT [a]	Completes subroutine activated by command GS and returns to main program.	yes	1≦ a ≦9999 a is the destination.	3-91
57	Set Counter	SC a,[b]	Loads value (b) into counter (a).	○	1 ≦ a ≦ 99 -32768 ≦ b ≦ 32767 &8000 ≦ b ≦ & 7FFF $1 ≦ a ≦ $99 1 ≦ b ≦ 122 (character number)	3-97
58	If Smaller	SM a(or &b), c	Causes a jump to line number (c) if the internal register value is smaller than (a) (or &b).	○	-32768 ≦ a ≦ + 32767 &8000 ≦ b ≦ & 7FFF $1 ≦ a ≦ $99 1 ≦ c ≦ 9999	3-102

Hand control commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
59	Grip Close	GC[a]	Closes hand grip.	yes	a= 0: The 1st hand. 1: The 2nd hand.	3-26
60	Grip Flag	GF a	Defines the open/close state of hand grip used in conjunction with command PD.	yes	a= 0: Open 1: Close	3-27
61	Grip Open	GO[a]	Opens hand grip.	yes	a= 0: The 1st hand. 1: The 2nd hand.	3-28
62	Grip Pressure	GP a1,a2,a3	Defines motor-operated hand gripping force and gripping force retention time.	yes	0≦a1,a2≦63 0≦a3≦99 (Unit :0.1 second.)	3-29

I/O control commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
63	AND	AN a(or &b)	ANDs internal register value and specified value (a) (or &b).	yes	-32768 ≦ a ≦ + 32767 &8000 ≦ b ≦&7FFF	3-4
64	Input Direct	ID [a]	Gets signal from external input without condition.	yes	0 ≦ a ≦ 32767	3-38
66	Output Bit	OB [+/-] a	Sets the output state of bit (a) of external output terminal.	yes	0 ≦ a ≦ 32767 +: Bit on, -:Bit off.	3-66
67	Output Counter	OC a [,[a1] [,[a2]]]	Outputs the counter value (a) to external output terminal unconditionally. (a1: starting bit, a2: bit length)	yes	1 ≦ a ≦ 99 0 ≦ a1 ≦ 32767 1 ≦ a2 ≦ 16	3-67
68	Output Direct.	OD a [,[a1] [,[a2]]]	Outputs data (a) to external output terminal unconditionally. (a1: starting bit, a2: bit length)	yes	-32768 ≦ a ≦ + 32767 &8000 ≦ a ≦&7FFF 0 ≦ a1 ≦ 32767 1 ≦ a 2 ≦ 16	3-68
69	OR	OR a(or & b)	ORs internal register value and specified value (a) (or & b).	yes	-32768 ≦ a ≦ + 32767 &8000 ≦ b ≦&7FFF	3-71
71	Test Bit	TB [+/-] a,b	Causes a jump to line number (b) in accordance with the state of bit (a) of internal register value.	yes	0 ≦ a ≦ 15 +: Bit on, -:Bit off. 1 ≦ b ≦ 9999	3-102
72	Test Bit Direct	TBD [+/-] a,b	Causes a jump to line number (b) in accordance with the state of bit (a) of external input terminal.	yes	0 ≦ a ≦ 32767 +: Bit on, -:Bit off. 1 ≦ b ≦ 9999	3-103
73	EXclusive OR	XO a(or &b)	EXCLUSIVE ORs internal register value and specified value (a)(or &b).	yes	-32768 ≦ a ≦ + 32767 &8000 ≦ b ≦&7FFF	3-109

RS-232C communication commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
74	Counter Read	CR a	Reads out contents of counter (a)	yes	1 ≦ a ≦99 $1 ≦ a ≦$99	3-9
75	Data Read	DR [a]	Reads out data in external output terminal	yes	0 ≦ a ≦32767	3-16
76	Error Read	ER [a]	Reads out the error status.	no	1 ≦ a ≦ 128	3-24
77	LineRead	LR [a]	Reads out program on line number (a).	no	0 ≦ a ≦ 9999	3-45
79	Parameter Read	PMR ["a"]	Reads out the contents of parameter (a).	yes		3-78
80	Position Read	PR [a]	Reads out coordinates of position (a).	yes	0 ≦ a ≦ 999	3-60
81	Question Number	QN [a]	Reads out the selected program number or the information of program (a).	yes	a: program name information: used step number, used position number, used counter number	3-87

RS-232C communication commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
82	Step Read	STR [a]	Reads out the contents of step number (a).	no	0 ≦ a ≦ 9999	3-101
83	Version Read	VR	Reads out the version name of system ROM.	yes		3-106
84	Where	WH	Reads out coordinates of current position.	yes		3-107
85	What Tool	WT	Reads out current tool length.	yes		3-108

Other commands.

No	Nomenclature	Input Format	Function	Program	Note	Page
86	Input	INP a,b,[,[c]]	Reads the value of counter (b) or position (b) or character strings (b) from channel (a).	yes	0 ≦ a ≦ 2 1 ≦ b ≦ 99 1 ≦ b ≦ 999 $1 ≦ b ≦ $99 C=0 : Counter, 1 : Position, 2 : Character	3-39
87	Number	N a	Selects program (a)	no	"a" is a program name.	3-60
88	Open	OPN a,b	Opens I/O (b) for channel (a).	yes	0 ≦ a ≦ 2 b = 1 : Standard RS-232C 2 : Standard RS-422	3-70
89	Parameter Write	PMW "a","b"	Substitutes (b) for the contents of parameter (a).	yes	"a" is a parameter name. "b" is a setting data.	3-79
90	Print	PRN a,b or c	Sends out the contents of counter (a) or position (b) or character string (c) from a personal computer in conjunction with commnand INP	no	"a" is counter value -32768 ≦ a ≦ + 32767 &8000 ≦ a ≦ & 7FFF "b" is coordinates value RV-E2/RV-E2M: (X, Y, Z, A, B,C,R/L, A/B,N/F,O/C) RV-E3J/RV-E3JM: (X,Y,Z,A,B,R/L,A/B, O/C) "c" is character strings 1 ≦ c ≦122 (character number)	8-82
91	Reset	RS [a]	Resets an alarm and program line number.	yes	a =0:(Alarm reset) 1: (Resets all counter data) 2: (Resets battery timer) 3: (Initializes all files) 4: (Resets the origin setting.)	3-90
94	Comment	'	Allows programmer to write a comment following '. (less than 120 characters)	yes		3-110

6.2 Parameter list.

Parameter list.

Parameter	Parameter name	Array number	Contents	Default value	Remarks
Tool coordinates	XTL	6 real number	Sets initial value of tool coordinate system (X, Y, Z, A, B, C). Only Z coordinate is changeable for RV-E3J/RV-R3JM. Unit: mm, mm, mm, deg, deg, deg.	0.0 , 0.0 , 123.0, 0.0 , 0.0 , 0.0	
Base coordinates	XBS	6 real number	Sets initial value of base coordinate system (X, Y, Z, A, B, C). Only X, Y, Z coordinates are changeable for RV-E3J/RV-R3JM. Unit: mm, mm, mm, deg, deg, deg	0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0	
Perpendicular area limit	PAR	6 real number	Sets overrun limits of XYZ coordinate system for minus and plus dirction (-X, +X, -Y, +Y, -Z, +Z). Unit: mm	-10000.0 , 10000.0, -10000.0 , 10000.0, -10000.0 , 10000.0,	Note 1
Joint area limit	JAR	12 real number	Sets overrun limits of each joint. coordinate system for minus and plus dirction (-J1, +J1, -J2, +J2, -J3, +J3, -J4, +J4, -J5, +J5, -J6, +J6). Unit: Deg.	-160.0 , 160.0 , -45.0 , 135.0 , 0.0 , 135.0 , -160.0 , 160.0 , -120.0 , 120.0 , -200.0 , 200.0	RV-E2/ RV-E2M
				-160.0, 160.0, -45.0, 135.0, 0.0, 135.0, -160.0, 160.0, -120.0, 120.0, -200.0, 200.0	RV-E3J/ RV-E3JM
User-defined area	UAR	6 real number	Defines the area where the robot outputs a signal. (-X, +X, -Y, +Y, -Z, +Z)Unit: mm.	1.0 , 0.0 , 1.0, 0.0 ,1.0 , 0.0	Note 2
Automatic program execution	ATP	1 string	Sets program name to be executed automatically when powered on.	" "(No setting).	
Continue Function	CTN	1 integer	Sets whether the robot continues to run from the last execution environment when powered on. (Execution step, Program inner variable, I/O status, etc.) 0: Disable 1: Enable	0	
ON/OFF of buzzer	BZR	1 integer	Sets ON/OFF of buzzer sound. 0 : OFF 1: ON	1	
Automatic operation speed	SPI	1 integer	Sets initial level of automatic operation speed.	12	
External override	EOV	2 real number	Sets initial override of automatic operation. (External override, Program override) Unit: %.	100.0 , 100.0	
Jog speed control	JGJ	3 real number	Sets jog and step operation speed for joint interpolation mode. (Inching, low speed, high speed) Unit: deg, %, %.	0.1 , 1.0 , 13.0	
	JGP	3 real number	Sets jog and step operation speed for linear and circular interpolation mode. (Inching, low speed, high speed) Unit: mm, mm/s, mm/s	0.1 , 1.5 , 100.0	
Continuous motion control	CNT	1 integer	Sets whether robot accelerates and decelerates at taught position when moving. 0: Disable 1: Enable	0	
Accel. and decel. period	ADL	2 real number	Sets acceleration and deceleration periods for reaching to maximum speed. Unit: second	0.2 , 0.2	
Delay time constant	TSR	1 real number	Sets first-order delay time constant. Unit: millisecond.	20.0	

Parameter	Parameter name	Array number	Contents	Default value	Remarks
Pulse width	PWI	1 real number	Sets accuracy of positioning. Unit: pulse.	10000	
Grip control	GCD	1 integer	Sets the initial state of hand operation at power on. (The 1st hand normal / reverse, the 1st hand initial condition, the 2nd hand normal / reverse, the 2nd hand initial condition) Normal / reverses: 0 means normal. 1 means reverse Initial state: 0-3.	0 , 1 , 0 , 1	See GC, GO commands.
The origin point setting permission	HOE	1 integer	Sets enable/ disable of origin point setting. 0: Diseble 1: Enable (Robot can not execute program when set to 1).	0	Note 3
The user-defined origin point	UOG	6 real number	Sets user-defined origin point. (W, S, E, T, P, R) Unit: degree	-160.0, -45.0, 50.0, -160.0, -120.0, -200.0	RV-E2/ RV-E2M
				-160.0, -45.0, 0.0, 0.0, -120.0, -200.0	RV-E3J/ RV-E3JM
Joint order of returning to origin point	UNG	6 real number	Sets joint motion order to return to the origin point.	2 , 1 , 1 , 1 , 2 , 2	Note 3
Wrist angle coordinate selection	RCD	1 integer	Selects wrist angle coordinate system for control and display. 0: General angle (Maintains posture), 1: Joint angle (Varies posture)	0	RV-E3J/ RV-E3JM
Stop input type selection	INB	1 integer	Sets stop input for A type contact or B type contact. 0: A type (Normally open), 1: B type (Normally closed)	0	
Parallel I/O input setting			Sets function for input board.		
	IN1	20 strings	Sets function for first I/O interface board.	PI0, , , , PI1, , , , , , , , , , , ,STA, STP, RST	
	IN2	20 strings	Sets function for second I/O interface board.		
	IN3	20 strings	Sets function for third I/O interface board.		
Parallel I/O output setting			Sets function for output board.		
	OT1	16 trings	Sets function for first I/O interface board.	, , , , , , , , , , , , , RUN, WAI, ERR	
	OT2	16 strings	Sets function for second I/O interface board.		
	OT3	16 strings	Sets function for third I/O interface board.		
Program start mode	PST	1 integer	Sets whether the external program number is read into the robot at startup. 0: No (Prezent program), 1:Yes (Appointed program)	0	
Z phase offset rewrite permission	ADJZ	1 integer	Sets enable / disable of Z phase offset rewrite. 0: Disable, 1: Enable	0	
Z phase offset	OFFZ	6 integers	Memorizes the offset pulses of the origin position from the encoder Z phase. Setting 0 initializes the value.	0, 0, 0, 0, 0, 0	
Communi-cation mode	CMO	7 integers	Sets communication mode of RS232C interface. Baud rate (1: 19200, 2:9600, 3:4800, 4:2400, 5:1200, 6:600), Data bit (7, 8), Parity bit (E:even, O:odd, N:none), Stop bit (1, 2), End code (0:auto, 1:CR+ LF, 2:CR), Com.way (M: no-protocol)	2, 8, E, 2, 0, M	Note 4

Note 1) Default setting does not exceed the limit.

Note 2) Default setting is out of the setting area.

Note 3) It becomes effective just after setting.

Note 4) When set to "0 : auto", the robot sends the same code as the receiving code from the external equipment. (The default setting is "CR".)

6.3 Sample program [1] Teaching playback system

This section explains the teaching method by illustrating an example of what kind of jobs the robot could do with the teaching playback system. We recommend you to draw up programming sheets as described later in order to make your program clear and understandable.

Table 6.1 Programming step

No.	Item	Operation description
1	Work plan	(1) Define the total work and describe the "WORK NAME" (2) If there is conditional branching, divide the above work into several programs. Then make entry of program number and program name. (3) Furthermore, divide the above program into series of independent operation. Then make entry of each operation and its work positions. (4) Check the above sequence again.
2	Cording	(1) Describe condition data corresponding to each step. (Interpolation mode, speed, timer, signal output, signal input) (2) Make entry of temporary number of step.
3	Teaching Check Modify	See "2. Operation" of this manual and "User's Manual" and complete the program gradually. Then check and modify the program if necessary.

6.3.1 Pick & place operation

(1) Work description

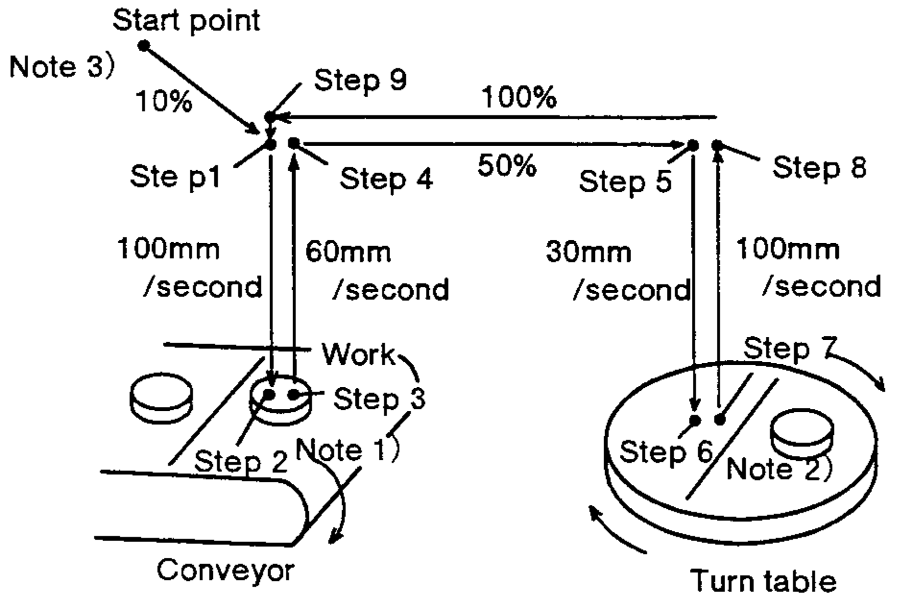
The robot picks up a workpiece on a conveyor with a hand. Then the robot places it on a turn table.

(2) Signal I/O

I/O	Description	Bit
Input	Conveyer in position.	Bit 1
	Turn table in position.	Bit 3
Output	Drives the conveyor to supply the next workpiece.	Bit 1
	Drives the turn table.	Bit 3

(3) Schematic diagram

Figure 6.1 shows description of operation.



Note 1) Step 3 is the condition step which follows step 2.
Note 2) Step 7 is the condition step which follows step 6.
Note 3) The robot moves from the start point at first cycle.

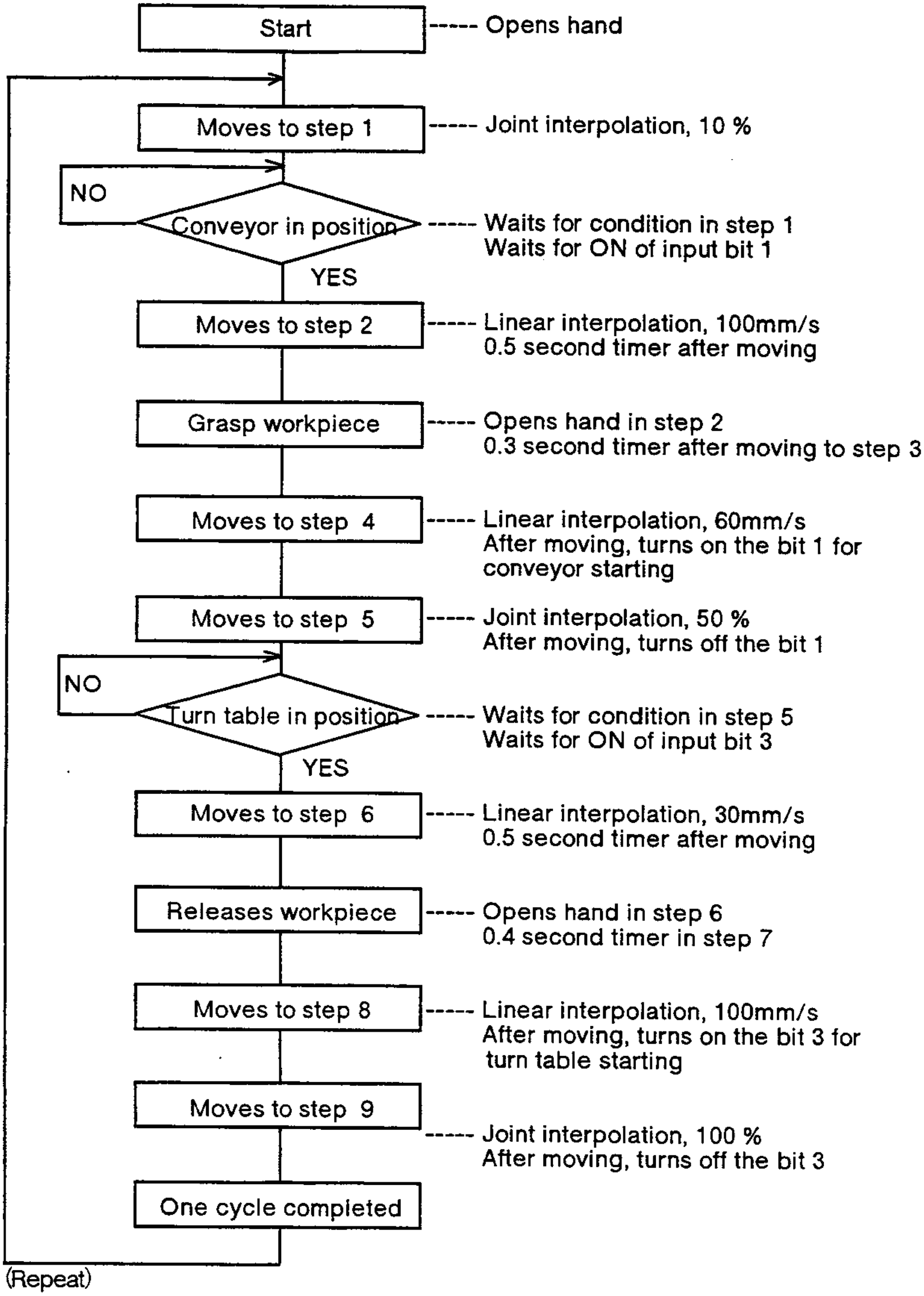
Fig. 6.1 Pick & place operation.

(4) Procedure

(1) The robot waits for a workpiece to finish positioning on a conveyer.
(2) The robot grabs the workpiece and waits for the turn table to finish positioning.
(3) The robot releases the workpiece.

(5) Operation flow

Figure 6.2 shows the flow of operation.



Start
----- Opens hand
Moves to step 1
----- Joint interpolation, 10 %
NO
Conveyor in position
----- Waits for condition in step 1
Waits for ON of input bit 1
YES
Moves to step 2
----- Linear interpolation, 100mm/s
0.5 second timer after moving
Grasp workpiece
----- Opens hand in step 2
0.3 second timer after moving to step 3
Moves to step 4
----- Linear interpolation, 60mm/s
After moving, turns on the bit 1 for conveyor starting
Moves to step 5
----- Joint interpolation, 50 %
After moving, turns off the bit 1
NO
Turn table in position
----- Waits for condition in step 5
Waits for ON of input bit 3
YES
Moves to step 6
----- Linear interpolation, 30mm/s
0.5 second timer after moving
Releases workpiece
----- Opens hand in step 6
0.4 second timer in step 7
Moves to step 8
----- Linear interpolation, 100mm/s
After moving, turns on the bit 3 for turn table starting
Moves to step 9
----- Joint interpolation, 100 %
After moving, turns off the bit 3
One cycle completed
(Repeat)

Fig. 6.2 Flowchart

(6) Programming sheet

Work name : Pick and place
Program name : Pick and place.
Program number: 2

Step number	Operation position (Operation description)	Interpolation	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	Waiting point above conveyer (Robot waits for a workpiece coming.)	Joint	10%	0	Open	—	bit 1 Wait ON
2	Waiting point above conveyer (Robot waits for a workpiece coming.)	Linear	100㎜/s	0.5sec	Close	—	—
3	The same position: Condition step (Robot waits for the hand closed.)	—	—	0.3sec	Close	—	—
4	The position above conveyer (Robot picks up the workpiece.)	Linear	60㎜/s	0	Close	bit 1 ON	—
5	The position above turn table (Robot waits for the turn table in position.)	Joint	50%	0	Close	bit 1 OFF	bit3 Wait ON
6	The position on the turn table (Robot moves down and releases the workpiece.)	Linear	30㎜/s	0.5sec	Open	—	—
7	The same position: Condition step (Robot waits for the hand opened.)	—	—	0.4sec	Open	—	—
8	The position above turn table (Robot moves up.)	Linear	100㎜/s	0	Open	bit 3 ON	—
9	The position above conveyer	Joint	100%	0	Open	bit 3 OFF	—

6.3.2 Handling operation (The robot sorts out defective resistors from non-defective ones.)

(1) Work description

The robot inspects resistors supplied from a conveyer with inspection apparatus. After inspection, the robot puts non-defective registors on another conveyer and throws defective ones into a box.

(2) Signal I/O

I/O	Description	Bit
Input	Completion signal of resistor feeding	Bit 0
	Completion signal of inspection	Bit 1
	OK signal from inspection apparatus	Bit 2
	NG signal from inspection apparatus	Bit 3
Output	Non-defective resistor feeding out signal	Bit 0
	Trigger signal for inspection	Bit 1

(3) Schematic diagram

Figure 6.3 shows the schematic diagram of this work.

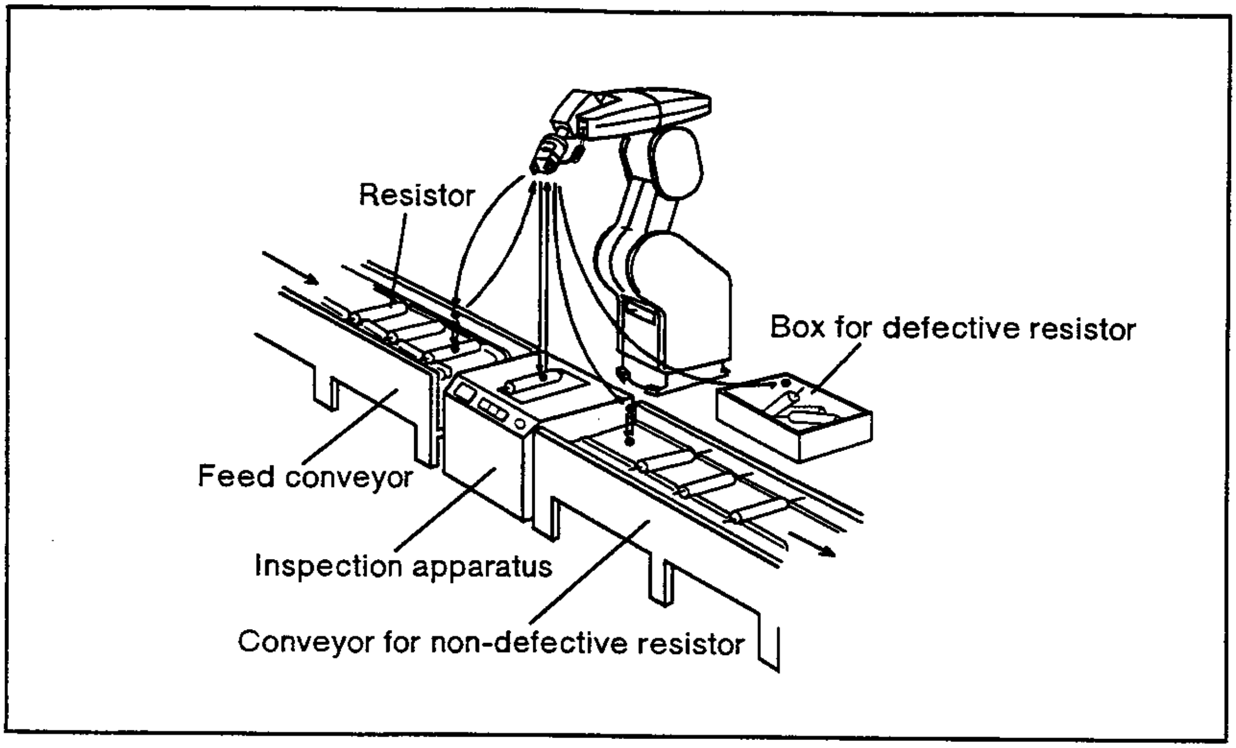

Resistor
Box for defective resistor
Feed conveyor
Inspection apparatus
Conveyor for non-defective resistor

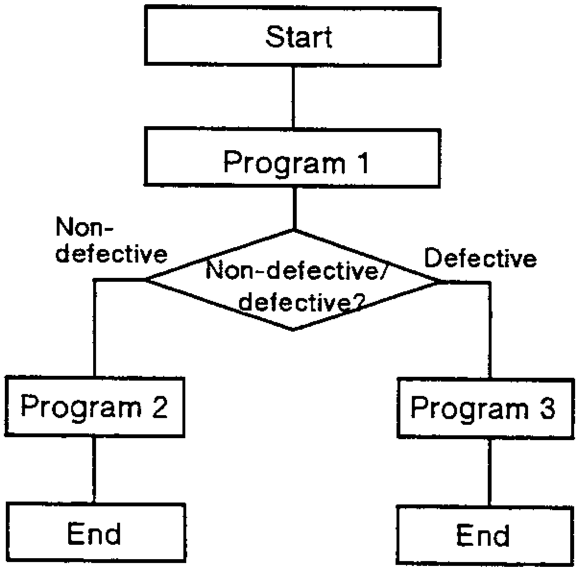
Fig. 6.3 Sorting out registors

(4) Procedure

1) The robot waits for a registor to be fed above a conveyer.
2) The robot carries the fed registor to an inspection apparatus and activates inspection.
3) The robot waits for inspection to finish.
4) If the registor is non-defective, the robot puts it on a conveyer and outputs feeding out signal for 0.5 second. If it is defective, the robot throws it into a box.

(5) Operation flow

1) Simple teaching playback method



In the case of teaching playback method, there is no conditional branching function according to input signal. Therefore, an external PLC should start the program 1, then select the program 2 or 3 according to OK/NG signal of the inspection apparatus.

Program 1
Carries a resistor from feeding location to inspection apparatus.

Program 2
Picks up the resistor from inspection apparatus and puts it on the feeding out conveyor.

Program 3
Picks up the resistor from inspection apparatus and throws it into to a box.

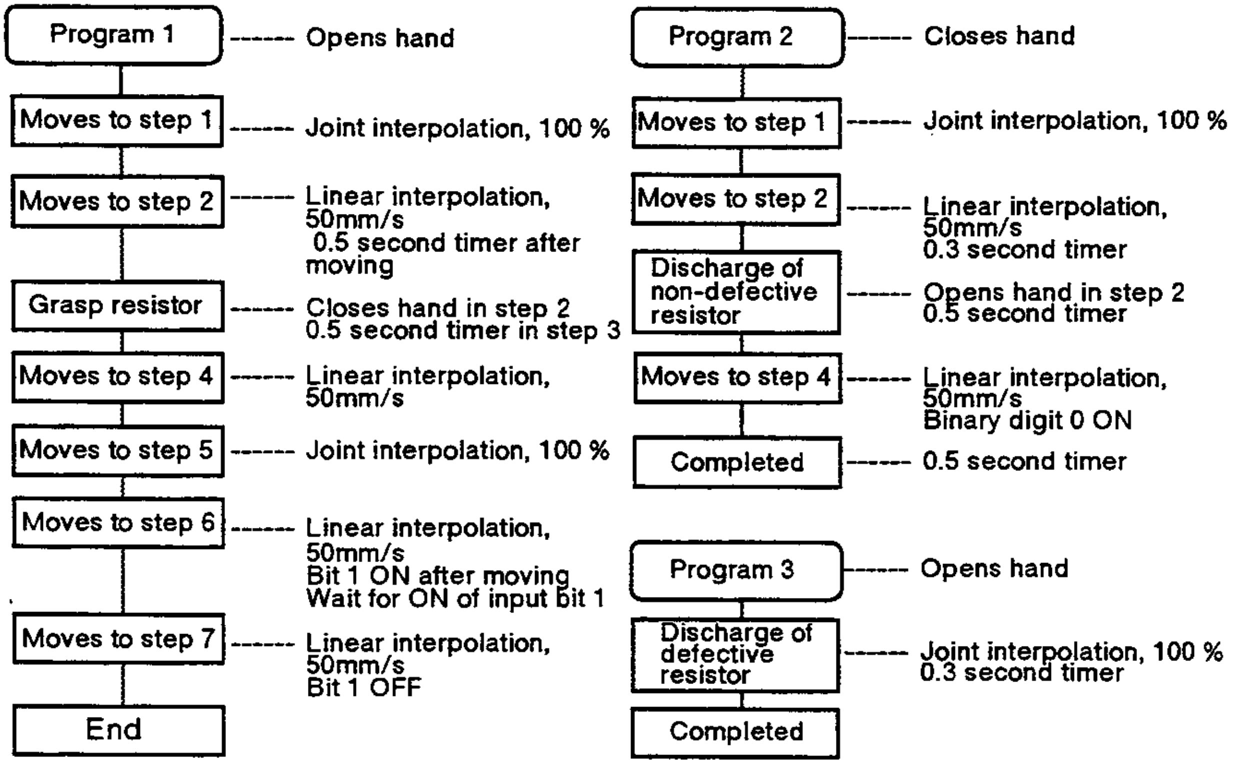

Program 1
Opens hand
Moves to step 1
Joint interpolation, 100 %
Moves to step 2
Linear interpolation, 50mm/s 0.5 second timer after moving
Grasp resistor
Closes hand in step 2 0.5 second timer in step 3
Moves to step 4
Linear interpolation, 50mm/s
Moves to step 5
Joint interpolation, 100 %
Moves to step 6
Linear interpolation, 50mm/s Bit 1 ON after moving Wait for ON of input bit 1
Moves to step 7
Linear interpolation, 50mm/s Bit 1 OFF
End
Program 2
Closes hand
Moves to step 1
Joint interpolation, 100 %
Moves to step 2
Linear interpolation, 50mm/s 0.3 second timer
Discharge of non-defective resistor
Opens hand in step 2 0.5 second timer
Moves to step 4
Linear interpolation, 50mm/s Binary digit 0 ON
Completed
0.5 second timer
Program 3
Opens hand
Discharge of defective resistor
Joint interpolation, 100 % 0.3 second timer
Completed

Fig. 6.4 Flowchart

2) <u>Combination of command method and teaching playback method</u>

The combination of command method and teaching playback method enables conditional selection of programs without PLC. In this case, the robot requires OK/NG signal from inspection apparatus to sort out registors.

‹Program 10› (An example of command program)

```
10  'MAIN PROGRAM......... Means the main program that controls the total flow.
20  GS  , 1            ......... Calls the teaching playback program 1.
                                 (Setting of resistor) .
30  ID                 ......... Inputs signal from the inspection apparatus.
40  TB  +2,100         ......... If it is OK, jumps to the line 100.
50  TB  +3,200         ......... If it is NG, jumps to the line 200.
60  GT 30              ......... Otherwise, jumps to the line 30.
100 GS  , 2            ......... Calls the teaching playback program 2.
                                 (Feeding out non-defective resistors)
110 ED                 ......... END of program.
200 GS  , 3            ......... Calls the teaching playback program 3.
                                 (Throwing the defective registor)
210 ED                 ......... END of program.
```

Note : The first parameter of GS command should be vacant in this example.

(6) Program seat.

Work name : Test of resistor
Program name : Setting of resistor

Program number:1

Step number	Work position (Work description)	Interpolation	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	The position above conveyor (Robot waits for the resistor to come.)	Joint	100%	0	Open	—	bit 0 Wait ON
2	Resistor chucking position (Robot moves down and grasps the resistor.)	Linear	50㎜/s	0.3 sec	Close	—	—
3	The same position: Condition step (Robot waits for the hand closed.)	—	—	0.5 sec	Close	—	—
4	The position above conveyor (Robot picks up the resistor.)	Linear	—	0	Close	—	—
5	The position above inspection apparatus	Joint	—	0	Close	—	—
6	Testing position (Robot turns on the trigger signal and waits till finishing.)	Linear	—	0.5 sec	Close	bit 1 ON	bit 1 Wait ON
7	The position above inspection apparatus (Robot turns off the signal.)	Linear	—	0	Close	bit 1 OFF	—

Work name:Test of resistor
Program name:Feeding out of OK resistor

Program number:2

Step number	Work position (Work description)	Interpolation	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	The position above feeding out conveyor	Joint	100%	0	Close	—	—
2	Feeding out position (Robot moves down and releases the resistor.)	Linear	50㎜/s	0.3sec	Open	—	—
3	The same position: Condition step (Robot waits for the hand opened.)	—	—	0.5sec	Open	—	—
4	The position above feeding out conveyor (Robot turns on the feed out signal.)	Linear	—	0	Open	bit 0 ON	—
5	The same position: Condition step (Robot turns off the feeding out signal.)	—	—	0.5sec	Close	bit 0 OFF	—

Work name:Test of resistor
Program name:Throwing of NG resistor

Program number:2

Step number	Work position (Work description)	Interpolation	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	Throwing position (Robot moves down and releases the resistor.)	Joint	100%	0.3sec	Open	—	—

6.3.3 Assembly operation

(1) Work description
The robot picks up a relay from the relay feeder and fixes it onto the printed wiring board on a conveyer.

(2) Signal I/O.

I/O	Description	Bit
Input	Relay supply completion	Bit 0
	Printed wiring board in position	Bit 1
Output	Relay assembly completion	Bit 1

(3) Schematic diagram
Figure 6.5 shows general description of operation.

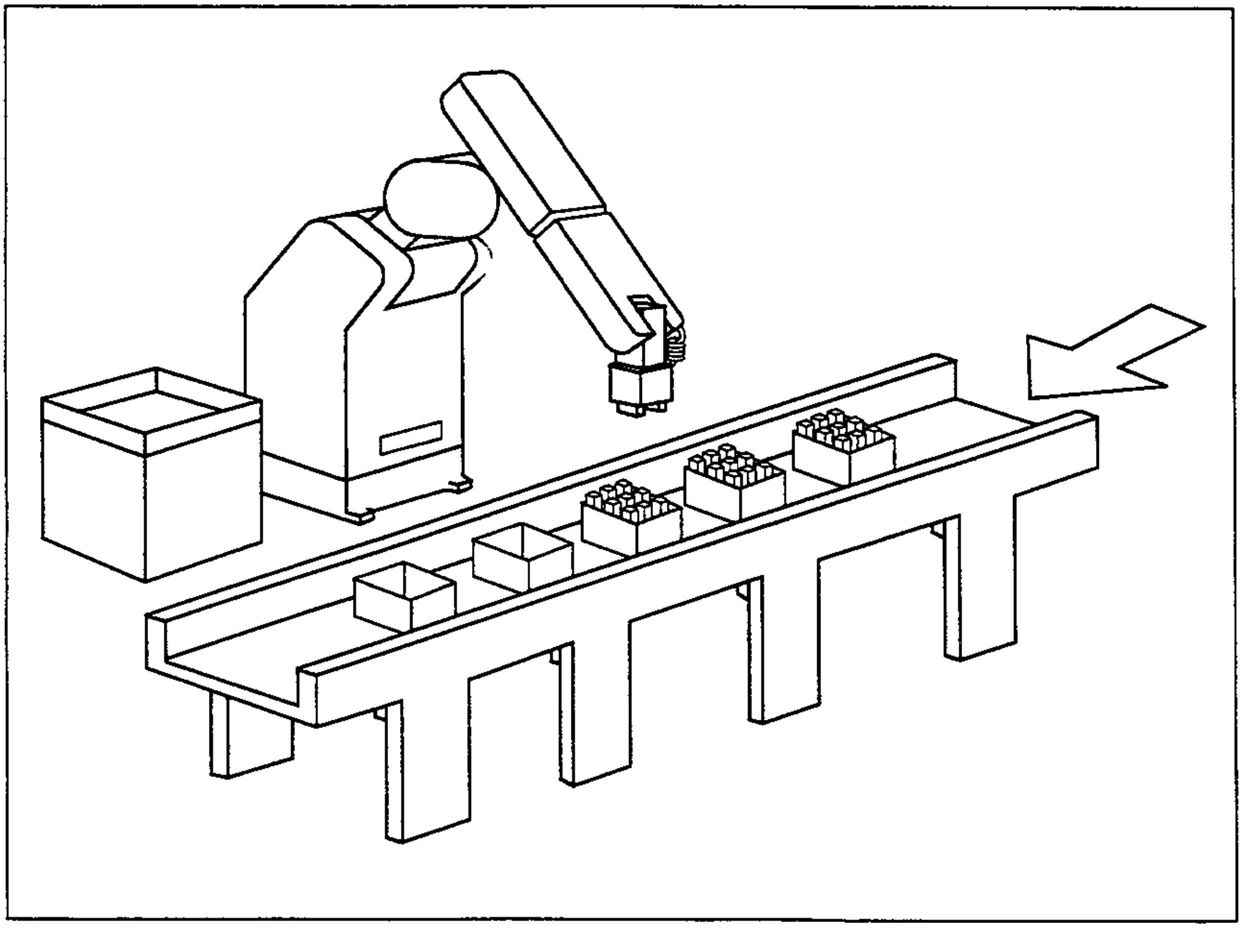


Fig. 6.5 Fixing a relay onto a printed wiring board

(4) Procedure
1) The robot waits for a relay to come from a relay feeder.
2) The robot grasps the relay and waits for a printed wiring board to come above a conveyor.
3) The robot fixes the relay onto the printed wiring board.
4) The robot turns on the completion signal for 0.5 second.

(5) Operation flow

Figure 6.6 shows the flow of operation.

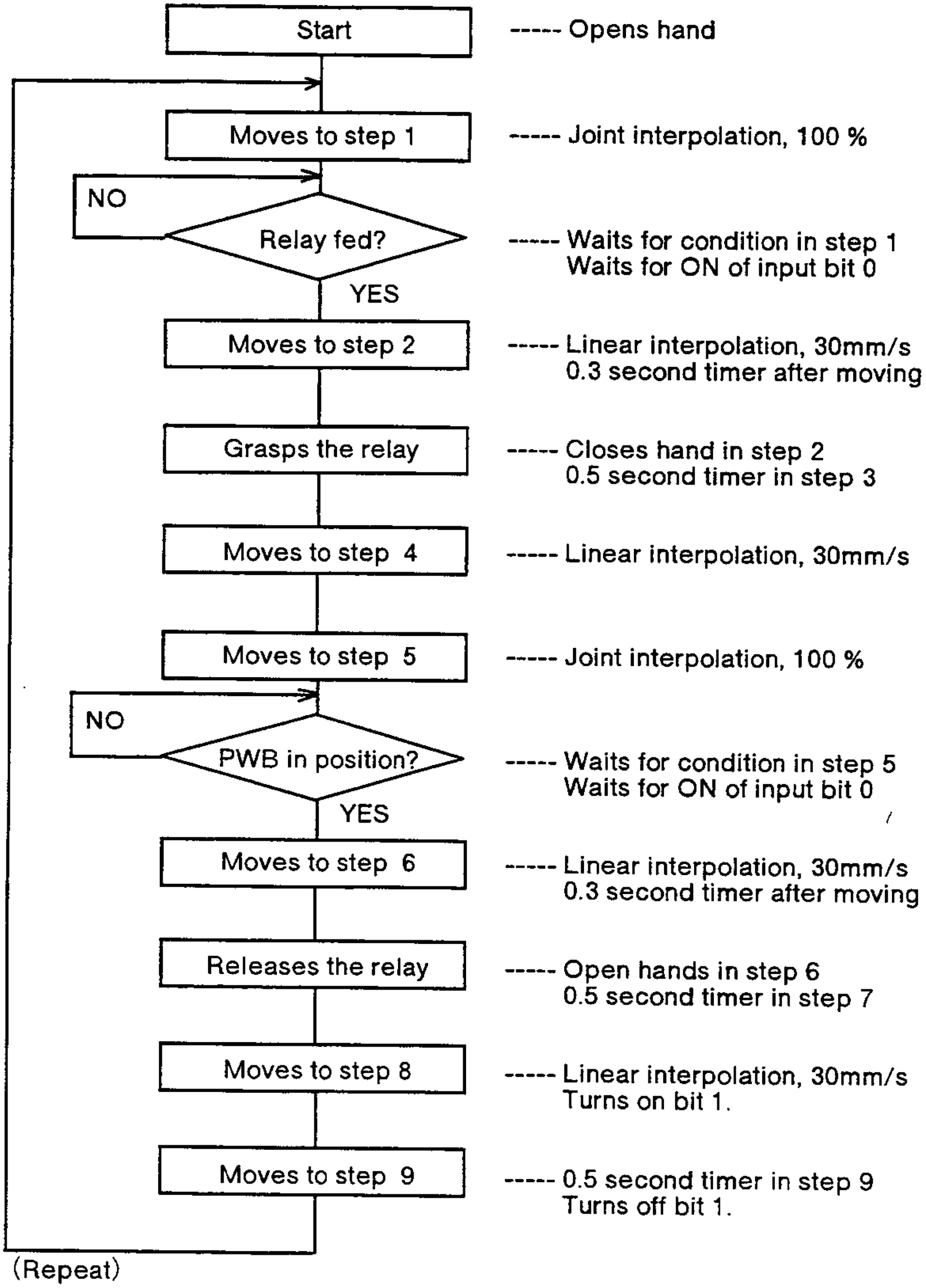

Start
----- Opens hand
Moves to step 1
----- Joint interpolation, 100 %
NO
Relay fed?
----- Waits for condition in step 1
Waits for ON of input bit 0
YES
Moves to step 2
----- Linear interpolation, 30mm/s
0.3 second timer after moving
Grasps the relay
----- Closes hand in step 2
0.5 second timer in step 3
Moves to step 4
----- Linear interpolation, 30mm/s
Moves to step 5
----- Joint interpolation, 100 %
NO
PWB in position?
----- Waits for condition in step 5
Waits for ON of input bit 0
YES
Moves to step 6
----- Linear interpolation, 30mm/s
0.3 second timer after moving
Releases the relay
----- Open hands in step 6
0.5 second timer in step 7
Moves to step 8
----- Linear interpolation, 30mm/s
Turns on bit 1.
Moves to step 9
----- 0.5 second timer in step 9
Turns off bit 1.
(Repeat)

Fig. 6.6 Flowchart.

(6) Programming seat.

Work name:Assembly of relay
Program name:Assembly of relay Program number:1

Step number	Work position (Work description)	Interpola tion	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	Relay chucking position (Robot moves down and grasps the relay.)	Joint	100%	0	Open	—	bit 0 Wait ON
2	Relay chucking position (Robot moves down and grasps the relay.)	Linear	30mm/s	0.3 sec	Close	—	—
3	The same position: Condition step (Robot waits for the hand closed.)	—	—	0.5 sec	Close	—	—
4	The position above relay supply device (Robot picks up the relay.)	Linear	—	0	Close	—	—
5	The position above printed board (Robot waits for the printed board in position.)	Joint	—	0	Close	—	bit 1 Wait ON
6	Relay assembling position (Robot moves down and releases the relay.)	Linear	—	0.3 sec	Open	—	—
7	The same position (Robot waits for the hand opened.)	—	—	0.5 sec	Open	—	—
8	The position above printed board (Robot turns on the relay assembly completion signal.)	Linear	—	0	Open	bit 1 ON	—
9	The same position: Condition step (Robot turns off the relay assembly completion signal.)	—	—	0.5sec	Open	bit 1 OFF	—

6.3.4 Sealing operation

(1) Work description
The robot dispenses seal on a workpiece fed from conveyor.

(2) Signal I/O.

I/O	Description	Bit
Input	Conveyor in position	Bit 3
Output	Starts conveyor (Request for next workpiece)	Bit 3
	Seal ON/OFF	Bit 4

(3) Schematic diagram

Figure 6.7 shows general description of operation.

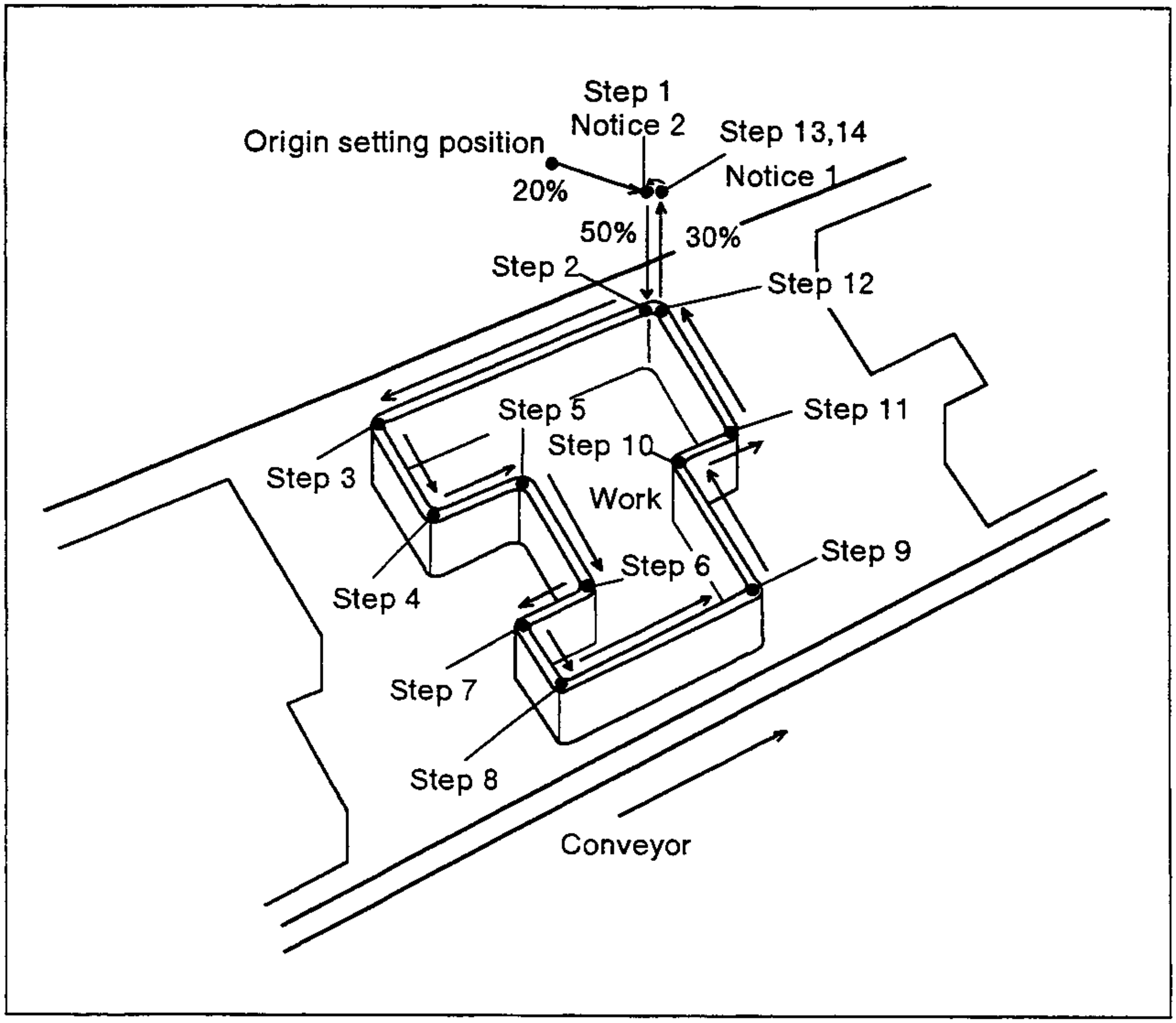


Fig. 6.7 Sealing operation

(4) Parameter control set

As sealing operation needs continuous motion without acceleration nor deceleration, set the parameter of continuous control (CNT) to 1 to make the robot move continuously. Refer "1.3 : Parameter setting" of this manual to set the parameter.

(5) Operation flow

Figure 6.8 shows the flow of operation.

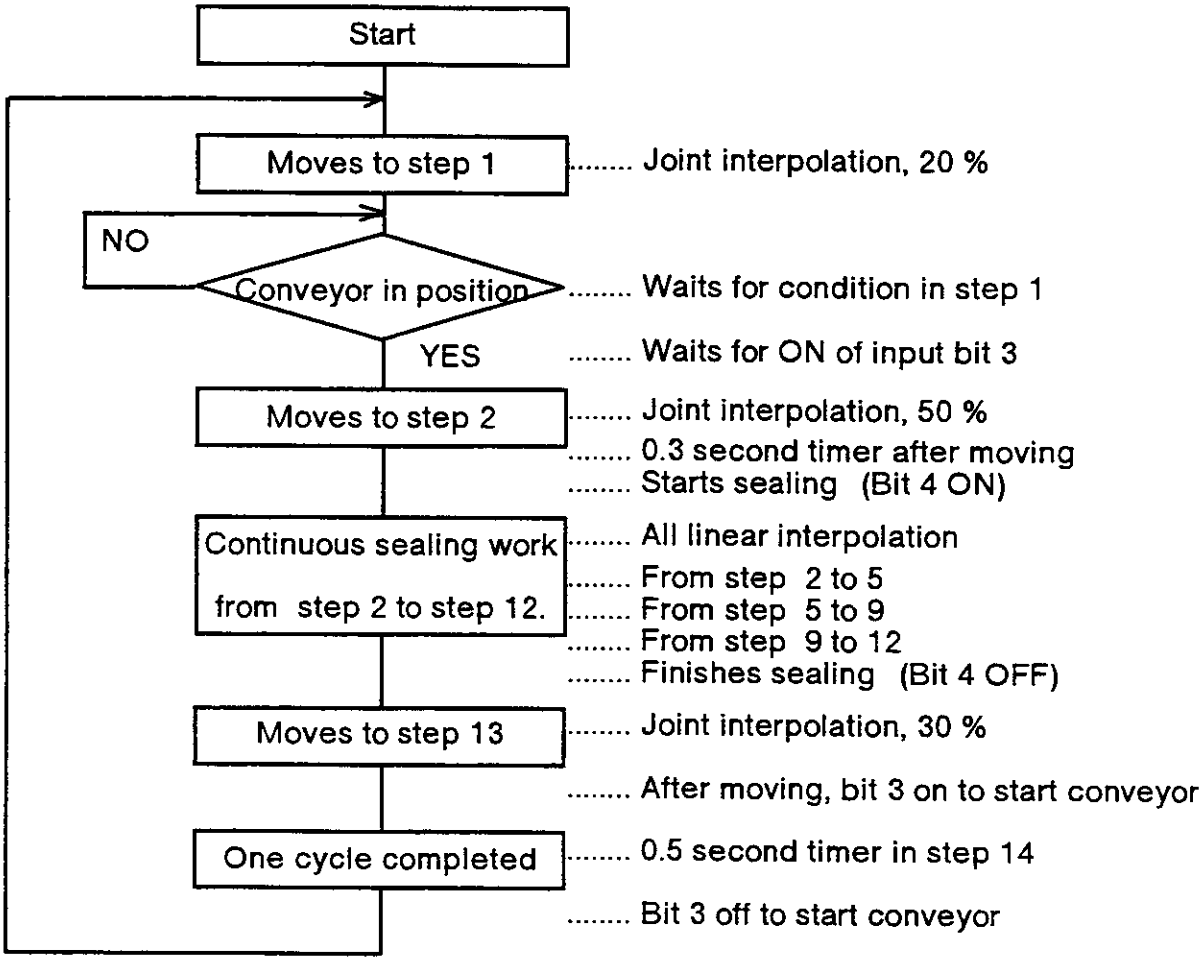

Start
Moves to step 1
Joint interpolation, 20 %
NO
Conveyor in position
Waits for condition in step 1
YES
Waits for ON of input bit 3
Moves to step 2
Joint interpolation, 50 %
0.3 second timer after moving
Starts sealing (Bit 4 ON)
Continuous sealing work from step 2 to step 12.
All linear interpolation
From step 2 to 5
From step 5 to 9
From step 9 to 12
Finishes sealing (Bit 4 OFF)
Moves to step 13
Joint interpolation, 30 %
After moving, bit 3 on to start conveyor
One cycle completed
0.5 second timer in step 14
Bit 3 off to start conveyor

Fig. 6.8 Flowchart.

(6) Programming seat.

Work name : Sealing
Program name : Sealing
Program number:1

Step number	Work position (Work description)	Interpolation	Speed	Timer	Hand Open/ Close	Signal output	Signal input
1	Waiting position above conveyor	Joint	20%	0	—	—	—
2	Sealing start position on workpiece	Joint	50%	0.3 sec	—	bit 4 ON	—
3	Sealing position on workpiece	Linear	30mm/s	0	—	—	—
4	Sealing position on workpiece	Linear	30mm/s	0	—	—	—
5	Sealing position on workpiece	Linear	30mm/s	0	—	—	—
6	Sealing position on workpiece	Linear	35mm/s	0	—	—	—
7	Sealing position on workpiece	Linear	35mm/s	0	—	—	—
8	Sealing position on workpiece	Linear	35mm/s	0	—	—	—
9	Sealing position on workpiece	Linear	35mm/s	0	—	—	—
10	Sealing position on workpiece	Linear	25mm/s	0	—	—	—
11	Sealing position on workpiece	Linear	25mm/s	0	—	—	—
12	Sealing end position on workpiece	Linear	25mm/s	0	—	bit 4 OFF	—
13	Waiting position above conveyor	Joint	30%	0	—	bit 3 ON	—
14	The same position: Condition step	—	—	0.5sec	—	bit 3 OFF	

6.4 Sample program (2) Movemaster command method

This section explains how to make a program with Movemaster command method. Table 6.2 shows effective programming procedure.

Table 6.2 programming procedure

No.	Item	Operation description
1	Work plan	①Assume main flow of work. ②Define the work of the robot.
2	Drawing of flow-chart	①Divide the whole work into separate jobs. ②If there are conditional branchings in the work, divide the work into different programs or divide one program into different blocks. ③Appoint position number at each work position. ④Assign input/output signals. ⑤Based on the above procedure, draw flowchart.
3	Programming	Based on the above flowchart, make the program.

6.4.1 Pick and place operation

(1) Work description

The robot transfers a workpiece from one place to another place.

(2) Defined position

Position No.	Position description	Teaching way
Position 1	Picking position	By actual teaching
Position 2	Placing position	
Position 10	Movement distance above position 1	By numerical value setting
Position 20	Movement distance above position 2	

(3) Operation flow

Figure 6.9 shows the flow of operation

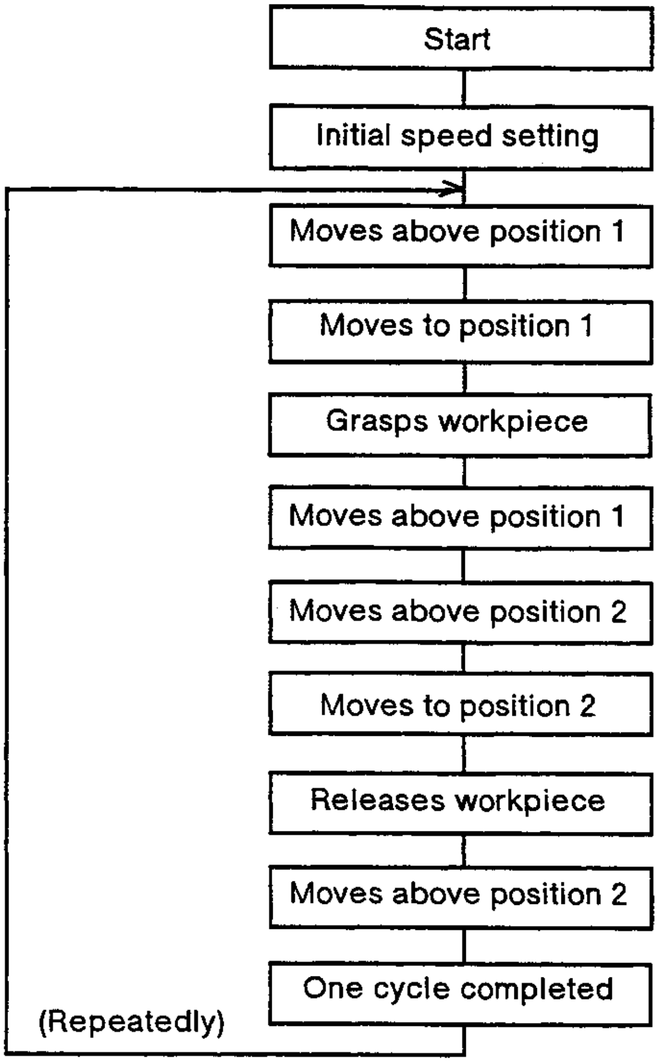


Fig. 6.9 Flowchart.

(4) Schematic diagram

Figure 6.10 shows general description of operation.

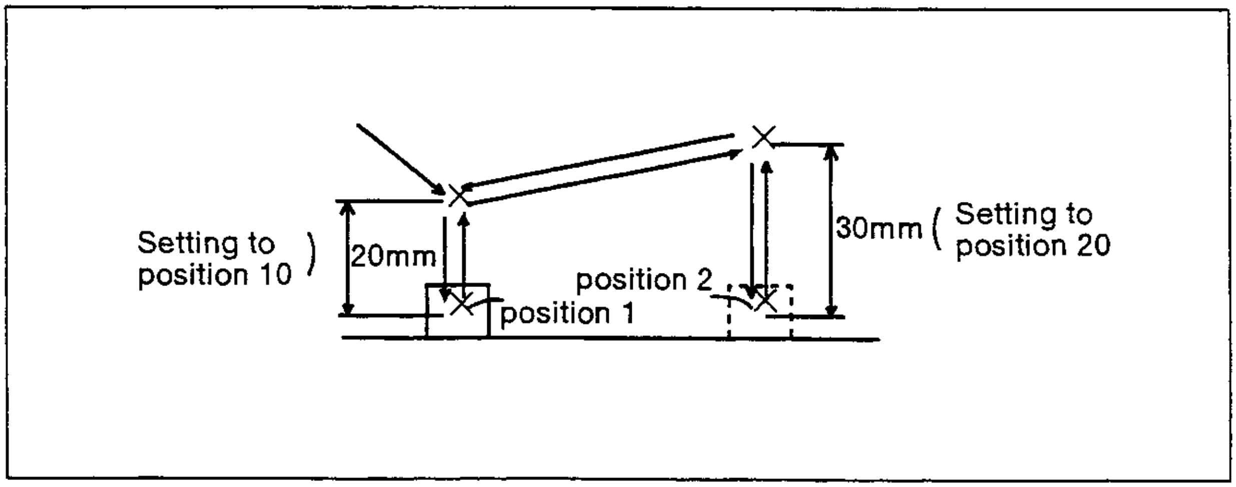


Fig. 6.10 Pick and place operation

(5) Example program

```
 10  PD  10,0,0,20,0,0,0  Defines the aerial distance of travel from position 1 (Z=20
                          mm) .
 20  PD  20,0,0,30,0,0,0  Defines the aerial distance of travel from position 2 (Z=30
                          mm) .
 30  SP  17               Sets the initial speed.
 40  MA  1,10,O           Opens hand and moves 20 mm above the position 1.
 50  MO  1,O              Moves to the position 1.
 60  GC                   Closes hand and grasps the workpiece.
 70  MA  1,10,C           Moves 10 mm above the position 1 with hand closed.
 80  MA  2,20,C           Moves 30 mm above the position 2 with hand closed.
 90  MO  2, C             Puts the workpiece.
100  GO                   Opens hand and releases the workpiece.
110  MA  2,20,O           Moves 30 mm above the position 2 with hand opened.
120  GT  40               Repeats this program.  ( Jumps to the line 40.)
```

6.4.2 Application of interrupt motion

(1) Work description

The robot grasps the workpieces that have different height using a limit switch fixed inside the hand. The robot gets a signal of the limit switch through hand check I/O.

(2) Defined position

Position No.	Position description	Teaching way
Position 1	Position above work	By actual teaching

(3) Signal I/O

I/O	Description	Bit
Input	Work presence signal	Bit 900

(4) Example program

```
        ≀
 90  SP   20          ; Sets speed 20.
100  EA   +900, 140   ; Enables interrupt of bit 900.
110  MO   1, O        ; Moves above a workpiece.
120  DS   0,0,-50     ; Moves 50 mm in the -Z direction.  (Linear interpolation)
130  GT   110         ; Jumps to the line 110 to return to the position 1 as no
                        workpiece has been detected.
140  DA   900         ; Disables interrupt of bit 900.
150  GC               ; Closes hand and it grasps the workpiece.
160  MO   1, C        ; Moves to position 1 with hand closed.
        ≀
```

(5) Explanation.

In this example, the robot moves 50 mm in the -Z direction in line 120. If there is a workpiece, the limit switch signal is input and the robot stopped . Then, the robot jumps to line 140, grasps the workpiece after disabling interrupt and moves to position 1. If there is no workpiece, the robot jumps from line 130 to line 110 returning to position 1. Thus, the robot repeats the same operation again.

(6) Schematic diagram.

Figure 6.11 shows brief description of motion.

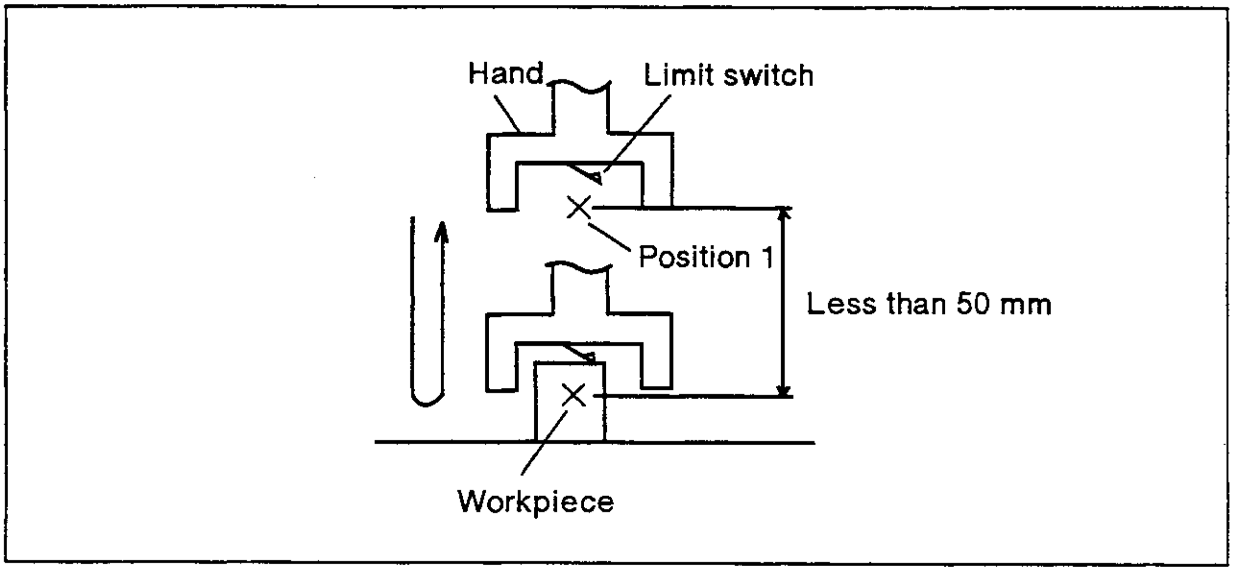


Fig. 6.11 interrupt motion

(7) Operation flow

Figure 6.12 shows the flow of operation.

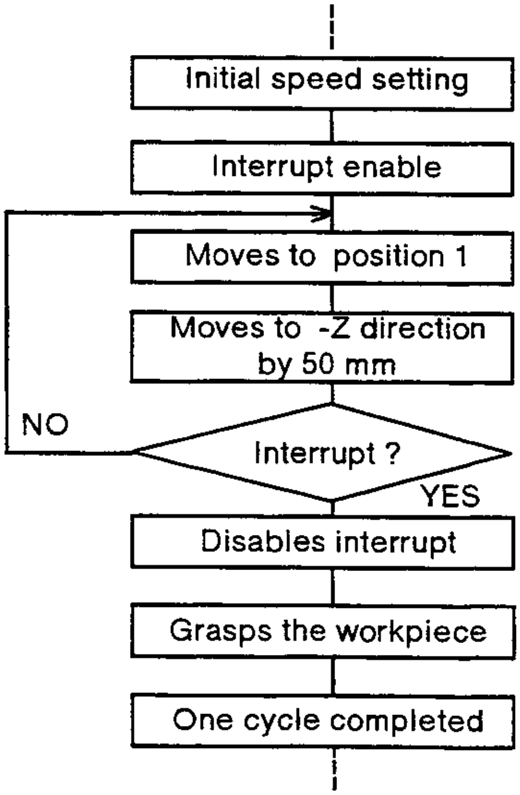


Fig. 6.12 Flowchart

6.4.3 Application of palletizing

(1) Work description

The robot picks up a workpiece from a feeding pallet and places it on an inspection equipment. After inspection, the robot picks up and places it in another pallet. This program assumes that the shapes of the two pallets are different.

(2) Defined position

Position No.	Position description	Teaching way
Position 1	Palette 1 setting position	Defined by PT command
Position 2	Palette 2 setting position	
Position 10	Palette 1 reference position	By actual teaching
Position 11	Palette 1 column terminating position	
Position 12	Palette 1 row terminating position	
Position 13	Palette 1 corner position opposite to reference	
Position 20	Palette 2 reference position	
Position 21	Palette 2 column terminating position	
Position 22	Palette 2 row terminating position	
Position 23	Palette 2 corner position opposite to reference	
Position 30	Test equipment set position	
Position 50	Distance of travel from pallets	

(3) Defined counter.

Counter No.	Description
Counter 11	Palette 1 column counter
Counter 12	Palette 1 row counter
Counter 21	Palette 2 column counter
Counter 22	Palette 2 row counter

(4) Defined input signal.

I/O	Description	Bit
Input	Test completion signal	Bit 7

(5) Example program.

(Initial setup)

```
10   PD  50,0,0,20,0,0,0  ; Defines the aerial distance of travel  (Z=20mm)  as position 50.
                            Note 1 )
15   TL  145              ; Sets tool length at 145mm.
20   GP  10,8,10          ; Sets hand open/close parameters.
25   PA  1,10,6           ; Defines the number of grid points in the column and row
                            directions for pallet 1.  (ver.10 × hor.6)
30   PA  2,15,4           ; Defines the number of grid points in the column and row
                            directions for pallet 2.  (ver.15 × hor.4)
35   SC  11,1             ; Initializes the column counter 11 of the pallet 1.  (Sets 1 to the
                            counter.)
40   SC  12,1             ; Initializes the row counter 12 of the pallet 1.  (Sets 1 to the
                            counter.)
45   SC  21,1             ; Initializes the column counter 11 of the pallet 2.  (Sets 1 to the
                            counter.)
50   SC  22,1             ; Initializes the row counter 12 of the pallet 2.  (Sets 1 to the
                            counter.)
```

(Main routine)

```
100  RC  60               ; Sets the number of repeat cycles of a loop up to line 140.
110  GS  200              ; Calls the subroutine of picking workpieces from the 1st pallet.
120  GS  300              ; Calls the subroutine of setting workpieces on the inspection
                            equipment.
130  GS  400              ; Calls the subroutine of placing workpieces on the 2nd pallet.
140  NX                   ; Returns to line 100.
150  ED                   ; End
```

The following program shows subroutines used in the main program:

(Subroutine : Picking up the workpieces to be tested)

```
200  SP  25               ; Sets speed.
202  PT  1                ; Sets the grid point of the pallet 1 to the position 1.
204  MA  1,50, O          ; Moves to a location above position 1.
206  SP  8                ; Sets speed.
208  MO  1, O             ; Moves to the position 1.
210  GC                   ; Closes hand and grasps the workpiece.
212  MA  1,50, C          ; Moves to a location 20  mm above the position 1 with the
                            workpiece grasped.
214  IC   11              ; Increments the column counter 11 of the  pallet 1 by 1.
216  CP  11               ; Sets the value of counter 11 to the internal register.
218  EQ  11,230           ; Jumps to line 230 on completing the column line.  (compares
                            with value 11.)
220  RT                   ; Ends the subroutine otherwise.
230  SC  11,1             ; Initializes counter 11.  (Sets 1 to the counter.)
232  IC   12              ; Increments the row counter 12 of the pallet 2.
234  RT                   ; Ends the subroutine.
```

(Subroutine : Setting up the workpieces on the test equipment)

```
300  SP  25               ; Sets speed.
302  MT  30,-50, C        ; Moves to a location 50mm ahead of the test equipment.
304  SP  8                ; Sets speed.
306  MO  30, C            ; Sets the workpiece to the inspection equipment.
308  ID                   ; Inputs external data.
310  TB  -7, 308          ; Waits for the test to complete.
312  MT  30,-50, C        ; Moves to a position 50 mm ahead the inspection equipment.
314  RT                   ; Ends the subroutine.
```

(Subroutine : Placing the tested workpiece in pallet 2)

```
400  SP  25               ; Sets speed.
```

```
402 PT  2           ;Sets the grid point of the pallet 2 to the position 2.
404 MA  2,50, C     ;The robot moves to the sky after position 2.
406 SP  8           ;The robot sets speed.
408 MO  2, C        ;The robot moves to the 2nd position.
410 GO              ;The robot opens the hand and it puts the work.
412 MA  2,50, O     ;Moves to a location 20 mm above the position 2.
414 IC  21          ;Increments the column counter 21 of the pallet 2.
416 CP  21          ;Sets the value of the counter 21 to the internal register.
418 EQ  16,430      ;Jumps to line 430 on completing the column line. (compares
                    with value 16.)
420 RT              ;Ends the subroutine otherwise.
430 SC  21,1        ;Initializes counter 21. (Sets 1 into the counter.)
432 IC  22          ;Increments the row counter 22 of the pallet 2.
434 RT              ;Ends the subroutine.
```

Note1) For "RV-E3J/RV-E3JM", the line 10 should be "10 PD 50,0,0,20,0,0".

(6) Explanation.

1) In this example, the robot increments the column counter of each pallet. The robot initializes them to return to the top of column when it reaches the end of the column. And it increments the row counter of each pallet to move to the next column. (From the 214 line to the 232nd lines and from the 414th line to the 432nd line)
2) The robot waits for the test completion signal. (The 310th line) .
3) The completion of the entire sequence is determined by the number of main program cycles. (See line 100)

(7) Schematic diagram.

Figure 6.13 shows general description of operation.

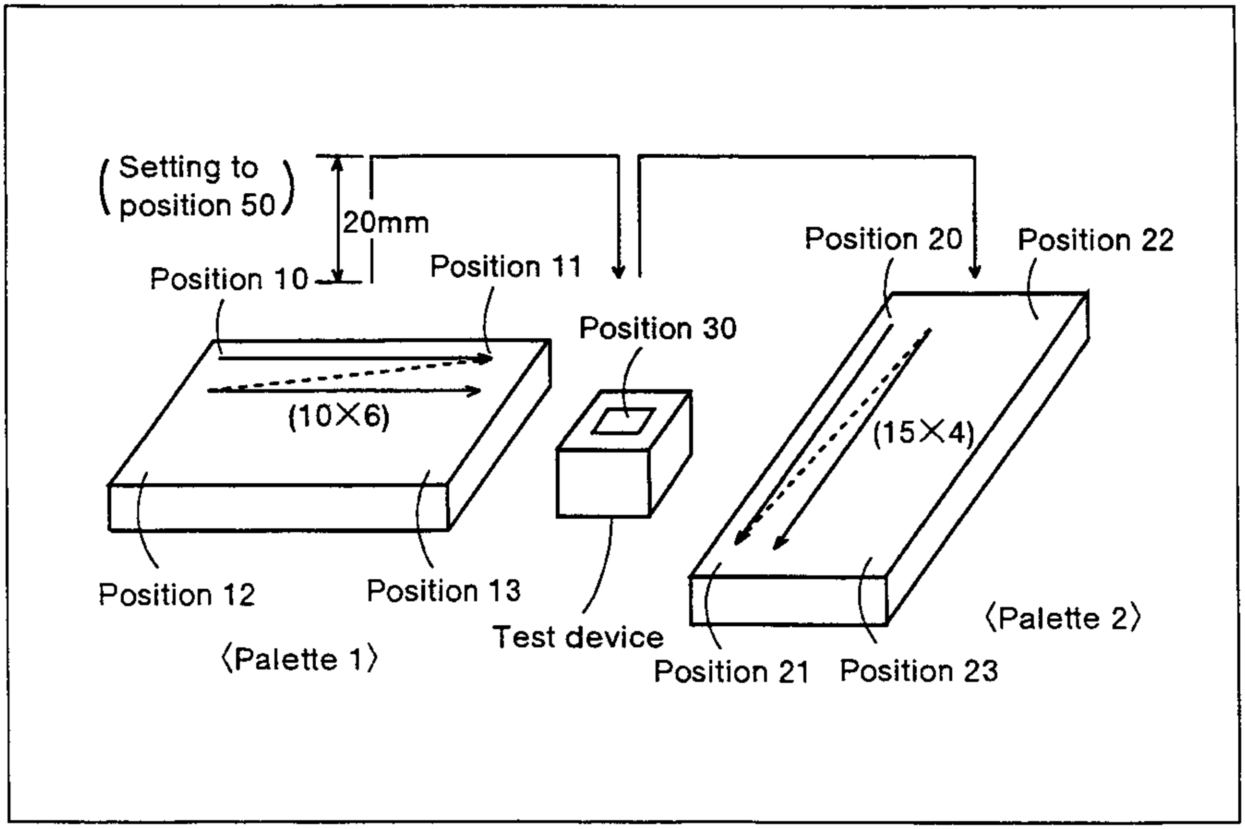


Fig. 6.13 Palletizing

(8) Operation flow

Figure 6.14 shows the flow of operation.

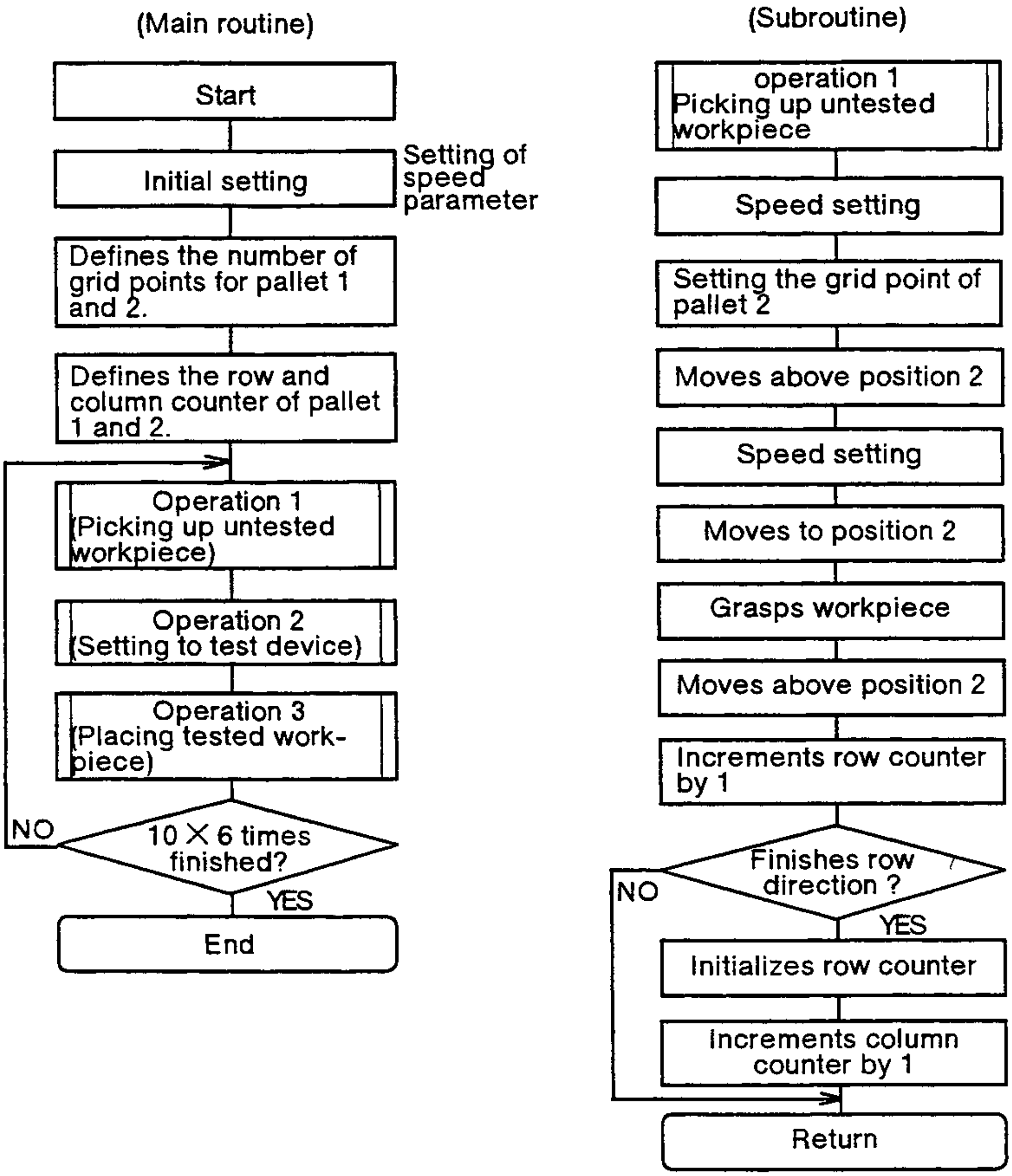

(Main routine)
Start
Initial setting
Setting of speed parameter
Defines the number of grid points for pallet 1 and 2.
Defines the row and column counter of pallet 1 and 2.
Operation 1 (Picking up untested workpiece)
Operation 2 (Setting to test device)
Operation 3 (Placing tested work-piece)
10 × 6 times finished?
NO
YES
End
(Subroutine)
operation 1 Picking up untested workpiece
Speed setting
Setting the grid point of pallet 2
Moves above position 2
Speed setting
Moves to position 2
Grasps workpiece
Moves above position 2
Increments row counter by 1
Finishes row direction ?
NO
YES
Initializes row counter
Increments column counter by 1
Return

Fig. 6.14 Flowchart (Continue)

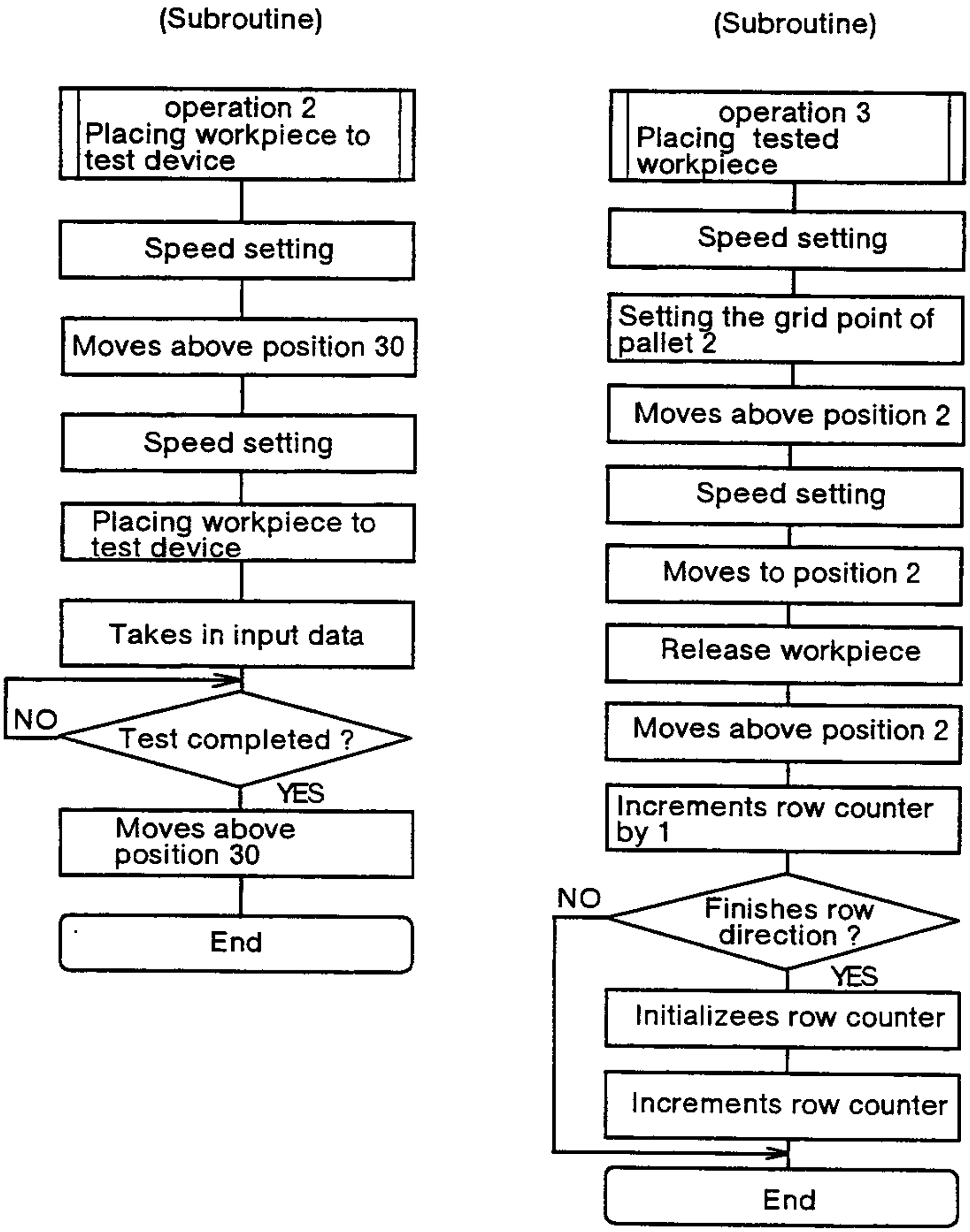


Fig. 6.14 Flowchart

6.4.4 Example of connection with external I/O equipment

(1) Work description
The following program causes the robot to select any of 8 jobs through 8 switches connected to the input for use as external I/O equipment and display the job currently being executed by any of the 8 LEDs connected to the outputs.

(2) Connection.
Figure 6.15 shows connection example.

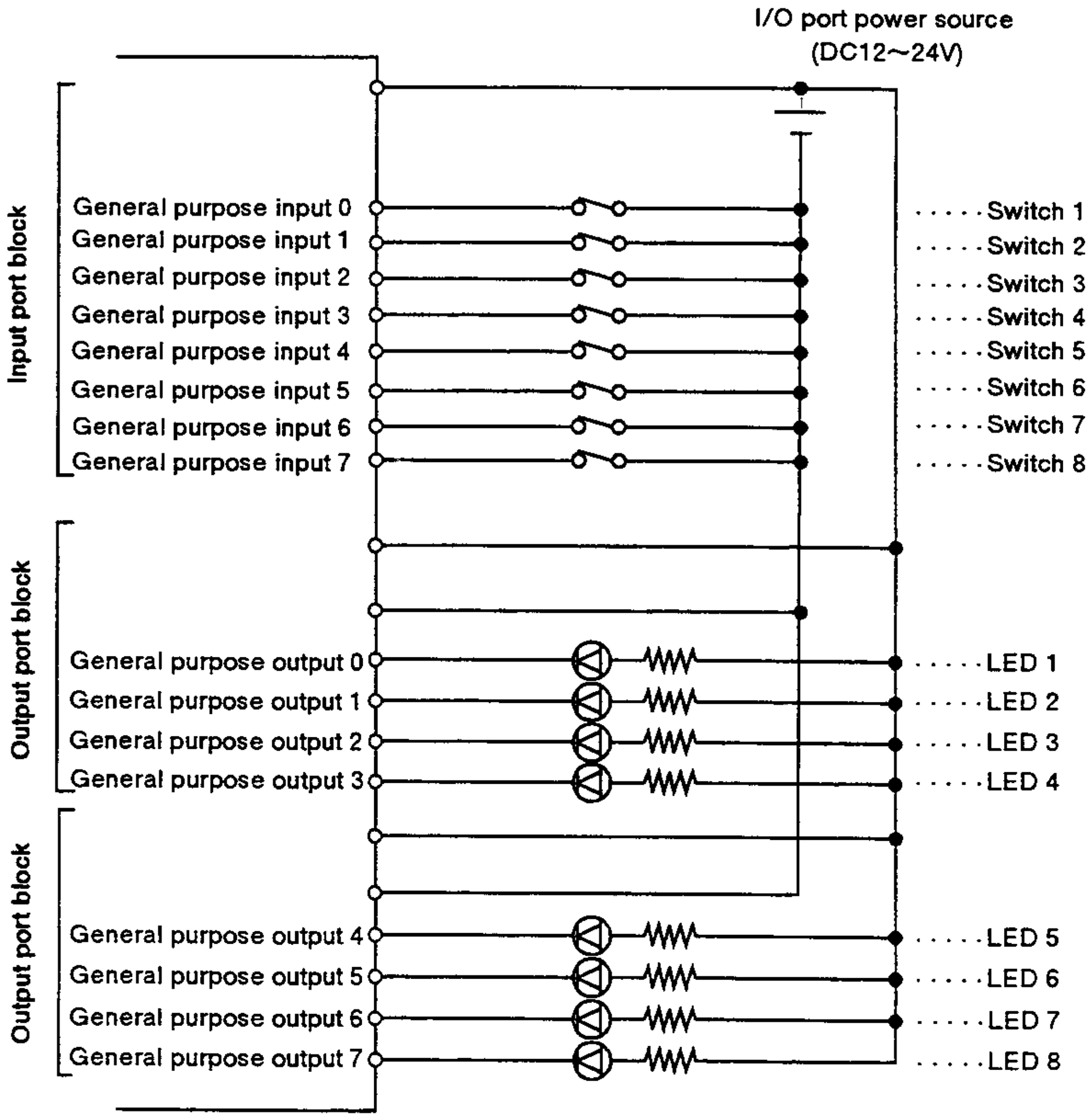
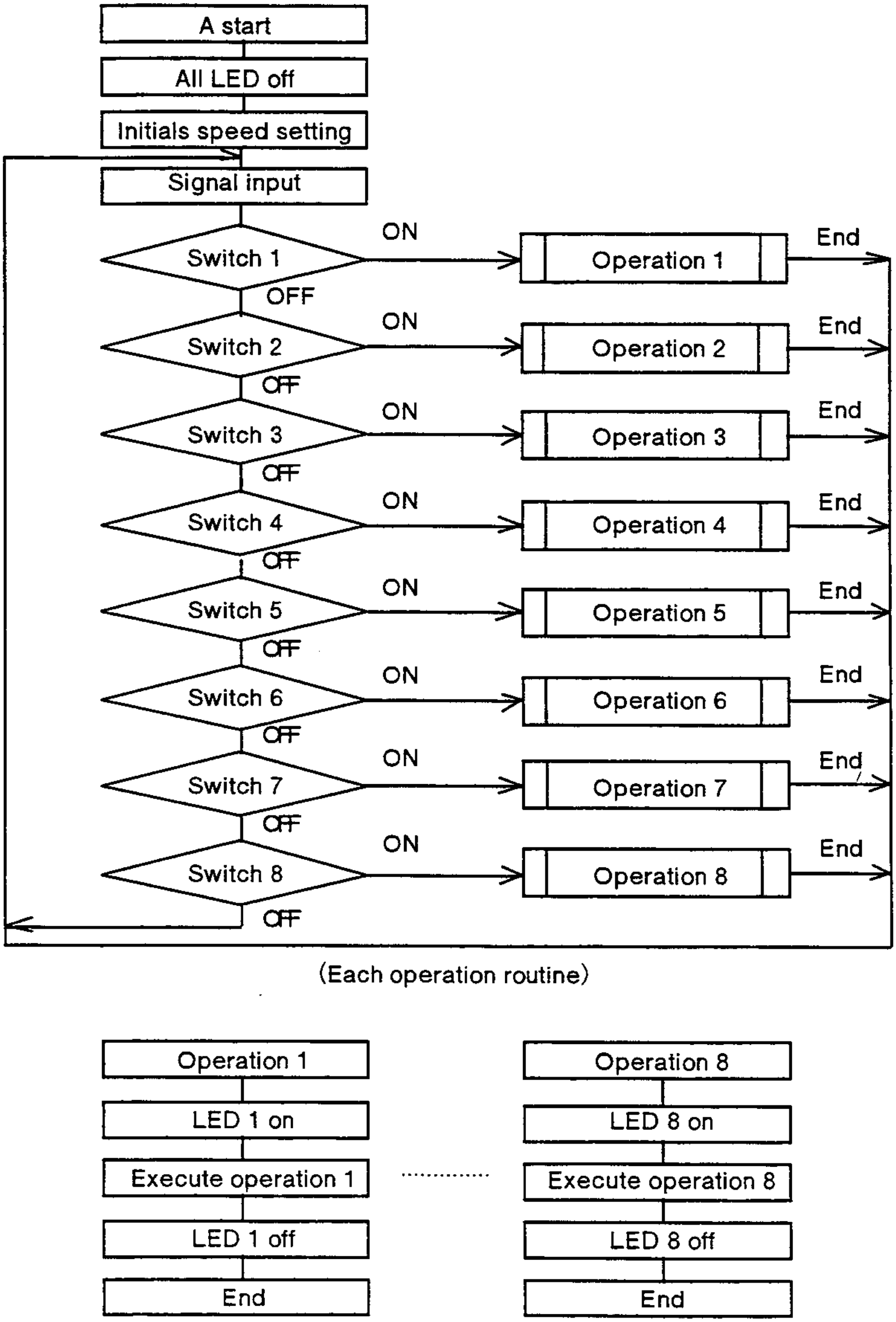


Fig. 6.15 Connection example with external I/O equipment.

(3) Operation flow

Figure 6.16 shows the flow of operation.


(Main routine)

(Each operation routine)

Fig. 6.16 Flowchart

(4) Example program

```
(Main routine)
15   OD  0        ; All LED turns off.
20   SP  10       ; Initializes speed.
25   ID           ; Inputs signals.
30   TB  +0,100   ; Jumps to the line 100 when the switch 1 is turned on.
                                                                  (Job 1)
31   TB  +1,200   ; Jumps to the line 200 when the switch 2 is turned on.
                                                                  (Job 2)
32   TB  +2,300   ; Jumps to the line 300 when the switch 3 is turned on.
                                                                  (Job 3)
33   TB  +3,400   ; Jumps to the line 400 when the switch 4 is turned on.
                                                                  (Job 4)
34   TB  +4,500   ; Jumps to the line 500 when the switch 5 is turned on.
                                                                  (Job 5)
35   TB  +5,600   ; Jumps to the line 600 when the switch 6 is turned on.
                                                                  (Job 6)
36   TB  +6,700   ; Jumps to the line 700 when the switch 7 is turned on.
                                                                  (Job 7)
37   TB  +7,800   ; Jumps to the line 800 when the switch 8 is turned on.
                                                                  (Job 8)
38   GT  25       ; The robot jumps to the line 25 when all switches are off.

(Each operation routine)
100 OB  +0        ; Turns on the 1st LED.
                     (Job  started)                         }
105 MO  10        ; Carries out the 1st operation.          } The 1st operation.
     :                                    :                 }
198 OB  -0        ; Turns off the 1st LED.                  }
                     (Job completed)
199 GT  25        ; Jumps to the line 25.

800 OB  +7        ; Turns on the 8th LED.                   }
                     (Job  started)                         }
805 MO  80        ; Carries out the 8th operation.          } The 8th operation.
     :                                    :                 }
898 OB  -7        ; Turns off the 8th LED.                  }
                     (Job completed)
899 GT  25        ; Jumps to the line 25.
```

6.5 Alarm list.

ALARM NO.		ERROR GENRATING CAUSE AND ITS REMEDY
0100 ~0120	Cause)	Error occurs in the power supply
	Remedy)	After turning off power, turn on the power again.
0200~0266	Cause)	Encoder detection is faulty.
	Remedy)	After turning off power, turn on the power again. If the error still persists, reset the encoder and set origin point.
	0200~0206	Encorder data range over
	0210~0216	Encorder data error
	0220~0226	Encorder data initialization error
	0230~0236	Encorder data comunication error
	0240~0246	Encorder data misscount error
	0250~0256	Encorder data thermal error
	0260~0266	Encorder data circuit error.
	Note)	The lowest digit means joint number.
0300	Cause)	Battery backup error occurs in the controller.
	Remedy)	Check the connection of battery cable to the controller. If the error still persists, change the battery in accordance with the changing procedure.
0311~0316	Cause)	Battery backup error occurs in the encoder. The connection of battery cable is faulty.
	Remedy)	Check the connection of battery cable to the printed board. If the error still persisits, change the battery in accordance with the changing procedure.
	Note)	The lowest digit means joint number.
0400~0450	Cause)	Saved data is invalid. Clear the
	Remedy)	data.
0500~0566	Cause)	Fault is detected in the servo system.
	Remedy)	After turning off power, turn on the power again.
	0500~0506	2-port memory fails.
	0510~0516	Servo memory fails.
	0520~0526	Watch dog time out
	0530~0536	Amplifier/encorder not connected
	0540~0546	Gate array error
	0550~0556	A/D converter unusual
	0560~0566	Over current.
	Note)	The lowest digit means joint number.
0600	Cause)	Use of communication protocol is improper.
	Remedy)	Check the protocol and data of the connected equipment.
0610	Cause)	The communication line is not confirmed.
	Remedy)	Check the connection of cables and the power on of the connected equipment.
0620	Cause)	Data receive error
	Remedy)	Check the connection of cables.
0630	Cause)	Date send error
	Remedy)	Check the connection of cables.
0640	Cause)	The connection of teaching box is not confirmed.
	Remedy)	After turning off power, turn on the power again.
0650	Cause)	The communication channel is not opened yet.
	Remedy)	Open communication line with "OPN" command.
0700	Cause)	Motor-operated hand interface fuse is broken.
	Remedy)	Check the hand cables and replace the fuse.
0710	Cause)	Pneumatic hand interface fuse is broken.
	Remedy)	Check the hand cables and replace the fuse.
0720	Cause)	Parallel I/O interface fuse is broken.
	Remedy)	Check the I/O cables and replace the fuse.

ALARM NO.		ERROR GENERATING CAUSE AND ITS REMEDY
1200	Cause)	Emergency stop signal is input. (external switch)
	Remedy)	Inactivate the emergency stop and reset the error.
1210	Cause)	Emergency stop signal is input. (operating panel)
	Remedy)	Inactivate the emergency stop and reset the error.
1220	Cause)	Emergency stop signal is input. (teaching box)
	Remedy)	Inactivate the emergency stop and reset the error.
1230	Cause)	The teaching box is connected with the EMG. cancel switch of the controller pushed.
	Remedy)	Remove the teaching box when not used. Release the EMG. cancel switch of the controller when using the teaching box.
1400~1456	Cause)	Overload
	Remedy)	Check whether the load capacity exceeds the rating or the robot motion is disturbed by obstacle.
	1400~1406	Over current
	1410~1416	Mechanically locked
	1420~1426	Over speed (command level)
	1430~1436	Over load
	1440~1446	Excessive positioning error
	1450~1456	Over speed (feedback level)
	Note)	The lowest digit means joint number.
1501~1506	Cause)	Can't turn on servo power for no Z phase position is memorized.
	Remedy)	Memorize the Z phase position in accordance with the detecting procedure with the teaching box. The lowest digit means joint number.
1600	Cause)	Can't move in XYZ mode for no origin point is set.
	Remedy)	Set the origin point with the teaching box.
1700~1799	Cause)	Servo paramenter error.
	Remedy)	Correct the parameter.
1800	Cause)	Stop signal or stop switch is active.
	Remedy)	Reset stop state.
2300	Cause)	Warnimg of battery overtime.
	Remedy)	Replace the old battery with new one quickly.
2400	Cause)	Impossible operation while the program is running
	Remedy)	After stopping the program, try it again.
2410	Cause)	Impossible operation while the servo is on
	Remedy)	After turnning off the servo, try it again.
2420	Cause)	Impossible operation while the servo is off
	Remedy)	After turnning on the servo, try it again.
2600	Cause)	The line number assigned for execution does not exist.
	Remedy)	Check the line number in the program, and assign again.
2700	Cause)	The position data assigned for execution dose not exist.
	Remedy)	Check the position date in the program, and assign it again.
2800	Cause)	Syntax error occurs in the command statement.
	Remedy)	Correct the command statement, and input it again.
2810	Cause)	Parameter value of the command is out of range.
	Remedy)	Correct the value.

ALARM NO.	ERROR GENERATING CAUSE AND ITS REMEDY	
2820	Cause)	Parameter value of the INP command is incorrect.
	Remedy)	Correct the command statement, and input it again.
2900	Cause)	Incorrect number used in the command
	Remedy)	Correct the command statement, and input it again.
3400	Cause)	The depth of looping in RC and NX command exceeds 9 levels.
	Remedy)	Set the looping depth within 9 levels.
3500	Cause)	The number of interrrupt commnands exceeds the limit.
	Remedy)	Check the program and delete the interrupt command.
3600	Cause)	Can't calculate correctly.
	Remedy)	Check the program and correct it.
3700	Cause)	Incorrect matching between GS and RT command
	Remedy)	Correct the program.
3710	Cause)	The depth of GS command exceeds 9 levels.
	Remedy)	Set the depth within 9 levels.
3800	Cause)	Assigned pallet is not defined.
	Remedy)	Define the pallet before execution.
3900	Cause)	Device is not valid yet.
	Remedy)	Make the device valid.
3910	Cause)	Upper-ranked device of priority is already valid .
	Remedy)	Make the upper-ranked device invalid.
4501～4506	Cause)	Joint angle exceeds the limit.
	Remedy)	Move the joint into the movable range with joint jog mode. If the
	Note)	error occurs during program operation, modify the position data.
		The lowest digit means joint number.
4511～4516	Cause)	Orthogonal position exceeds the limit.
	Remedy)	Move the joint into the movable range with joint jog mode. If the
	Note)	error occurs during program operation, modify the position data.
		The lowest digit means joint number.
4521～4526	Cause)	User defined area exceeds the limit.
	Remedy)	Move the joint into the movable range with joint jog mode. If the
	Note)	error occurs during program operation, modify the position data.
		The lowest digit means joint number.
4531～4536	Cause)	Robot arm comes into contact with the floor.
	Remedy)	Move the joint into the movable range with joint jog mode. If the
	Note)	error occurs during program operation, modify the position data.
		The lowest digit means joint number.
4541～4546	Cause)	Robot arm comes into contact with the robot itself.
	Remedy)	Move the joint into the movable range with joint jog mode. If the
	Note)	error occurs during program operation, modify the position data.
		The lowest digit means joint number.

ALARM NO.	ERROR GENERATING CAUSE AND ITS REMEDY	
4550	Cause)	Can't calculate joint angle of the arm.
	Remedy)	Move the joint into the movable range with joint jog mode.
4560	Cause)	The position where the robot can't reach is taught.
	Remedy)	Teach the position after moving the robot to the movable range.
4600	Cause)	The position is out of range, or the starting position and the destination position has a different structure flag in linear and circular interpolation.
	Remedy)	Correct the position data or the structure flag.
4700	Cause)	Can't calculate the posture of intermediate position between the start position and the destination position.
	Remedy)	Add the position with posture between the start position and the destination position.
4800~4810	Cause)	Speed is excessive.
	Remedy)	Reduce the assigned speed.
4900	Cause)	Number of program registration exceeds the limit.
	Remedy)	Check the program and delete unnecessary one.
4910	Cause)	Memory area is full. Commands cannot be entered.
	Remedy)	Check the program and delete unnecessary one.
5600	Cause)	Non-existing program is specified.
	Remedy)	Check the file name.
5610	Cause)	Double definition of file name
	Remedy)	Check the file name and delete the excessive one.
5620	Cause)	Write protected
	Remedy)	Specify another file.
5630	Cause)	Read protected
	Remedy)	Specify another file.
5640	Cause)	The file name is not proper.
	Remedy)	Check if the characters and the expression in the program name are proper.
5650	Cause)	Program is faulty.
	Remedy)	Select the proper program.
5700	Cause)	Program not seleted
	Remedy)	Specify the correct program.
5800	Cause)	The step number specified for renumbering is faulty.
	Remedy)	Check the step number.
5810	Cause)	Date assigned for edition does not exist.
	Remedy)	Check the program and operate again.
6700	Cause)	Uses special-purpose output signal for general-purpose output.
	Remedy)	Change the signal assignment of special-purpose output. Or change the program.

ALARM NO.	ERROR GENERATING CAUSE AND REMEDY	
6800	Cause)	Signal parameter assignment of PI0~PI2 is incorrect.
	Remedy)	Table 4.5 in the reference manual / 4.1.4 gives correct commands and parameters.
6900	Cause)	Signal parameter assignment of PO0~PO2 is incorrect.
	Remedy)	Table 4.6 in the reference manual / 4.1.4 gives correct commands and parameters.
7900	Cause)	Power voltage in the main circuit drops.
	Remedy)	Check that power voltage is as specified.
8900~8999	Cause)	Fault occurs in the self-diagnosis of the system software.
	Remedy)	After resetting the alarm, retry the operation.

6.6 Trouble shooting.

These tables explain how to shoot troubles.

Trouble shooting.

A trouble	A confirmation matter	Trouble shooting
Power source is not ready.	(1)Check power plug. (2)Check fuse. (3)Check voltage of power source.	(1)Connect it surely. (2)Change fuse. Reference manual / 5.3.6 explains how to replace it. (3)Supply power source with correct voltage.
Robot does not work.	(1)Check command. (2)Check cable. (3)Check emergency stop switch. (4)Check movable range. (5)Check collision between the robot arm and mechanical stopper or external machines. (6)Check motor power cable. (7)Check an external stop / emergency stop.	(1)Check command name and format. (2)Connect cable surely. (3)Reset emergency stop switch. (4)Move the robot into the movable range. (5)Move the robot to avoid collision. (6)Connect cable surely. (7)Reset stop / emergency switch.
A computer can not input a program.	(1)Check if the Enable/ Disable switch into Enable. (2)Check command (3)Check computer connection cable. (4)Is program stopping?	(1)Set Enable / Disable switch to Disable. (2)Check command name and format. (3)Connect cable surely. (4)Push the reset button to reset program.
Hand dose not work.(Motor hand)	(1)Check parameter of GP command. (2)Check grease of hand.	(1) Set parameter to a suitable value. 2) Charge grease to gears. (Contact our service division).
Hand dose not work. (Pneumatic hand)	(1) Check air supply. (2) Is solenoid valve compatible?	(1) Supply air. (2) Use a compatible solenoid valve.

A trouble	A confirmation matter	Trouble shooting
It stopped during operation.	(1)Check cables. (2)Check excessive load. (3)Check voltage of power source. (4)Check power failure or instantaneous power drop. (5)Check emergency stop switch. (6)Check smell from Motors. (7)Check abnormal sound and excessive vibration.	(1)Replace cables. (2)Reduce load. Specification /5.1.2 Rated load explains it. (3)Use it in voltage limit. (4)Operate it once again. (5)Reset the emergency stop switch and operate it again. (6)There is possibility of motor burned out. Change motor. (7)"Abnormal sound occurs." , "Vibration is large." explains it.
Positioning of accuracy slips off repeatedly.	(1)Check electric noise. (2)Check installation of the robot. (3)Check tension of timing belt. (4)Check installation screws of hand. (5)Check combination of the controller and the robot. (6)Check collision between robot and peripheral during operation.	(1)Reduce noise source. (2)Fasten installing bolts of the robot arm. (3)Adjust tension of timing belt. Reference manual/ 5.3.3 Timing belt. (4)Fasten screws securely. (5)Set origin. (6)Avoid collision.
Abnormal sound develops. Vibration is large.	(1)Check installation bolts of robot arm. (2)Check sound from reduction gears. (Harmoic Drives) Check vibration from reduction gears.	(1)Fasten screws securely. (2)Breakage of reduction gears is expected. Contact our service division.
Output of I/O does not work.	(1)Check voltage and polarity of external power source. (2)Check OD and OB command. (3)Check transistor damage from load short circuit.	(1)Connect the robot with correct voltage source and polarity. (2)Use commands correctly. (3)Contact our service division.
Robot worked abnormally.	(1)Check alarm. (2)Check if the robot arm hangs down just after power is ON. (3)Check which joint works abnormally.	(1)Contact our service division.

6.7 Structure flag.

6.7.1 What is a structure flag?.

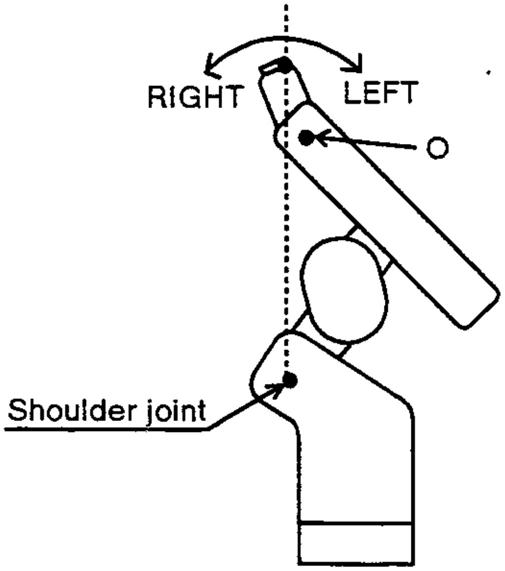
It is a flag that represents pose of robot. The robot decides the pose of the tip with position data of X, Y, Z, A, B, (C) coordinates. But there are complementary poses that robot can take with the same position data. The robot identifies these poses with this flag information.

6.7.2 Definition of each structure flag.

(1) RIGHT/LEFT.

RV-E2/RV-E2M

This flag shows the position of the wrist pitch joint center (O) from the vertical axis through the shoulder joint.



RV-E3J/RV-E3JM

This flag shows the position of the tool center (O) from the vertical axis through the shoulder joint.

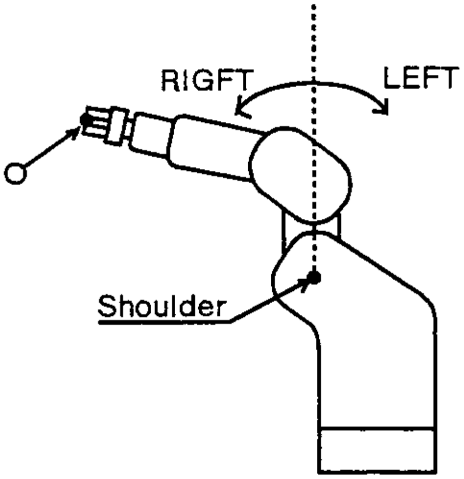


Fig. 6.17 RIGHT/LEFT.

(2) ABOVE/BELOW.

This flag shows the position of the wrist roll joint center [O] from the axis through the shoulder joint and elbow joint.

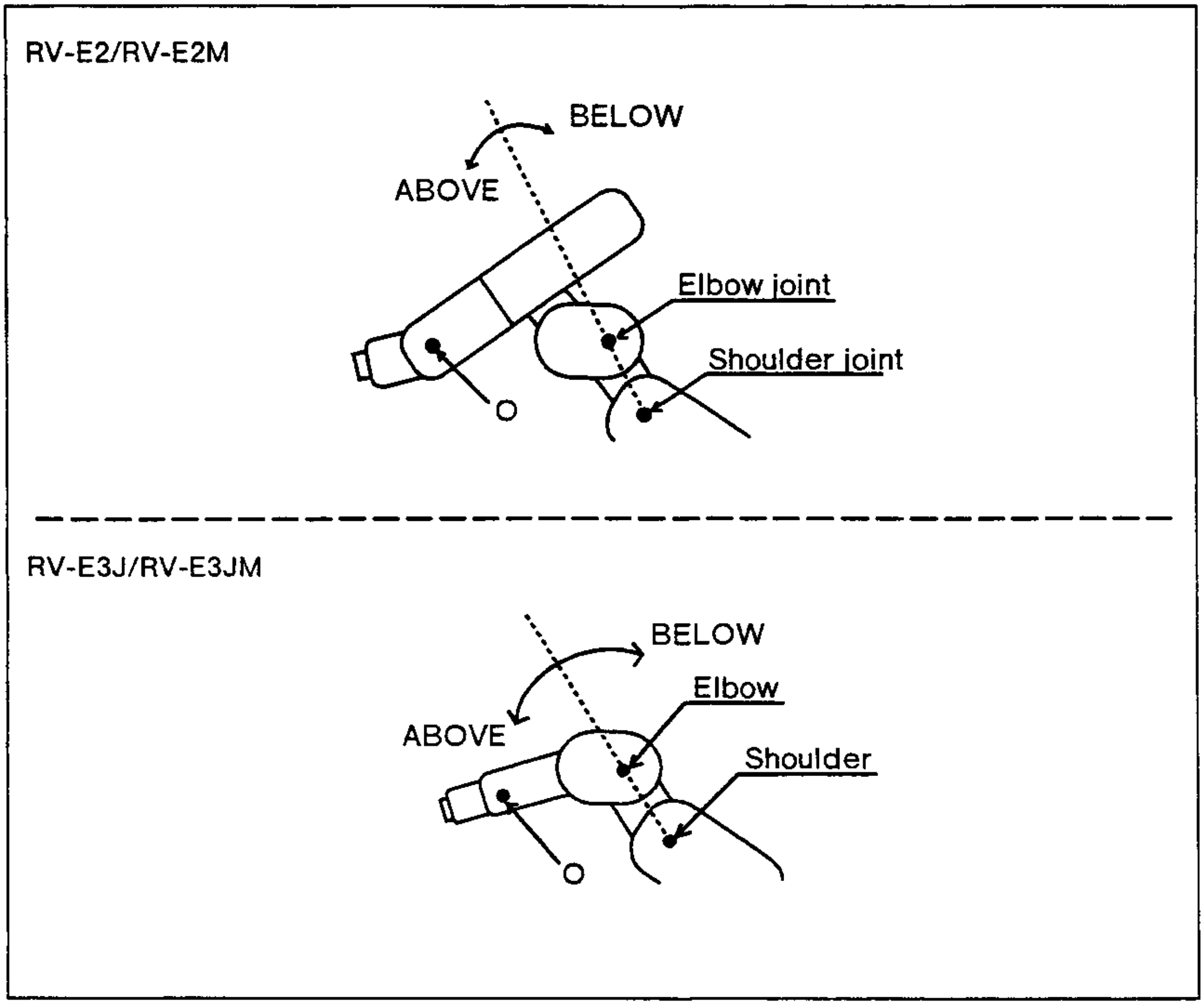


Fig. 6.18 ABOVE/BELOW.

(3) NONFLIP / FLIP (RV-E2/RV-E2M only.)

This flag shows the direction of the mechanical interface from the axis through the wrist twist joint and wrist pitch joint.

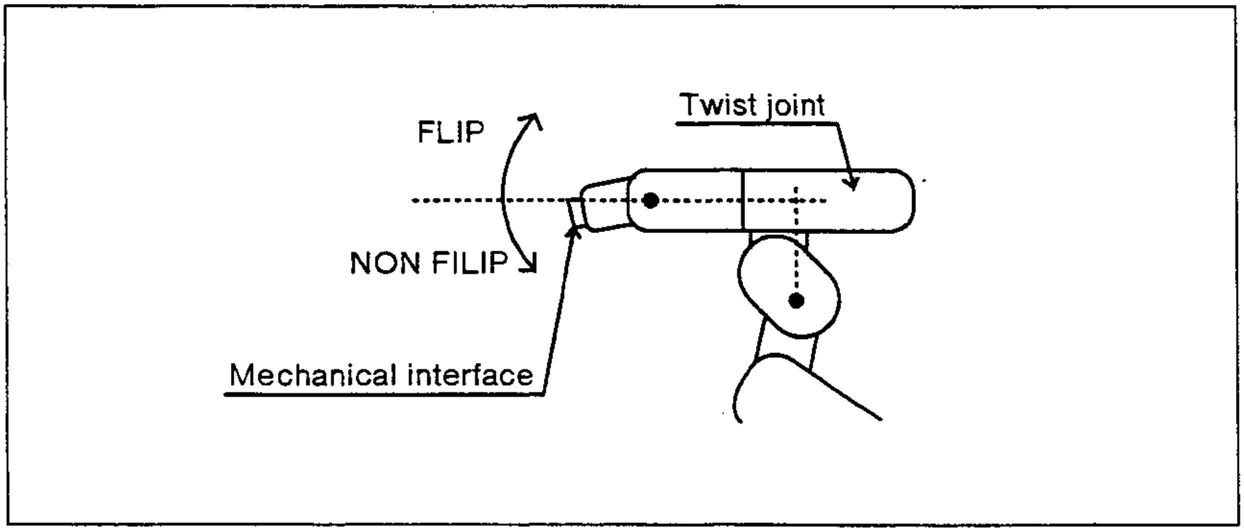


Fig. 6.19 NONFLIP / FLIP

Index

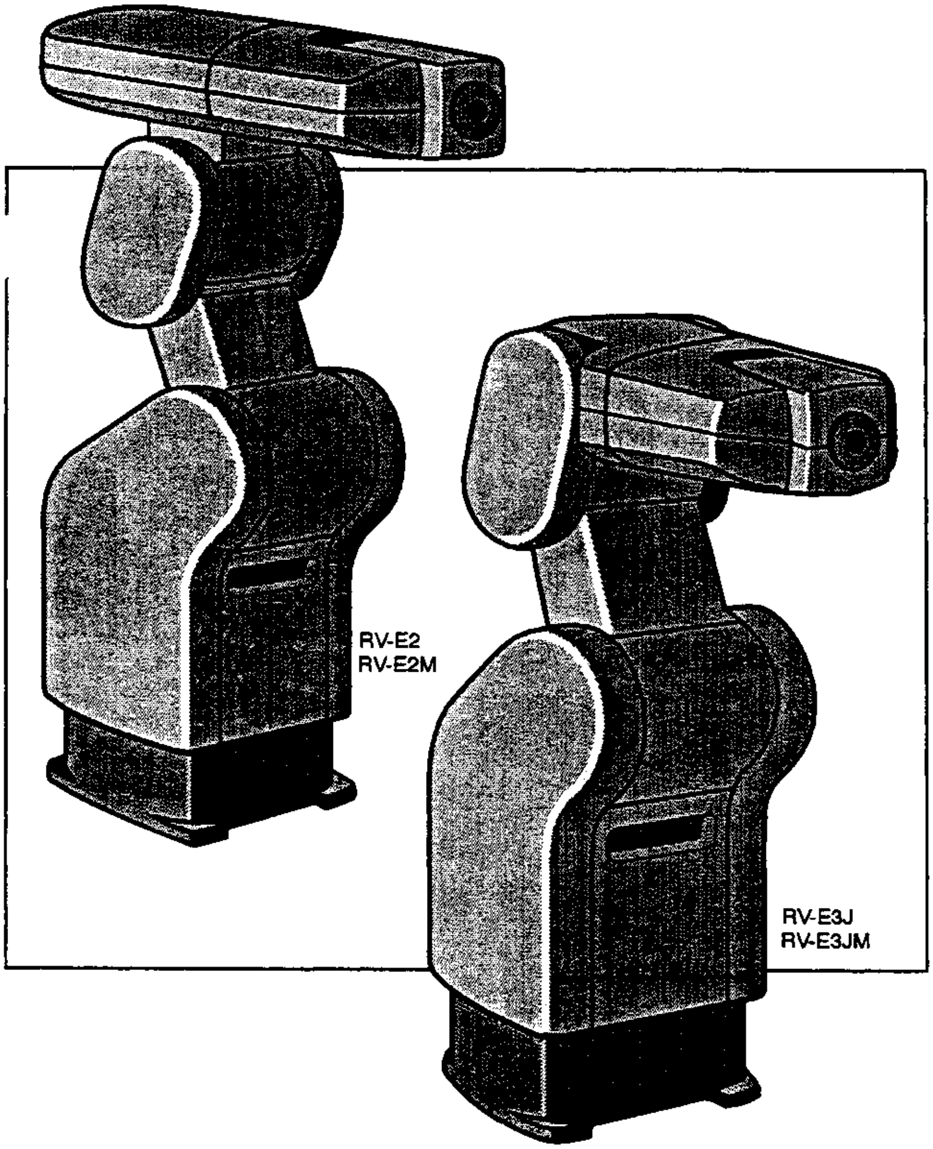
MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100 TELEX. J24532 CABLE MELCO TOKYO
NAGOYA WORKS 1-14 . YADA-MINAMI 5 . HIGASHI-KU . NAGOYA . JAPAN

MITSUBISHI

Mitsubishi Industrial Robot
RV-E2 / RV-E2M
RV-E3J / RV-E3JM

MOVEMASTER SUPER

Reference Manual



MELFA