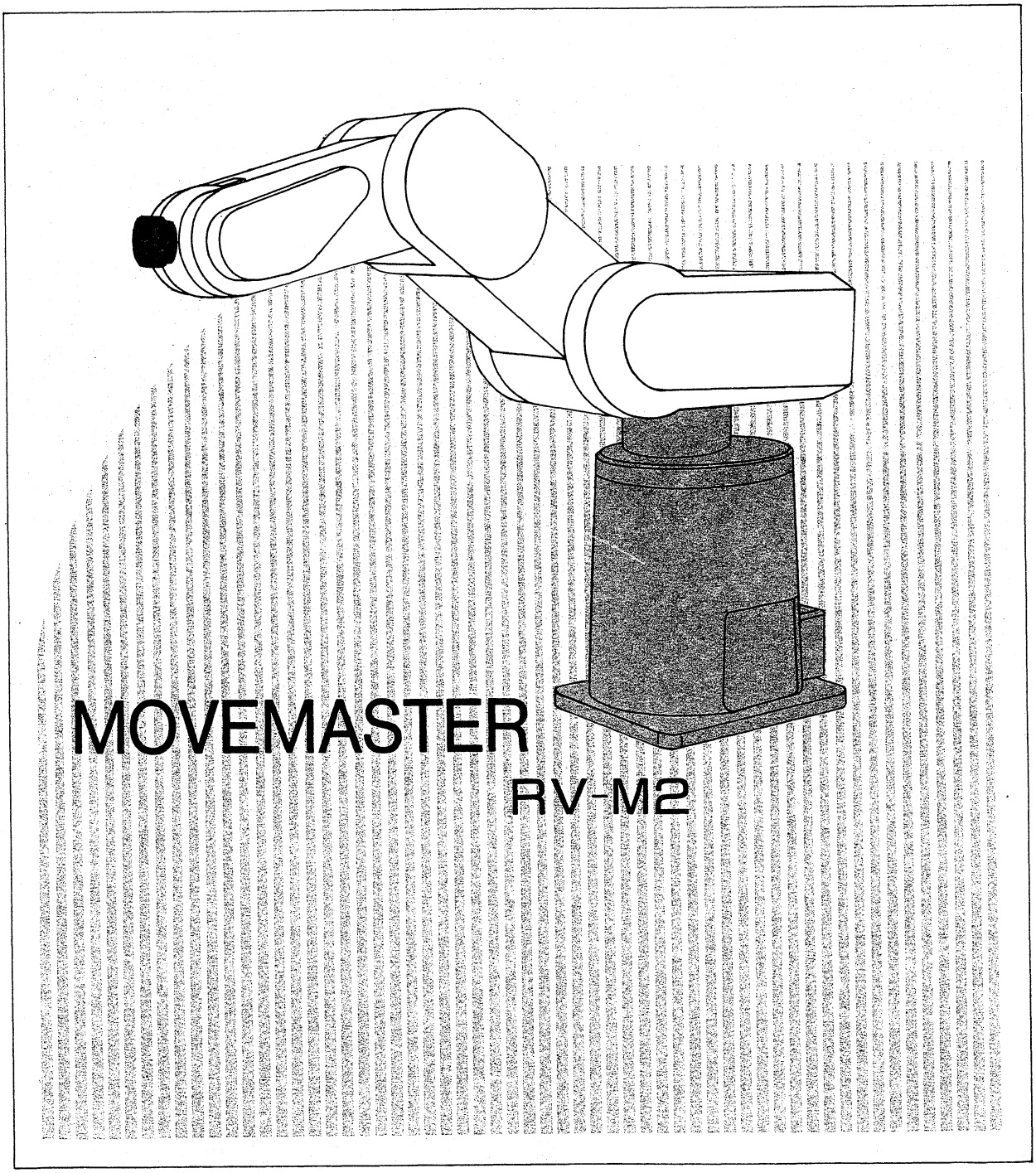
MITSUBISHI

INDUSTRIAL MICRO-ROBOT SYSTEM

Model

RV-M2

INSTRUCTION MANUAL



BFP-A5296E-A

INTRODUCTION

Thank you for choosing the Mitsubishi Electric "MOVEMASTER RV-M2" Industrial Micro-Robot. Developed for use at higher speed and with greater power and higher accuracy than the conventional "MOVEMASTER EX", the RV-M2 can be utilized in a wider range of applications such as educational and research purposes and the automation of handling works in production lines and inspection works at laboratories.
Please read this manual carefully so that the equipment is used to its optimum.

THIS MANUAL IS DIVIDED UP AS FOLLOWS:

[1] SPECIFICATIONS

Gives overall of construction, main specifications, using instructions, etc. Please read this part first.

[2] OPERATION

Gives installation and connection procedures, basic functions of system components, powering-up to position setting procedures, and program generation and execution procedures.

[3] DESCRIPTION OF THE COMMANDS

Gives formats and usages of intelligent commands which have been classified by functions and arranged in alphabetical order.

[4] MAINTENANCE AND INSPECTION

Gives maintenance, inspection, parts replacement procedures and service parts.

[5] APPENDICES

Gives interfacing with a personal computer and external I/O equipment, technical information, optional features, command list, application programs, etc.

CONTENTS

2 OPERATION

3 DESCRIPTION OF THE COMMANDS

Note: Commands marked ※ may only be executed directly (cannot be programmed with a line number).

4 MAINTENANCE AND INSPECTION

5 APPENDICES

1. SPECIFICATIONS

2. OPERATION
3. DESCRIPTION OF THE COMMANDS
4. MAINTENANCE AND INSPECTION
5. APPENDICES

CONTENTS (SPECIFICATIONS)

1. SPECIFICATIONS

1. UNPACKING AND ACCEPTANCE INSPECTION

1.1 Unpacking Instructions

(1) Carefully read "Section 1.1 Transportation of the Robot, Vol. 2" before removing the robot from the package.

(2) Do not hold the covers (areas A, A') shown in Fig. 1.1.1 when removing the robot.

(3) Carry the robot by holding the grips (area B) and supporting the hatched portions (areas C, D) by your body. (See Fig. 1.1.2.)

(4) Brakes are being applied to the arms (areas C, D). Do not force these arms to extend to protect the components.

(5) Do not remove the arm fixing plate (area E) until the robot installation is complete. This plate protects the arm during transportation. After the robot has been installed, the plate must be removed. Do not forget to remove the plate because it is invisible from the front side.

(6) The robot fixing plate (area F) allows the robot to be self-standing. This plate must be kept installed until the robot is fixed to a surface plate, etc. and must be installed before transportation.

(7) The arm and robot fixing plates must be installed before transportation of the robot and stored for future use after they are removed.

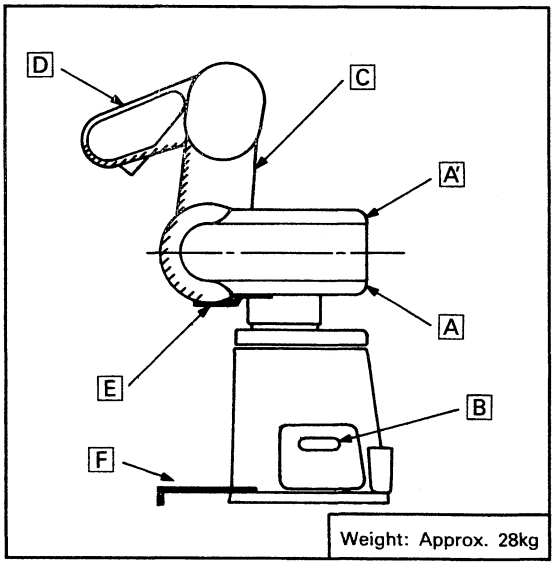


Fig. 1.1.1 Robot Attitude in the Package

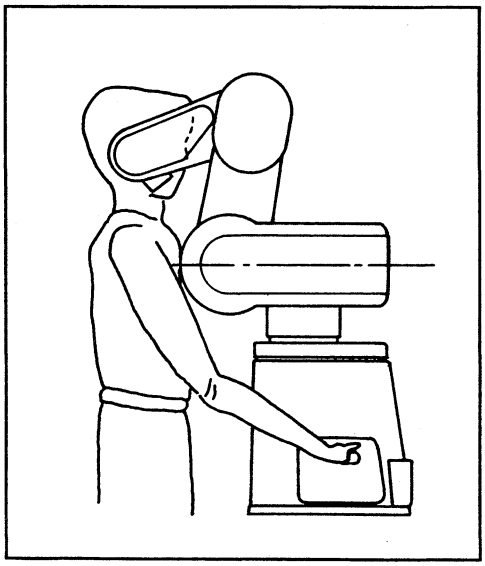


Fig. 1.1.2 Robot Transportation

1.2 Acceptance Inspection

Check that the received product conforms to your purchase order. The basic components that you have purchased are as follows:

No.	Description	Type	Quantity
1	Robot	RV-M2	1
2	Drive unit	DUM2-V5	1
3	Motor signal cable (5m)	MC50DRE	1
4	Motor power cable (5m)	MC50DRM	1
5	Power cord (2.5m)	MC25DRP	1
6	Spare fuse (10A)		1
7	Instruction manual	BFP-A5296	1
8	Warranty card		1
9	Installation bolt	M8 × 30	4
10	Spring washer for installation bolt	For M8	4
11	Flat washer for installation bolt	For M8	4

Table 1.1.1 Standard Components

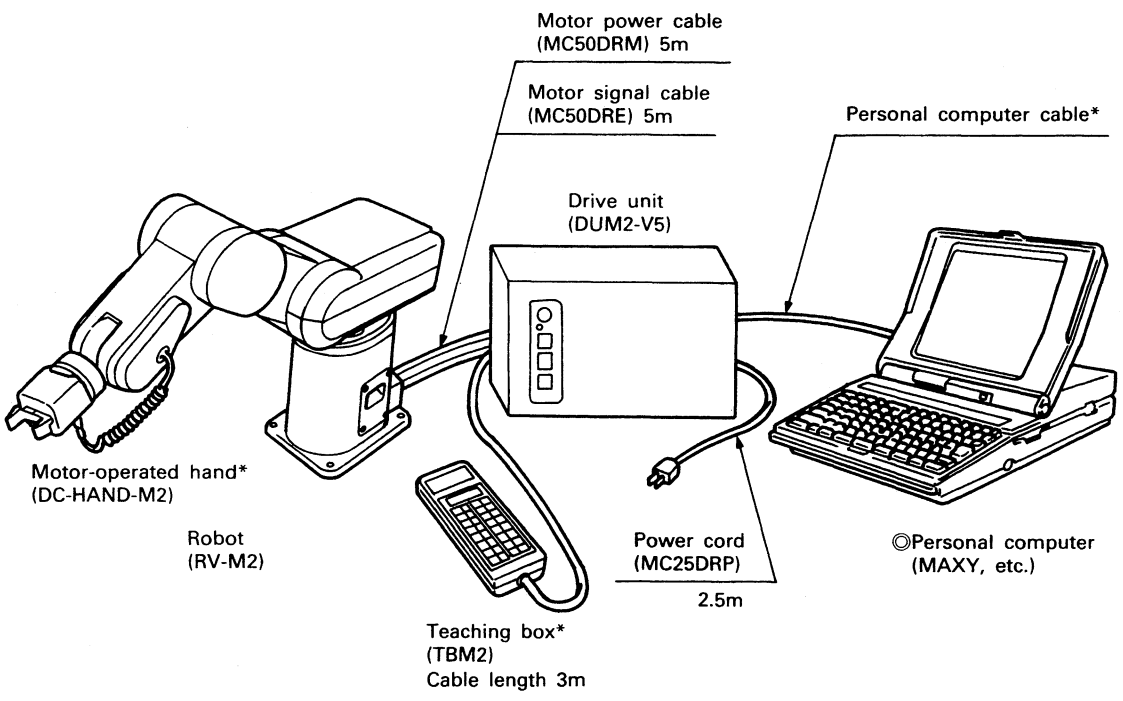
The following options are available. For more information, see Table 1.2.1 (page 1-4).

No.	Description	Type
1	Motor-operated hand	DC-HAND-M2
2	Pneumatic hand set	AIR-HAND-M2
3	Teaching box	TBM2
4	EP-ROM	512K-ROM
5	External I/O cable	I/O-CBL
6	Personal computer cable	RS-MAXY-CBL and eight others

Table 1.1.2 Options

2. SYSTEM OF CONSTRUCTION

2.1 Overall of Construction



* indicates an option.
The equipment indicated by ◎ should be prepared by the customer.

Fig. 1.2.1 Overall of Construction

2.2 Standard and Optional Equipment

Division	Description			Type	Remarks
Standard equipment	Robot			RV-M2	Vertical articulated robot with 5 degrees of freedom.
	Drive unit			DUM2-V5	Robot controller.
	Motor signal cable (5m)			MC50DRE	Gives control signals from the robot to the drive unit.
	Motor power cable (5m)			MC50DRM	Supplies motor power from the drive unit to the robot.
	Power cord (2.5m)			MC25DRP	Supplies power to the drive unit.
Optional equipment	Standard hand	Motor-operated hand		DC-HAND-M2	Only used with the Movemaster for 64-step power control.
		Pneumatic hand set		AIR-HAND-M2	Pneumatically-operated, compact, parallel, open/close type hand with solenoid valve, air tubes, and open/close end check signals.
	Teaching box			TBM2	Handy control switch box with a cable for teaching, checking, and correcting positions.
	EP-ROM (512K)			512K-ROM	Stores written programs and set positions.
	External I/O cable			I/O-CBL (5m)	For connection of an external peripheral, e.g. programmable controller.
	Personal computer cable	MULTI16	RS232C	RS-MULTI-CBL (3m)	RS-232C cable for connection of the MULTI16 (I, II, III). The cable for use with IV is available separately.
			Centronics	C-MULTI-CBL (2m)	Centronics cable for connection of the MULTI16 (I, II, III). The cable for use with IV is available separately.
		MAXY	RS232C	RS-MAXY-CBL (3m)	RS232C cable for connection of the MAXY.
			Centronics	C-MAXY-CBL (3m)	Centronics cable for connection of the MAXY.
		PC9801	RS232C	RS-PC-CBL (3m)	RS232C cable for connection of the PC9801 (UV, VM, VX, LT, etc.).
			Centronics	C-PC-CBL (1.5m)	Centronics cable for connection of the PC9801 (UV, VM, VX).
		Free cable	RS232C	RS-FREE-CBL (3m)	RS232C cable with one free end.
			Centronics	C-FREE-CBL (1.5m)	Centronics cable with one free end.

Table 1.2.1 Standard and Optional Equipment

3. MAIN SPECIFICATIONS

3.1 Robot

3.1.1 Nomenclature

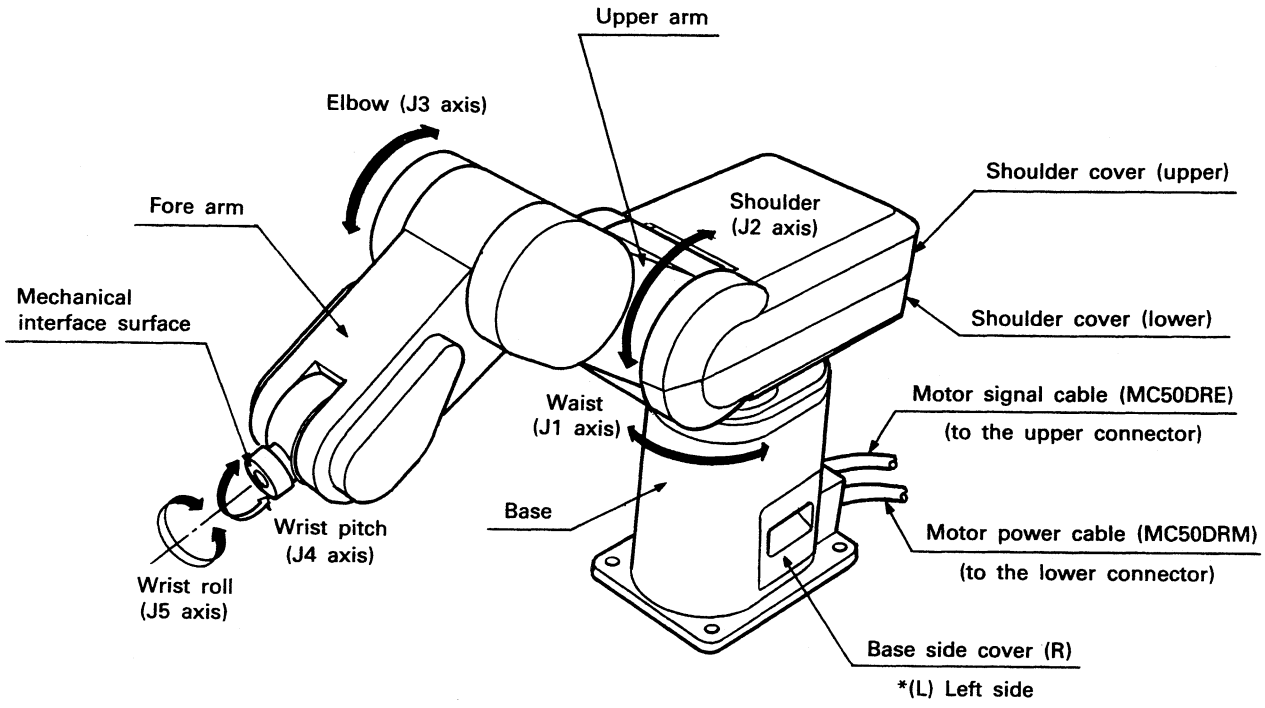


Fig. 1.3.1 Nomenclature (External View)

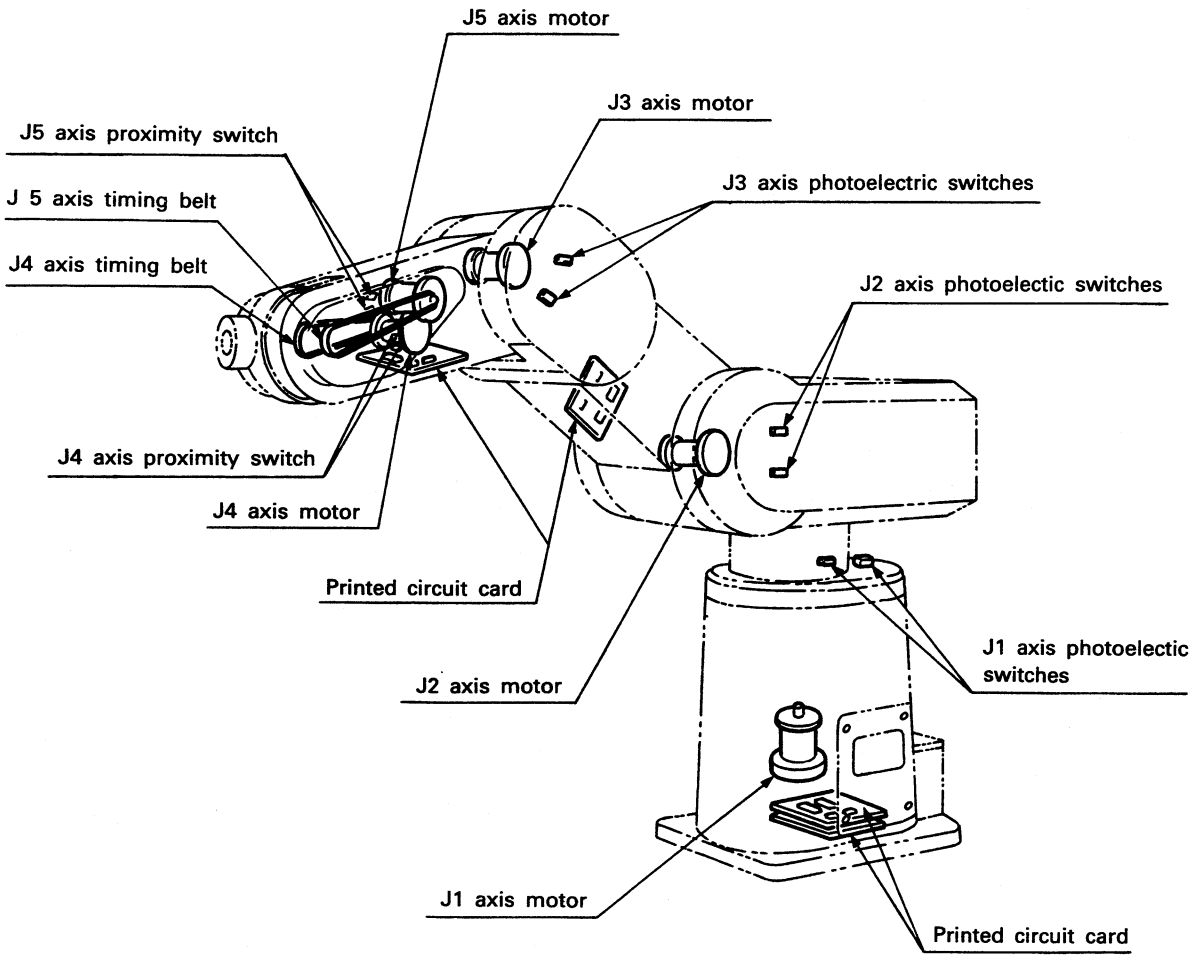


Fig. 1.3.2 Nomenclature (Internal View)

3.1.2 Standard specifications

Item		Specifications	Remarks
Mechanical Structure		5 degrees of freedom, vertical articulated robot	
Operation range	Waist rotation	300° (max. 140°/sec)	J1 axis
	Shoulder rotation	130° (max. 79°/sec)	J2 axis
	Elbow rotation	120° (max. 140°/sec)	J3 axis
	Wrist pitch	±110° (max. 163°/sec)	J4 axis
	Wrist roll	±180° (max. 223°/sec)	J5 axis
Arm length	Offset	120mm	
	Upper arm	250mm	
	Fore arm	200mm	
Weight capacity		Max. 2kgf (including the hand weight)	100mm from the mechanical interface (center of gravity)
Maximum path velocity		1500mm/sec (mechanical interface surface)	Speed at point P in Fig. 1.3.4
Position repeatability		±0.1mm (roll center of the mechanical interface surface)	Accuracy at point P in Fig. 1.3.4
Ambient temperature		5°C to 40°C	
Humidity		45 to 85%RH (non-condensing)	
Drive system		Electrical servo drive using DC servo motors	
Robot weight		Approx. 28kgf	
Motor capacity		J1, J2 axes: 60W; J3 axis: 40W; J4, J5 axes: 23W	

Table 1.3.1 Standard Specifications

3.1.3 External dimensions

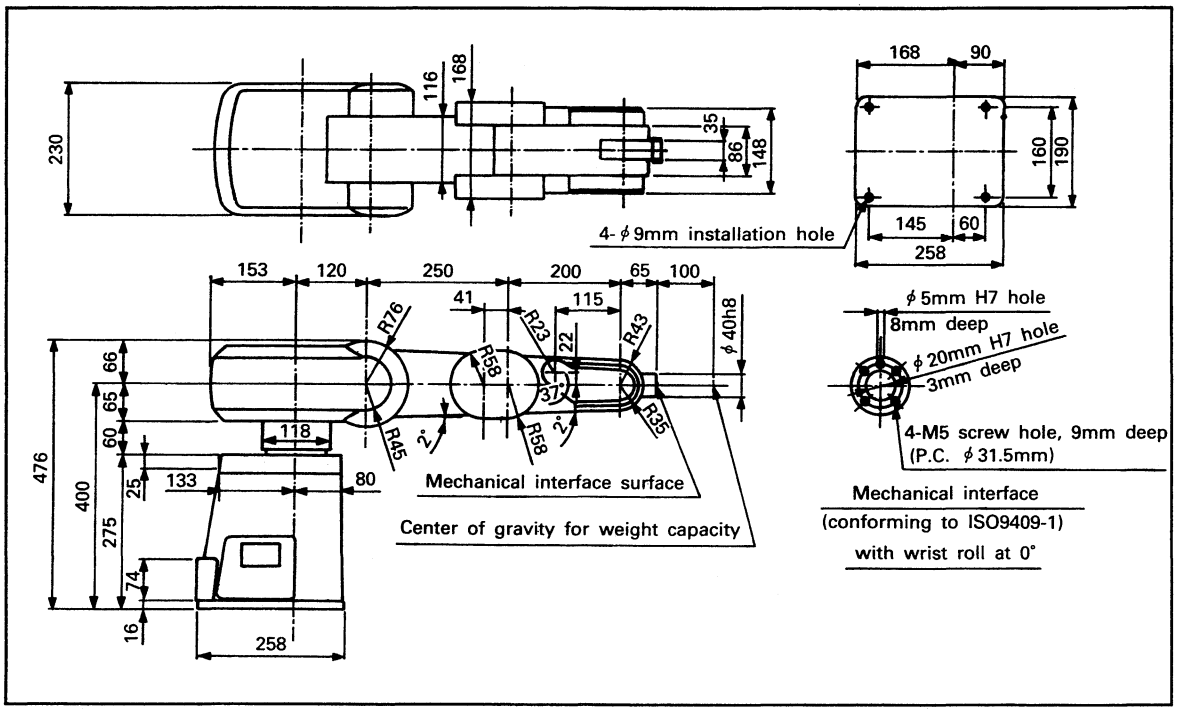


Fig. 1.3.3 External Dimensions

3.1.4 Operation space

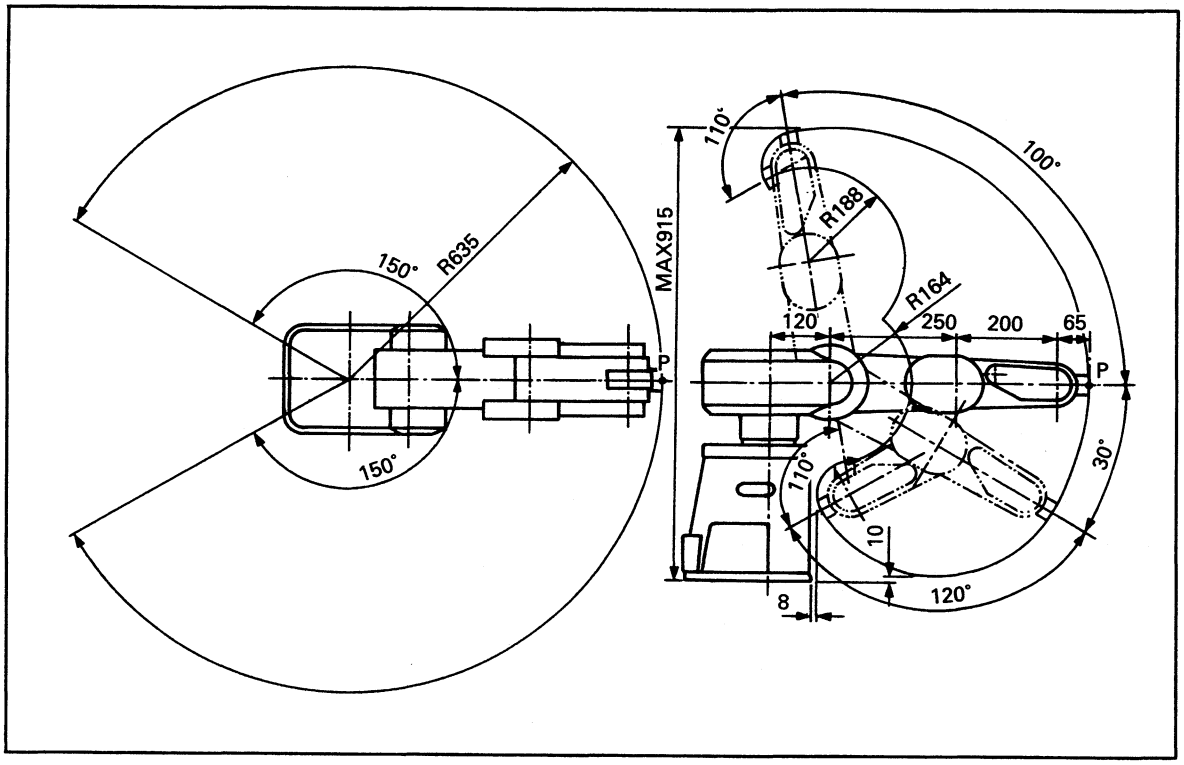


Fig. 1.3.4 Operation Space

Note 1: The operation space indicated in Fig. 1.3.4 assumes that the hand is not installed to the robot. (Trace of point P)

Note 2: Jog operation must be performed with special care because the wrist may interfere with the robot base and floor surface.

REMARKS

Jog operation indicates manual operation using the teaching box.

3.1.5 Basic operations

Fig. 1.3.5 shows axis operations in the articulated system.

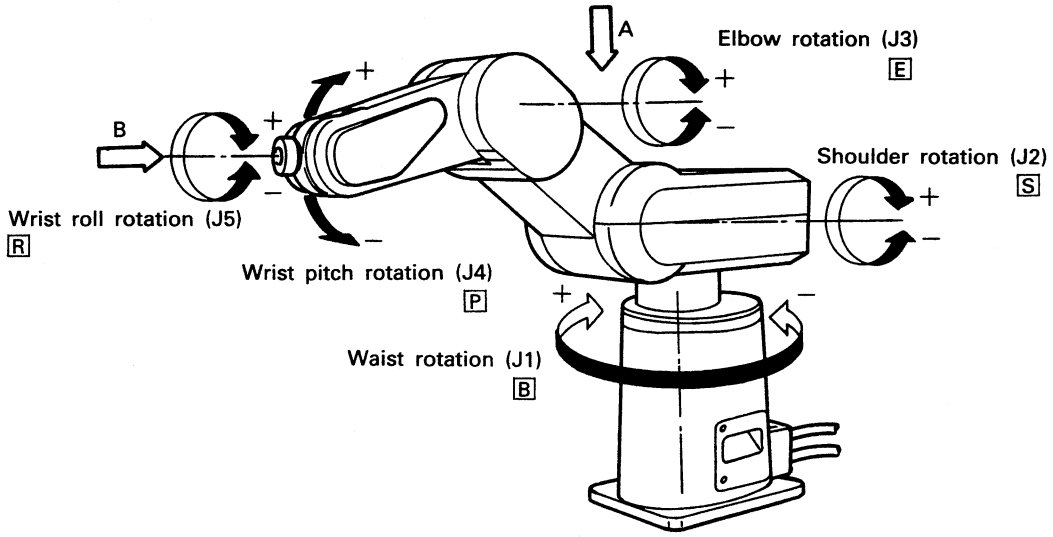


Fig. 1.3.5 Operations in the Articulated System

Note 1: The positive direction of the J1 and J5 axes is clockwise as viewed in the direction of arrows A and B, respectively.

Note 2: The positive direction of the J2, J3 and J4 axes is the upward direction of the arm and wrist.

Fig. 1.3.6 shows operations in the cartesian coordinate system.

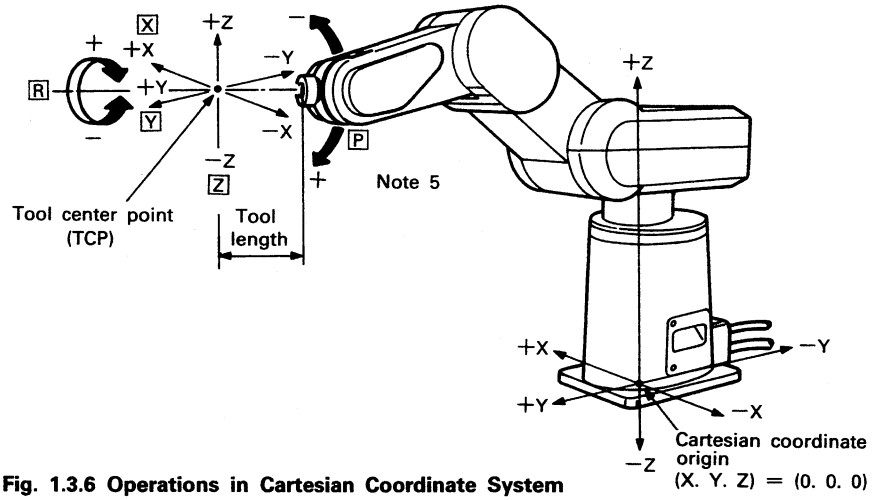


Fig. 1.3.6 Operations in Cartesian Coordinate System

Note 3: The TCP moves straight in the cartesian coordinate system.

Note 4: The tool length is set in the parameter. (See the TL command.)

Note 5: [P] in the cartesian coordinate system indicates the robot attitude changing operation without moving the TCP.

Note 6: [Z] in the tool system indicates the forward and backward operations in the tool length direction.

REMARKS

Symbols in "□" indicate the control keys of the teaching box.

3.1.6 Origin setting (Return to origin)

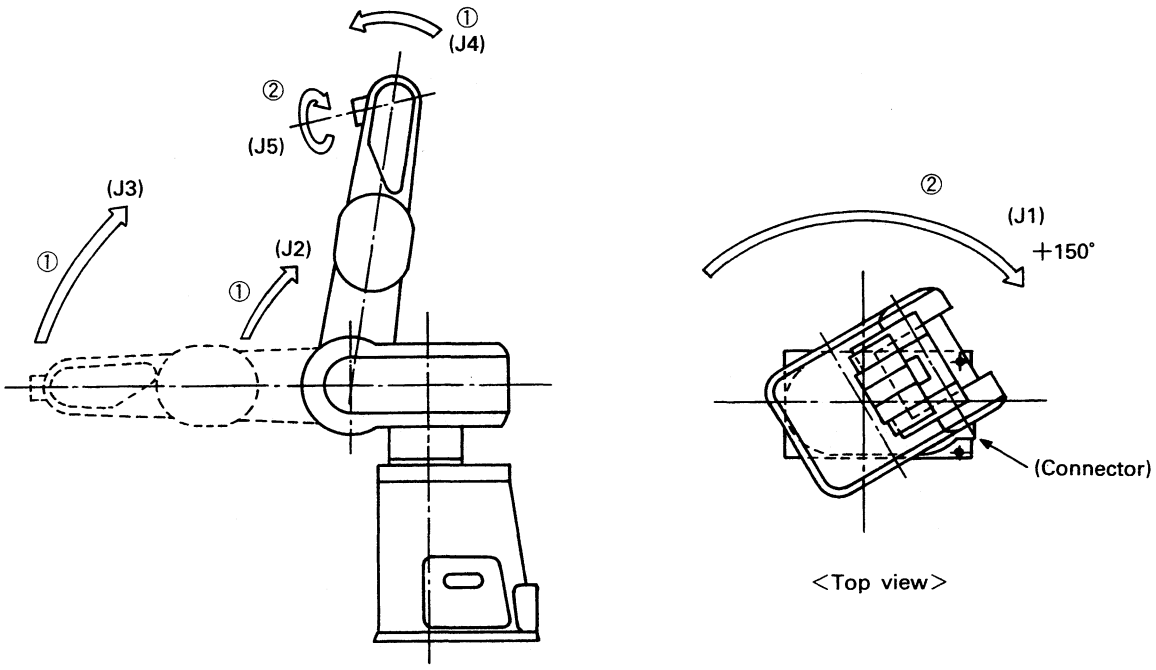


Fig. 1.3.7 Origin Setting

(1) The robot must be returned to origin after power on. (For details, see Section 4.4 Origin Setting, Vol. 2.)

(2) The arm should be moved as appropriate by jog operation before origin setting so that the robot will not interfere with any peripheral equipment during origin setting.

(3) Origin setting procedure (See Fig. 1.3.7.)

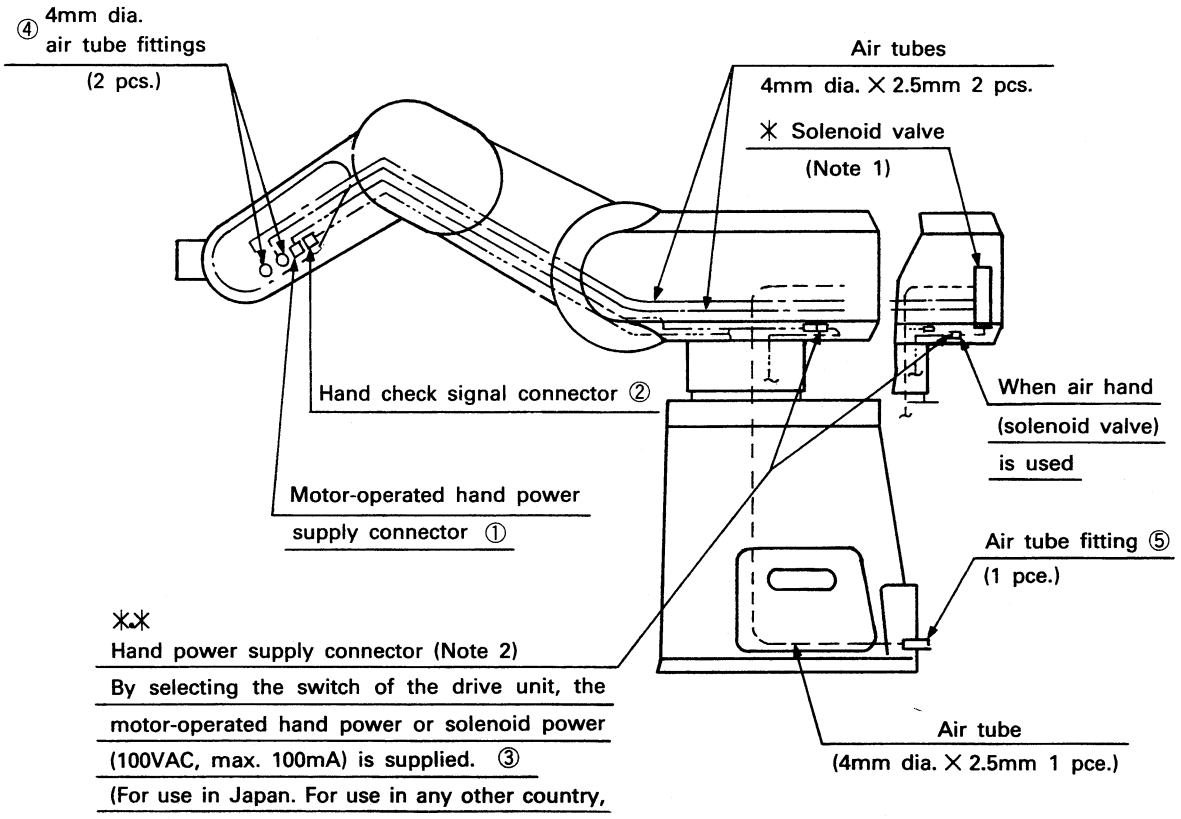
①		②
Move the shoulder (J2) in the positive direction. Move the elbow (J3) in the positive direction. Move the wrist pitch (J4) in the negative direction.	After step ① is complete. →	Move the waist (J1) in the positive direction. Move the wrist roll (J5) in the positive direction.

Note 1: All axes in step ① and those in step ② must be returned to origin at the same time.

3.1.7 Wiring and piping for hand

(1) Wiring and piping for hand (standard equipment)

✳ Note 1: When the air hand is used, a solenoid valve (power supply: 100VAC, starting current: 100mA max.) should be prepared by the customer. (For use in Japan. For use in any other country, see Appendix 9.1.)
Recommanded part: 110-4E1-83-PLL 100VAC manufactured by Koganei
The air hand set is available as option, which includes the solenoid valve.



✳✳ Note 2: The hand power supply connector indicated by ③ is factory-connected to the motor-operated hand power supply. When the air hand (solenoid valve) is used, disconnect this connector, connect the solenoid valve instead, and set the select switch of the drive unit to the AC position.

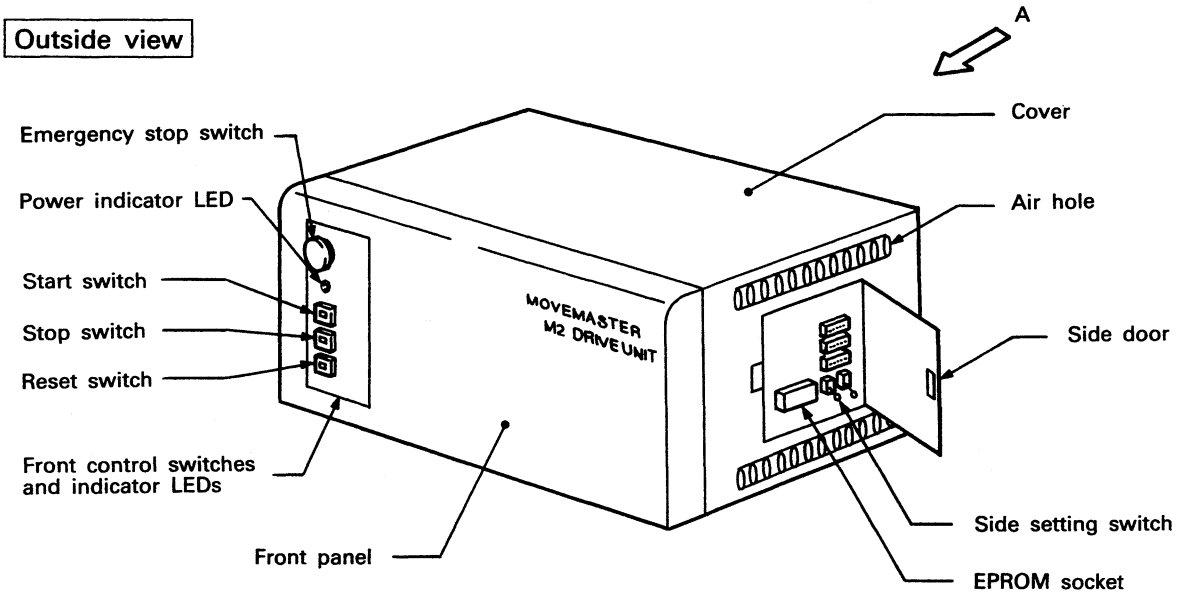
No.	Description	Robot Side		Mating Side		Manufacturer	Remarks
		Connector or fitting	Connector pin	Connector	Connector pin		
①	Connector	SMP-02V-BC	SHF-001T-0.8SS	SMR-02V-B	SYM-001T-0.6	Nippon Solderless Terminal Sales	
②	Connector	SMP-04V-BC		SMR-04V-B			
③	Connector	172328-9	170366-1	172336-9	170364-1	Nippon AMP	
④	Fitting hose nipple	M-5H-4	——	——	——	SMC	
⑤	Fitting hexagon socket head half union	KQS04-M5	——	——	——		

Fig. 1.3.8 Wiring and Piping for Hand

3.2 Drive Unit

3.2.1 Nomenclature

Outside view



Rear view (View A)

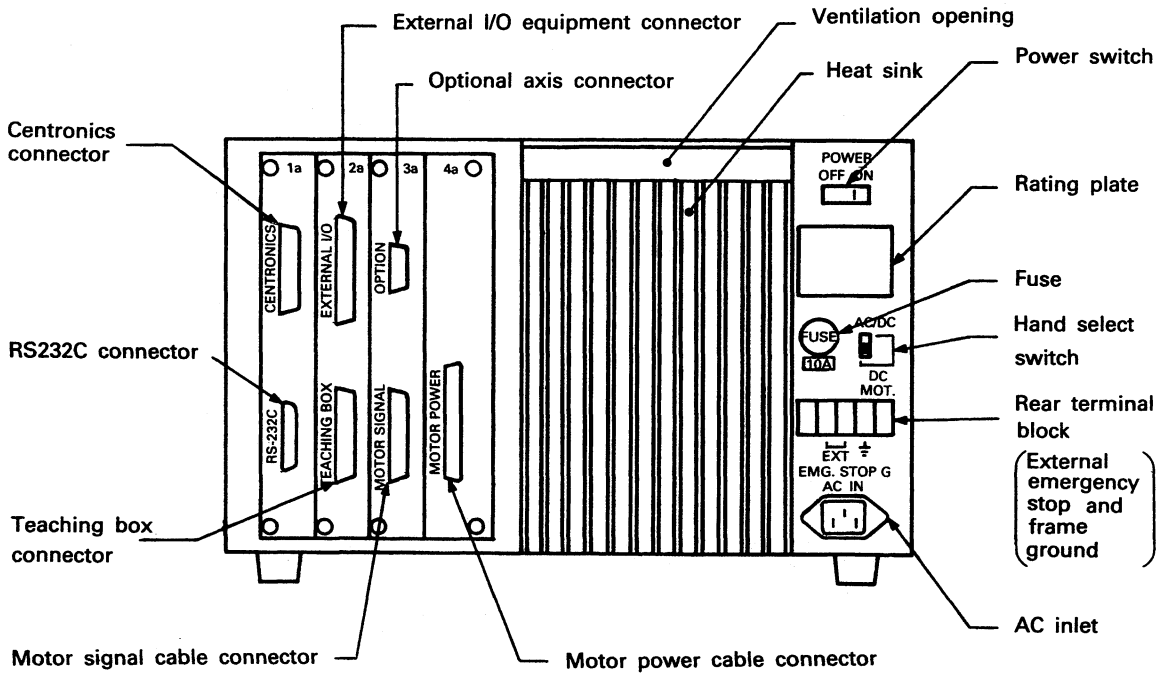


Fig. 1.3.9 Nomenclature (Drive unit)

3.2.2 Standard specifications

Item	Specifications
Teaching method	Teaching and MDI*1 systems combined (using a personal computer)
Control method	PTP and CP position velocity control system using DC servo motors
Number of control axes	5 axes (+ 1 optional axis)
Position detection	Pulse encoder system (incremental type)
Origin setting	Photoelectric magnetic switches and pulse encoders (Z phase detection system)
Interpolation function	Articulation interpolation, linear interpolation
Speed setting	21 steps (max. 1500mm/s) for PTP, 0.1mm/s increments (max. 500mm) for CP.
Number of positions	999 (24.4kB)
Number of program steps	3584 (28kB)
Data storage	Write to EPROM using the built-in EPROM writer or storage in the battery-backed static RAM (the standard battery backs up the RAM for about 2 years).
Position teaching equipment	Teaching box (option)*2 or personal computer*3
Programming equipment	Personal computer*3
External I/O	General-purpose I/O, 16 points each (2 hand open/close confirmation input points available separately) General-purpose I/O synchronization signals ($\overline{\text{STB}}$, $\overline{\text{BUSY}}$/ACK, $\overline{\text{RDY}}$) 3 dedicated I/O ($\overline{\text{START}}$, $\overline{\text{STOP}}$, $\overline{\text{RESET}}$/RUN, $\overline{\text{WAIT}}$, $\overline{\text{ERROR}}$) Power supply for external I/O should be user-prepared. (12V to 24V DC)
Interface	1 parallel interface (conforming to Centronics) 1 serial interface (conforming to RS232C)
Emergency stop	Using any of the controller front switch, teaching box switch and rear terminal block (N/C contact terminals)
Hand control	Motor-operated hand or pneumatic hand (using 100VAC*4 solenoid)
Brake control	J2 axis (shoulder), J3 axis (elbow)
Power supply	100/120/200/220/230/240VAC ±10%*5, 50Hz/60Hz, 0.5kVA*6
Ambient temperature	5°C to 40°C
Humidity	45 to 85% (non-condensing)
Weight	25kgf
Size	380 (W) × 331 (D) × 246 (H) mm

Table 1.3.2 Drive Unit Standard Specifications

Note 1: MDI stands for Manual Data Input.

Note 2: Optional.

Note 3: To be user-prepared.

Note 4: See Appendix 9.1.

Note 5: Depending on the source power voltage in your country.

Note 6: During ordinary operation. When the power is switched on, rush current (several ten amperes) may flow.

3.2.3 External dimensions

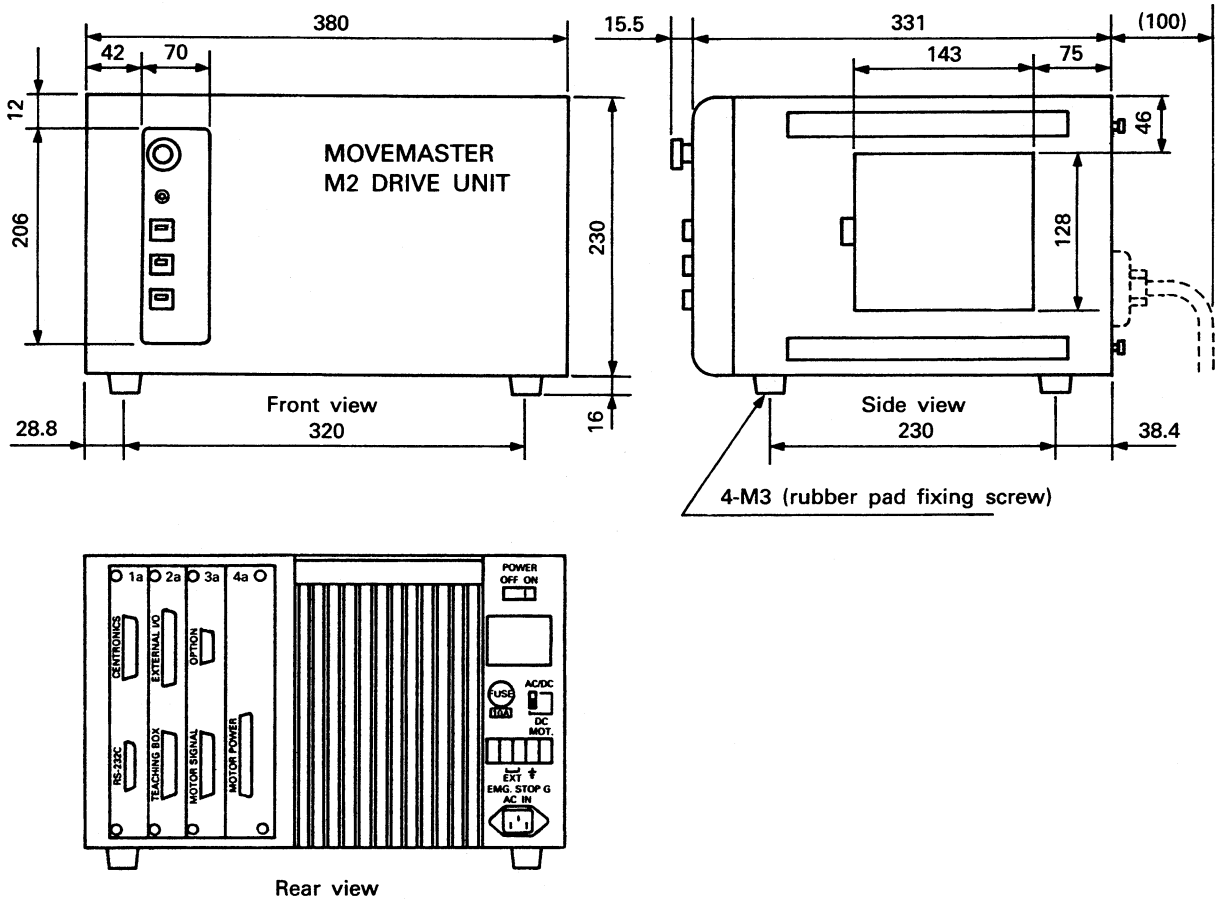


Fig. 1.3.10 External Dimensions (Drive unit)

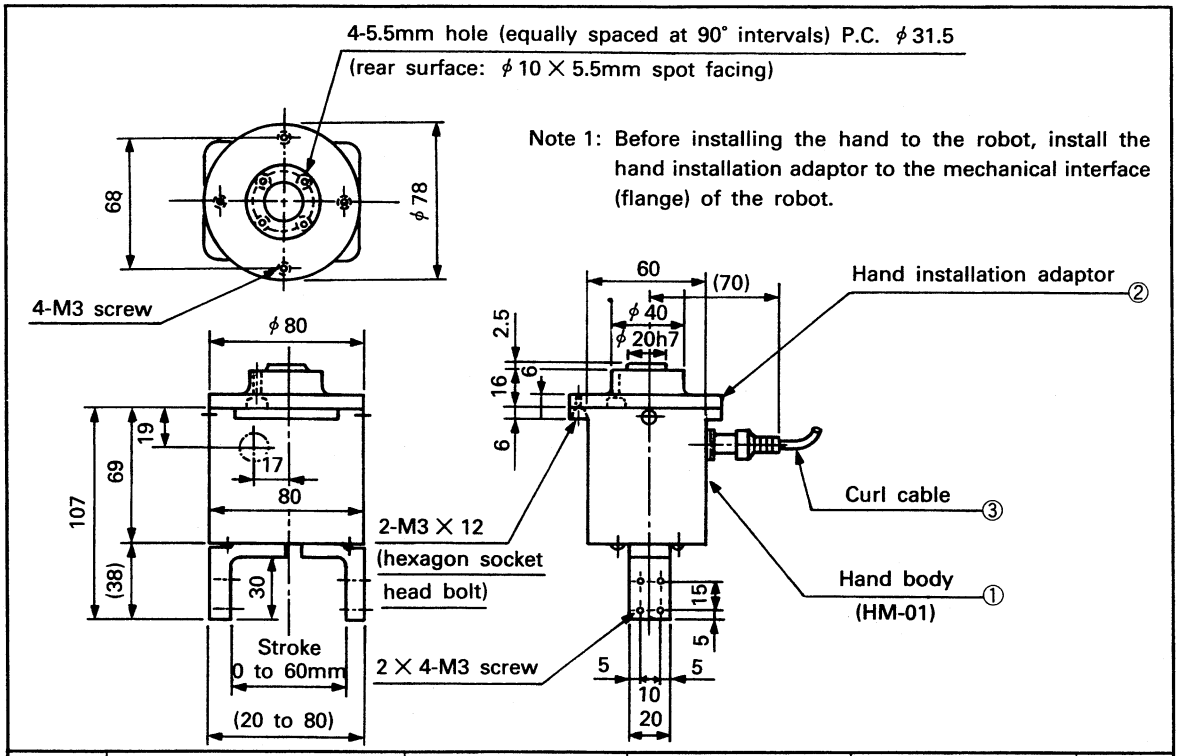
3.3 Options (Robot)

3.3.1 Motor-operated hand (DC-HAND-M2)

The motor-operated hand for the RV-M2 can be easily installed to the robot and allows the holding power to be set by current control.

Item	Specifications	Remarks
Description	Motor-operated hand (for RV-M2)	The holding power can be set in 64 steps. (See the GP command.)
Type	DC-HAND-M2 (hand body type: HM-01)	
Drive system	Servo motor drive	
Opening/closing stroke	0 to 60mm	
Grip power	Max. 3.5kgf	
Ambient temperature	5 to 40°C	
Service life	More than 300,000 times	
Weight	600gf	

Table 1.3.3 Motor-Operated Hand Specifications



No.	Description	Type	Manufacturer	Remarks	Qty
①	Motor-operated hand	HM-01	Mitsubishi Electric		1
②	Hand installation adaptor	BU144D697H01	Mitsubishi Electric		1
③	Curl cable	BU144D508G51	Mitsubishi Electric		1

Fig. 1.3.11 Motor-Operated Hand Outside Dimensions and Components

3.3.2 Pneumatic hand set (AIR-HAND-M2)

The structure of the pneumatic hand set is shown in Fig. 1.3.12. The components are shipped as single parts and must be installed by the user in accordance with "Section 1.7 Installation of the pneumatic Hand Set, Vol. 2."

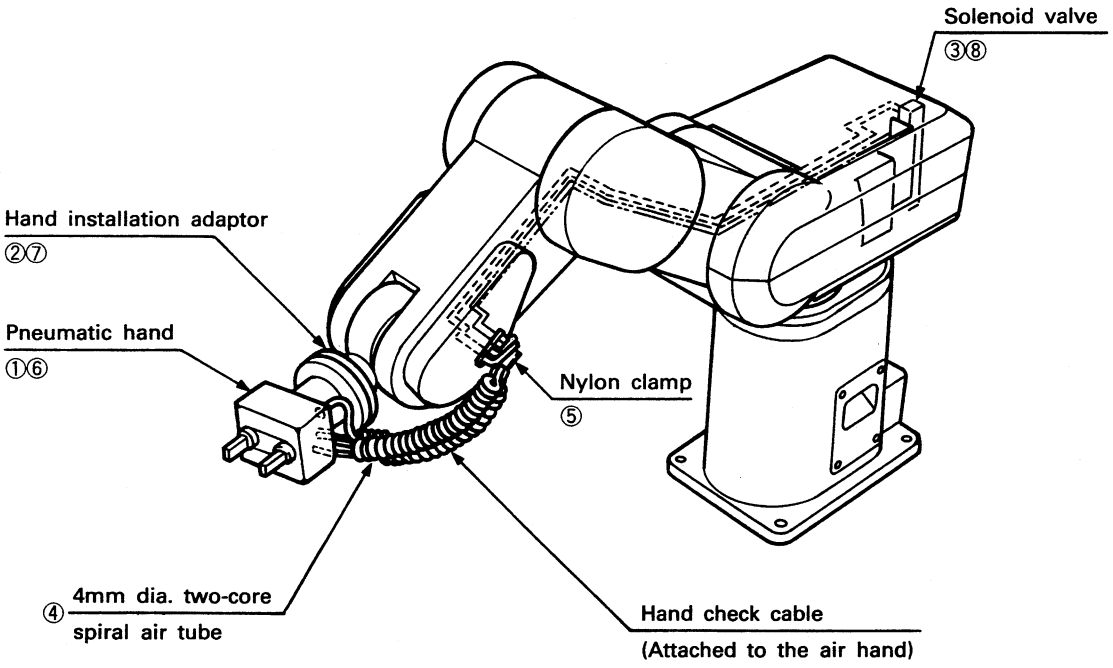


Fig. 1.3.12 Pneumatic Hand Set Structure

No.	Description	Type	Manufacturer	Remarks	Qty
①	Pneumatic hand	HM-51	Mitsubishi Electric	Including hand check cable	1
②	Hand installation adaptor	BU144D697H01	Mitsubishi Electric		1
③	Solenoid valve	110-4E1-32WY	Koganei	Mitsubishi Electric specifications	1
④	Spiral air tube	BU144D698H01	Mitsubishi Electric	4mm dia. × 2 cores	1
⑤	Nylon clamp	NK-6N	Kitagawa Kogyo		1
⑥	Hand installation bolt	M3 × 12 (socket head bolt)			4
⑦	Adaptor installation bolt	M5 × 16 (socket head bolt)			4
⑧	Solenoid installation bolt	M3 × 18 (socket head bolt)			2

Table 1.3.4 Pneumatic Hand Set Components

(1) Pneumatic hand specifications

Item	Specifications	Remarks
Name	Pneumatic hand	
Type	HM-51	
Working fluid	Air	
Working pressure range	1.5 to 7kgf/cm^2	150 to 700kPa
Ambient temperature	5°C to 40°C	
Opening/closing stroke	12mm	
Grip force	Max. 6.5kgf	
Service life	More than 10,000,000 times	
Operation method	Double-acting	
Piping port	4mm dia. tube fitting	
Hand weight	400g	
Opening/closing position detection	Detected at opening and closing ends	

Table 1.3.5 Air Hand Specifications

Note 1: Before installing the hand to the robot, install the hand installation adaptor to the mechanical interface (flange) of the robot.

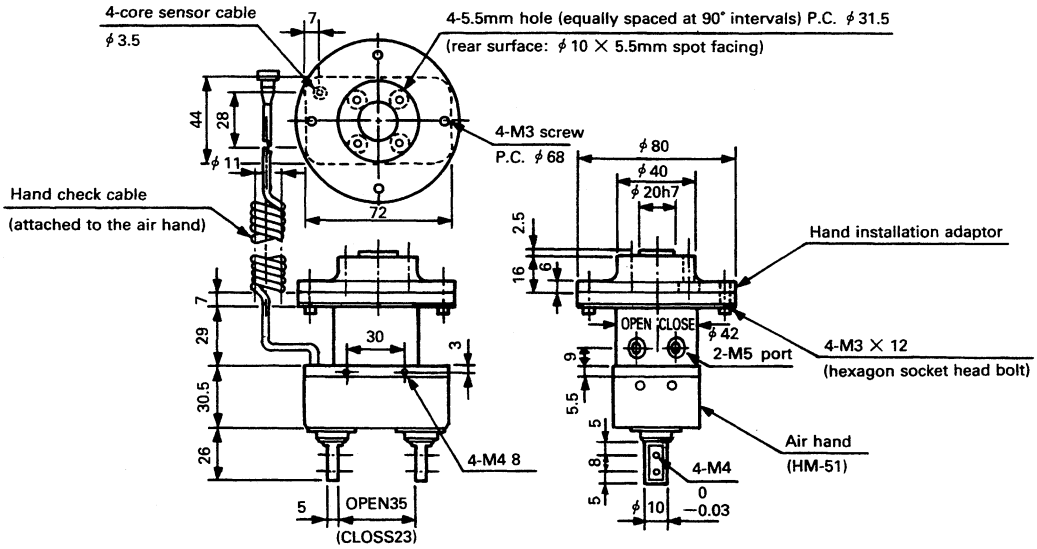


Fig. 1.3.13 Pneumatic Hand Outside View

(2) AC solenoid valve specifications

Item	Specifications	Remarks
Name	Solenoid Valve	
Type	110-4E1-32WY	Manufactured by Koganei (conforming to Mitsubishi Electric specifications)
Number of positions	2	
Number of ports	5	
Valve function	Single solenoid	
Specified fluid	Air	
Specified pressure range	1.5 to 7kgf/mm^2	150 to 700kPa
Ambient temperature	5°C to 60°C	
Line voltage	100VAC	(For use in Japan. For use in any other country, see Appendix 9.1.)
Starting current	36mA	
Effective sectional area	4.2mm^2	
Weight	80g	
Indicator lamp	LED	
Surge suppression	Varistor incorporated	

Table 1.3.6 AC Solenoid Valve Specifications

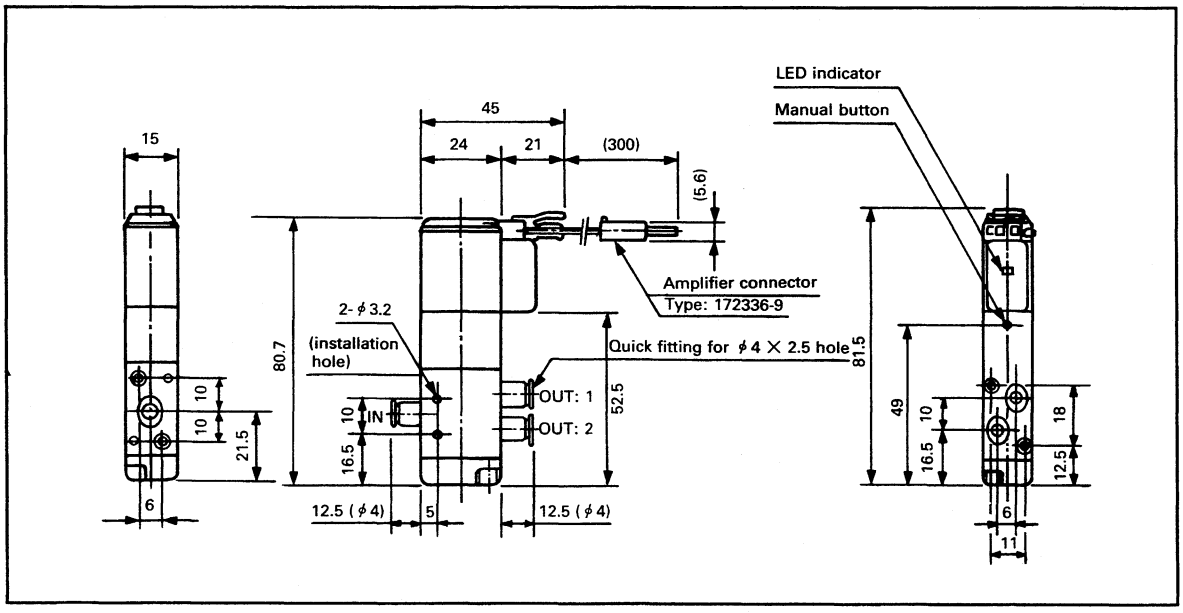


Fig. 1.3.14 AC Solenoid Valve Outside View

3.4 Options (Drive Unit)

3.4.1 Teaching box

(1) Introduction

A handy control switch box used to perform jog operation (articulation, cartesian and tool coordinate systems), hand opening/closing operation, position teaching/confirming/erasing operation, step operation, etc. of the robot. The teaching box may be installed to the drive unit only when it is needed. The program should be written using the personal computer, not using the teaching box.

(2) Nomenclature

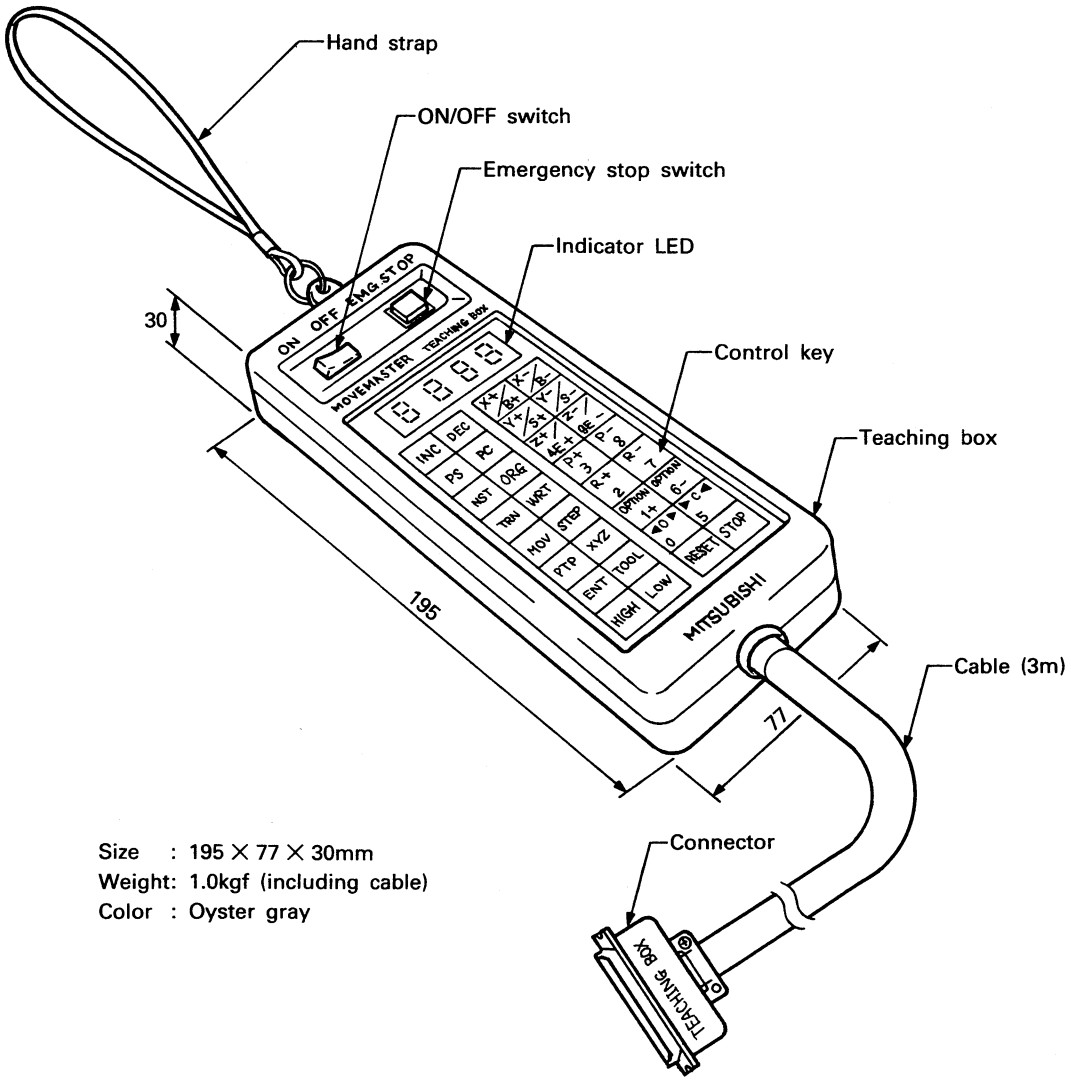


Fig. 1.3.15 Nomenclature of the Teaching Box

3.4.2 EP-ROM (512K-ROM)

EPROM (ultraviolet ray erasable type) used to store the user-written program and position data. One EPROM allows data of up to 3584 steps and 999 positions to be stored. Specify EPROM (type 512K-ROM) in your purchase order.

3.4.3 External I/O cable (I/O-CBL)

Exclusively used to connect the external I/O equipment connector in the drive unit rear panel and the peripheral equipment such as a programmable controller. (5m length) Since this cable is free on one end, use the cable by connecting a signal line corresponding to the peripheral equipment.

3.4.4 Personal computer cable

Centronics and RS232C interface cables used to connect the drive unit and the personal computer. Before placing an order, check "Section 1 or 2 INTERFACE WITH PERSONAL COMPUTER, Vol. 5" because the above interface cable of your personal computer may be used.
(Cables for use with the Mitsubishi's MULTI16 series and MAXY, NEC's PC9801 and free single-end type are also available.)

4. USING INSTRUCTIONS

4.1 Safety

To ensure safety during use of the robot, note the following:

(1) Secure sufficient working space for teaching, maintenance and other works of the robot and peripheral equipment.

(2) Install the drive unit where the operator may have a whole view of the robot operation outside the operation space of the robot (outside the safety fence, if any).

(3) Install an indicator light or the like that will tell the operators about the operating status of the robot. Signals corresponding to the robot operating states from the drive unit I/O connector may be used as appropriate.

(4) Install a safety fence or an enclosure to keep the operators from the operating region of the robot. It is recommended to provide the door of the safety fence or enclosure with an interlock mechanism that will bring the robot to an urgent or emergency stop. An emergency stop switch must be installed at a location readily accessible in case of emergency.

(5) Prepare a robot operating manual and instruct the operators to work in accordance with that manual.

(6) To ensure safety, signals should be determined for any joint work done by several operators.

(7) Inspection must be done daily and periodically to check that the robot will operate without any fault before starting operation.

4.2 Operating Environments

To ensure high reliability, smooth operation and long life, avoid operating the robot in the following environments:

(1) Power supply
- Voltage variations exceeding ±10% of the rated voltage.
- Instantaneous power failure of over 15ms.

(2) Noise
- Surge voltage of more than 1000V, 1 μs applied to the power line. Neighborhood of large-sized inverter, large-output high-frequency generator, large-sized contactor or welder.
- Neighborhood of radio and television.

(3) Temperature, humidity
- Ambient temperature outside the range 5°C to 40°C.
- Direct sunlight and neighborhood of a heat source such as a heater.
- Relative humidity outside the range 45% to 85% and condensation.

(4) Ambience
- Dust, dirt, oil mist, corrosive gases, etc.
- Conductive materials such as metal filings.

(5) Vibration
- Excessive vibration and impact. (Allowed environments: 3.5G or less during transportation, 0.5G or less during operation)

4.3 Handling

(1) Protect the plastic covers from damage.

(2) If brakes are being applied to the upper arm and fore arm, do not force them to move in order to maintain high accuracy, avoid generation of looseness, and protect the reduction gear.

(3) Before transporting the robot, the attached fixing plate must be installed in accordance with "Section 1 TRANSPORTATION, INSTALLATION AND SETTING-UP, Vol. 2"
Particularly for the secondary transportation of the robot, it is recommended to reuse the packing materials used at the time of shipment from the manufacturer.

(4) When the robot is temporarily placed, one or more base installation holes must be bolted with the fore arm and upper arm pressed against the positive stoppers. If the arms are tilted forward, the robot will fall down.

(5) After power on, the RV-M2 must be returned to origin to match its mechanical origin with control origin. If there is a possibility that the robot may hit any obstacle around depending on its attitude before power on, start origin setting after avoiding that obstacle by jog operation from the teaching box.

(6) The drive unit must be grounded in class three to improve noise durability and prevent the risk of electrical shock.

(7) To use the robot for a long time without fault, maintenance and inspection must be done periodically in accordance with "Vol. 4 MAINTENANCE AND INSPECTION PROCEDURE."

(8) Note that the optional motor-operated hand loses its gripping power at power off. Note also that the optional pneumatic hand may open at power off or emergency stop depending on the way of piping (because the single solenoid is used).

(9) Do not loosen the card fixing screws or pull off the cards in the rear panel of the drive unit. The whole drive unit will be faulty if the sequence of card installation has been changed or any card of the RV-M1 drive unit has been installed.

(10) When the robot is to be traveled during operation (e.g. used on the sliding unit), the motor power and signal cables may be broken. In such a case, please consult.

4.4 Definitions of Specifications

4.4.1 Position repeatability

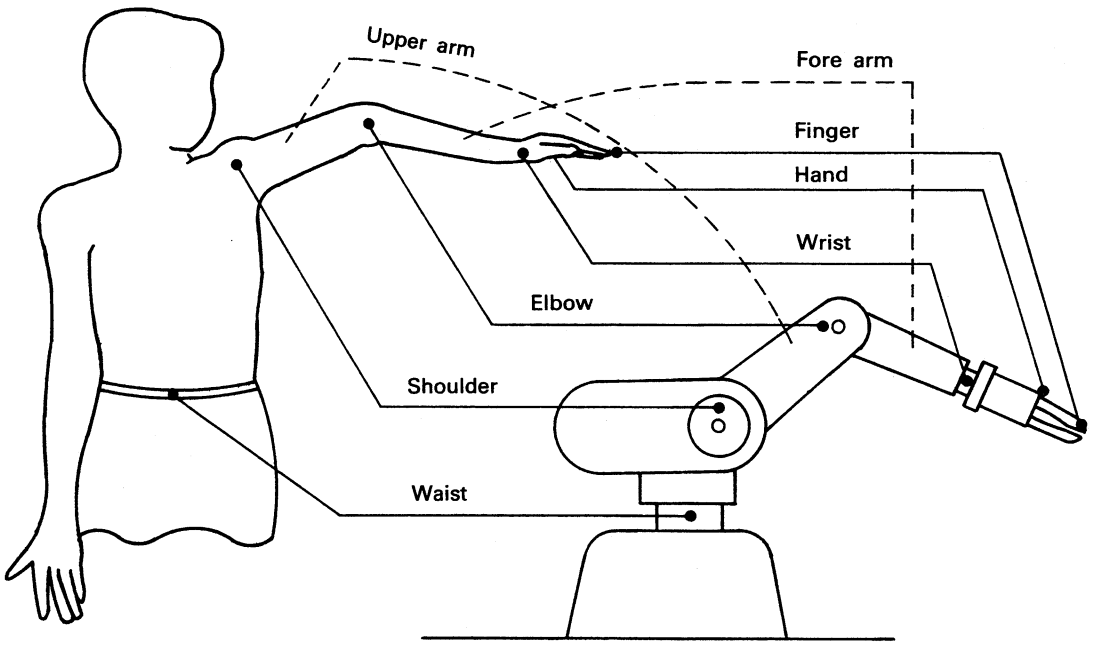
The position repeatability of the robot is based on the following industrial robot definition in the Japanese Industrial Standards. (JIS B 0134)

Position repeatability: The measure of the accuracy with which the robot is positioned under the same conditions and by the same method.

This definition does not apply to the position difference when the teaching speeds are different from execution speeds or when the ambient temperature has changed, to the numerically set coordinates, and to positioning precision, etc.

Movemaster Information (1)

Axis names The robot articulation names correspond to the human body as illustrated below.



4.4.2 Weight capacity

The weight capacity of the robot is generally indicated in weight only. Note that if two workpieces are identical in weight and different in position of the center of gravity, restriction may be put on the weight capacity of one workpiece and not on that of the other. Fig. 1.4.1 defines the weight capacity indicated in the catalog and specification. The weights of the workpiece and user-prepared hand should be determined in accordance with this definition.

Fig. 1.4.2 defines the weight capacity of the robot used with the motor-operated hand (option).

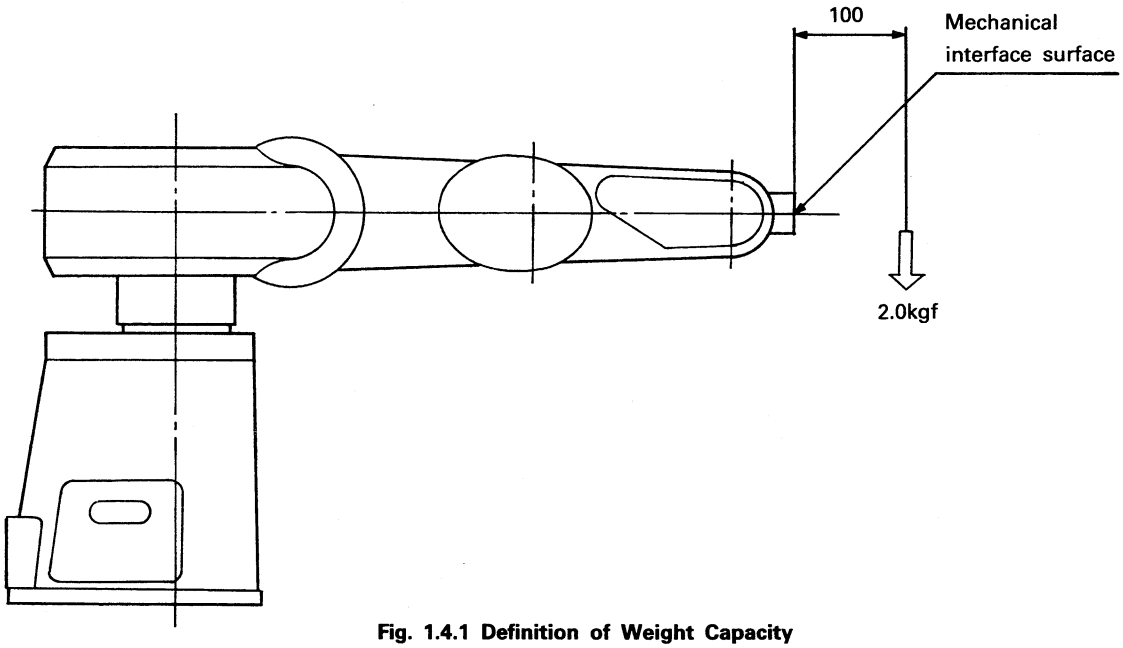


Fig. 1.4.1 Definition of Weight Capacity

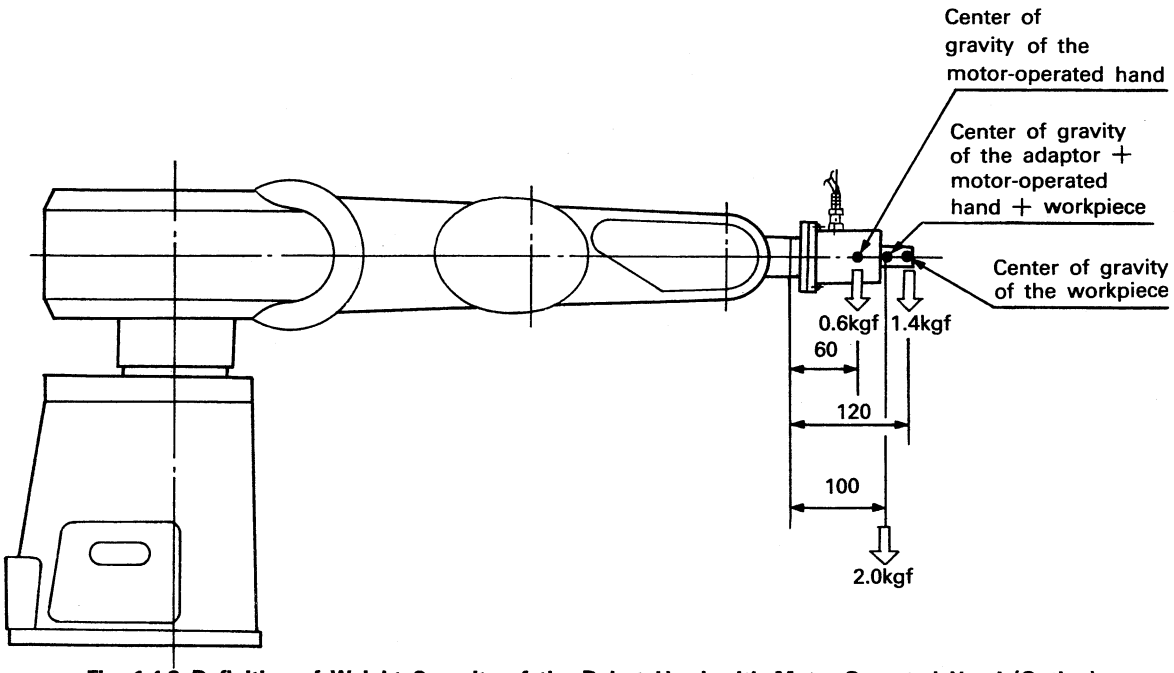


Fig. 1.4.2 Definition of Weight Capacity of the Robot Used with Motor-Operated Hand (Option)

5. WARRANTY PERIOD AND PAINT COLORS

5.1 Warranty Period

(1) The robot indicated in the warranty card is repaired free of charge within one year (i.e. 8 hours/day × 250 days) after the delivery date if a fault has occurred under the appropriate operating conditions and the fault is attributable to the manufacturing process. (This warranty does not apply to any consumable parts specified by Mitsubishi.)

(2) The robot will be repaired on a commercial basis even during the warranty period if the fault and/or damage is attributable to:

1) Act of God (e.g. earthquake, damages from wind and water), fire.
2) Using mistake.
3) Movement or transportation after installation.
4) Repair or modification made by other than Mitsubishi.
5) Equivalents for the above

REMARKS

Consumable parts (e.g. motor brush, timing belt) are changed on a commercial basis. (For consumable parts, see Section 4 MAINTENANCE PARTS, Vol. 4.)

(3) Any warranty card without the delivery date and sales representative name entered is invalid.

(4) Immediately contact the sales representative if any fault has occurred.

(5) The warranty card must be shown to the service engineer. If the warranty card is not shown, a servicing fee may be charged even during the warranty period.

5.2 Paint Colors

①Robot arm ····· Munsell 10Y8/1 70% gloss (oyster gray)
②Robot base ····· Munsell 10Y3/1 70% gloss (dark gray)
③Drive unit ····· Munsell 10Y8/1 70% gloss (oyster gray)
④Teaching box ····· Munsell 10Y7.5/1 embossed (oyster gray)

The standard paint colors are subject to change without notice.

1. SPECIFICATIONS

2. OPERATION

3. DESCRIPTION OF THE COMMANDS
4. MAINTENANCE AND INSPECTION
5. APPENDICES

CONTENTS (OPERATION)

1. TRANSPORTATION, INSTALLATION AND SETTING-UP

1.1 Transportation of the Robot

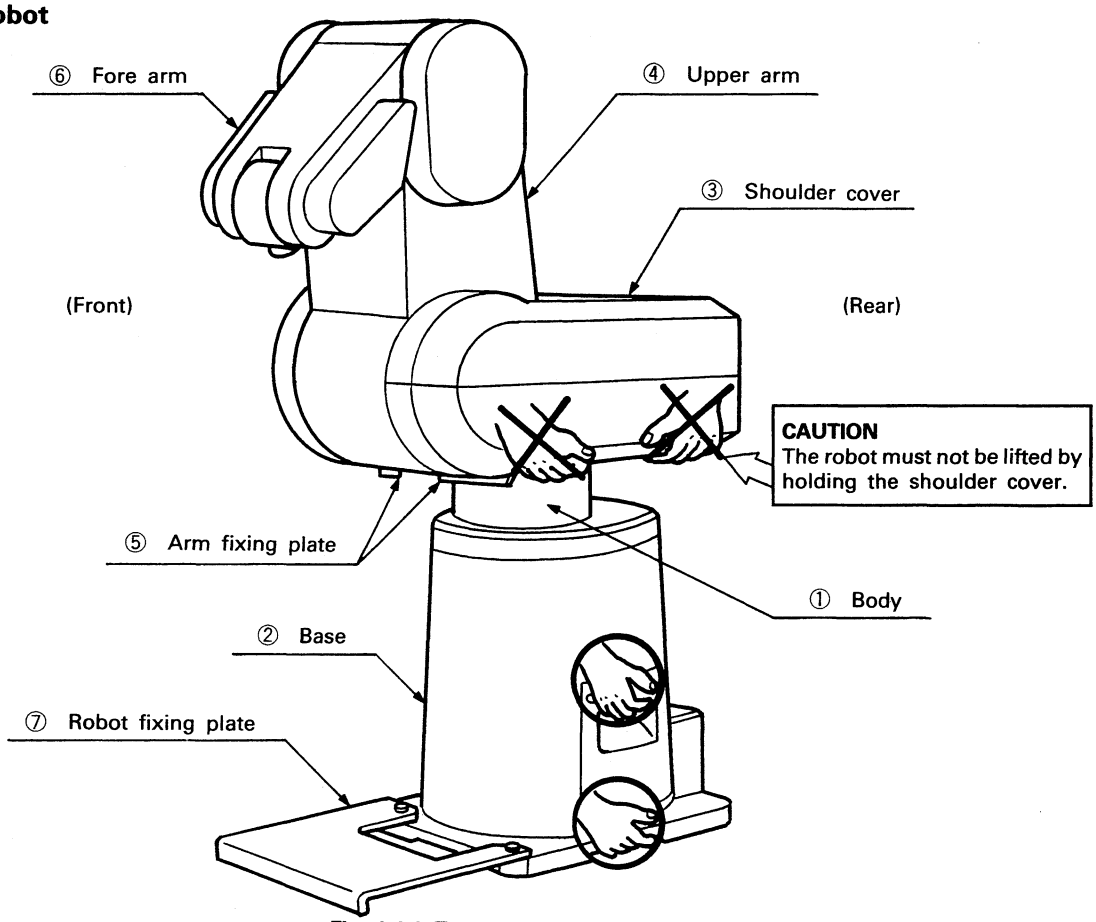


Fig. 2.1.1 Transporting the Robot

(1) Fig. 2.1.1 shows the locations to be held by hands when transporting the robot.

(2) Before transporting the robot, release the brakes in accordance with Section 2.2.4, Vol. 2, move the upper arm ④ in the positive direction and the fore arm ⑥ in the negative direction until they are pressed against the corresponding stoppers as shown in Fig. 2.1.1, and install the arm fixing plate ⑤ to the upper arm ④ and body ①. Also install the robot fixing plate ⑦ to the base ②.

(3) Transport the robot by holding the grips of the base ② as shown in Fig. 2.1.1 and supporting the arm with your body on the front side of the robot.

(4) Never carry the robot on the rear side or hold the sides or back of the shoulder cover ③ to prevent any accident, e.g. the robot may fall down or the covers may be damaged or drop.

(5) Before transporting the robot, the arm fixing plate and robot fixing plate must be installed in the attitude shown in Fig. 2.1.1.

(6) The above (2) applies to transportation of the other equipment where the robot has been installed. (At this time, the robot fixing plate need not be installed.)

1.2 Installation of the Robot

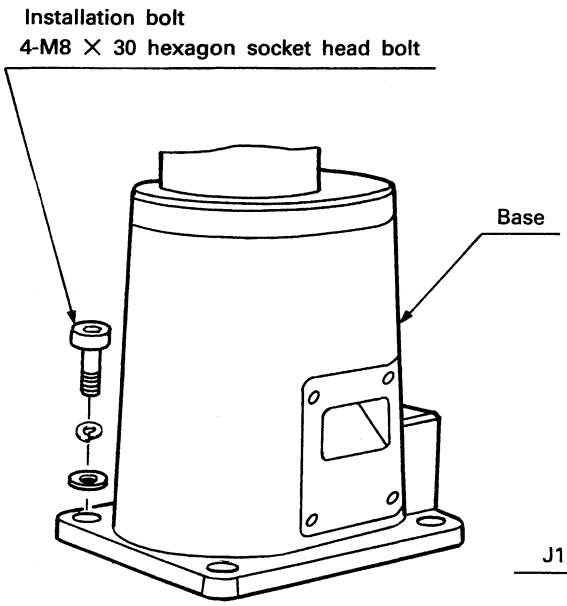


Fig. 2.1.2 Installation of the Robot

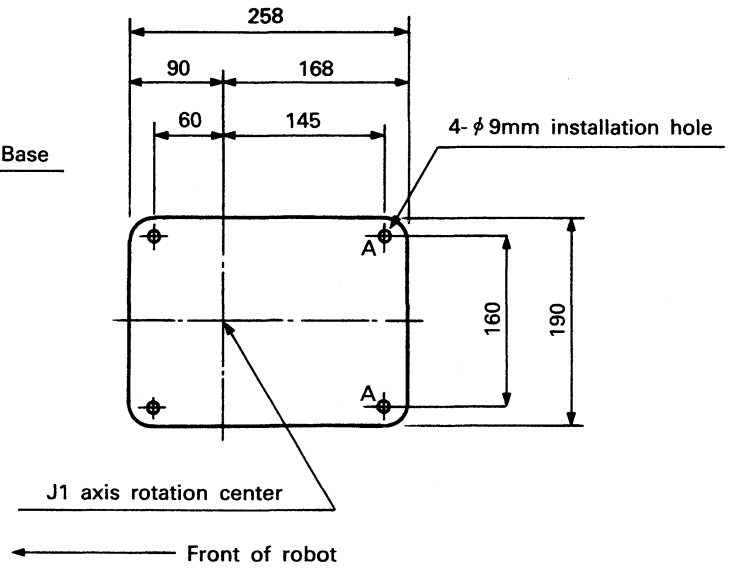


Fig. 2.1.3 Robot Installation Dimensions

(1) Install the robot in accordance with Fig. 2.1.2.

(2) The installation surface of the robot has been machined for accurate installation. Secure the robot onto the floor using the 9mm dia. installation holes provided at four corners of the base and the installation bolts (M8 × 30 hexagon socket head bolts) furnished. The installation surface must be level.

(3) Remove the robot fixing plate after fixing the robot base with the M8 bolts through the 9mm dia. holes which are not used for installing the robot fixing plate (indicated by A). Then fix the robot with the M8 bolts through the threaded holes (M10, starting hole of 9mm) which have been used for installing the robot fixing plate. It is recommended that common surface plates be used to ensure that the robot is properly aligned with the equipment, jigs and fixtures on which the robot will work.

(4) Surface roughness of the surface plates onto which the robot is to be secured must be ▽▽ or more. A rough surface can cause poor contact between the robot and surface plate, resulting in misalignment.

(5) Fig. 2.1.3 shows the robot installation dimensions.

(6) After the installation is complete, remove the arm fixing plate shown in Fig. 2.1.1. When the robot is transported as installed on a surface plate, the arm fixing plate must be kept fit on the robot until the robot is finally installed.

CAUTION

The arm fixing plate and robot fixing plate are required for the transportation of the robot and must be kept by the customer together with the bolts.

1.3 Transportation and Installation of the Drive Unit

(1) During transportation, do not subject the drive unit to excessive shocks and vibrations and do not let it topple over or tilt.

(2) Install the drive unit as horizontally as possible (within the tilting angle of 20°).

(3) Keep the ventilation opening in the drive unit open to the atmosphere.

(4) Allow a distance of 100mm or more between the back panel of the drive unit and the adjacent wall. When the drive unit is used in a cabinet, etc., make sure that it is well ventilated for ample heat dissipation.

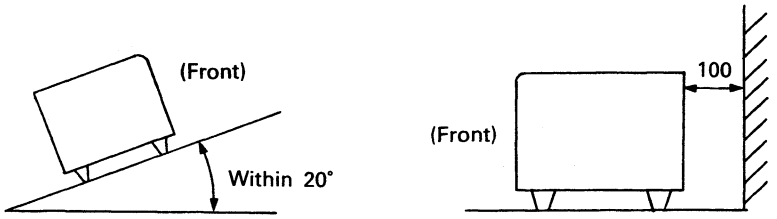


Fig. 2.1.4 Installation of the Drive Unit

1.4 Grounding

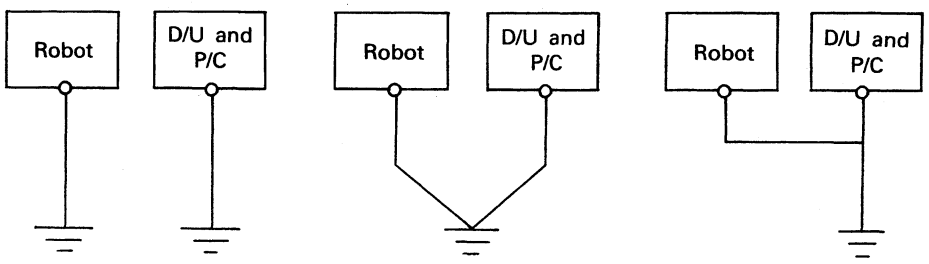
(1) Whenever feasible, ground the robot and the drive unit individually. For the drive unit, note that the frame ground terminal provided on its rear panel is the ground connection.

(2) Use the Class 3 grounding (ground resistance 100 ohms or below). The best method to be employed is the dedicated grounding.

(3) Use a 2mm2 or more grounding wire.

(4) Locate the grounding point as near the robot and drive unit and keep the grounding wire as short as possible (within 3m).

D/U: Drive unit; P/C: Personal computer



a) Dedicated (Best) b) Shared (Good) c) Common (Fair)

Fig. 2.1.5 Grounding Methods

1.5 Cable Connections

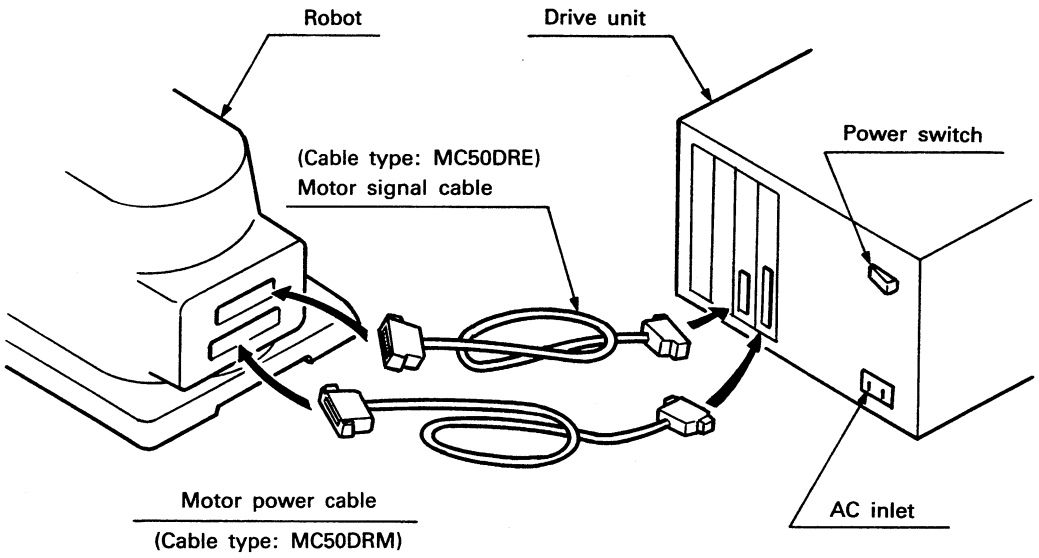


Fig. 2.1.7 Cable Connections

(1) Connect cables between the robot and drive unit as shown in Fig. 2.1.7.

(2) Before connecting any cable, check that the power switch on the drive unit rear panel is off.

(3) Plug the power cord into the AC inlet in the drive unit rear panel.

(4) Connect the motor signal cable and motor power cable to the corresponding connectors of the robot and drive unit.

(5) The two cables have no specific orientation for connection. Either end can be hooked up to either unit.

(6) When fixing any connector, raise the spring latches on both of its sides to secure the connector.

(7) Route the cables with special care. Do not pull hard or bend the cables to protect the cables and connectors.

1.6 Installation of the Motor-Operated Hand (Option)

The structure of the motor-operated hand is shown in Fig. 2.1.8 and its components are indicated in Table 2.1.1.

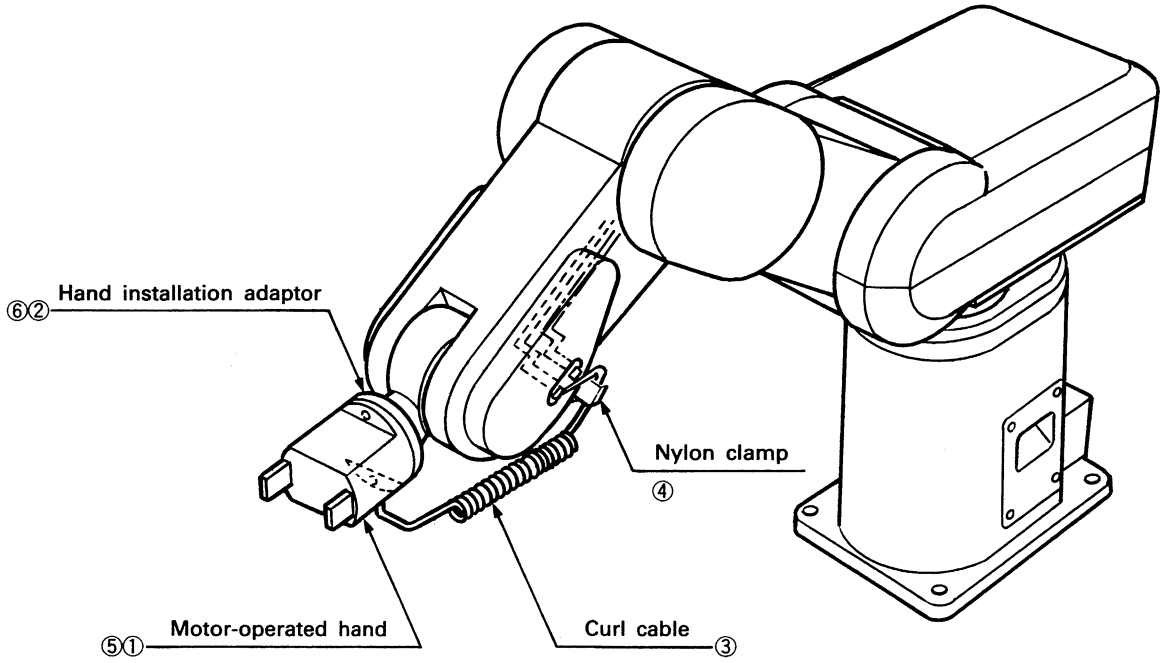


Table 2.1.1 Motor-Operated Hand Components

No.	Description	Type	Manufacturer	Remarks	Qty
①	Motor-operated hand	HM-01	Mitsubishi Electric		1
②	Hand installation adaptor	BU144D697H01	Mitsubishi Electric		1
③	Curl cable	BU144D508G51	Mitsubishi Electric		1
④	Nylon clamp	NK4N	Kitagawa Kogyo		1
⑤	Hand installation bolt	M3 × 12 (socket head bolt)			2
⑥	Adaptor installation bolt	M5 × 16 (socket head bolt)			4

Fig. 2.1.8 Motor-Operated Hand Structure

(1) Fig. 2.1.9 shows the way of installing the motor-operated hand (option).

(2) Set the wrist roll to the origin or perform jog operation to set it to the position after returning to origin (+180°).

(3) Install the hand installation adaptor to the mechanical interface of the robot with four M5 × 16 hexagon socket head bolts.

(4) Install the motor-operated hand to the hand installation adaptor with two M3 × 12 hexagon socket head bolts. At this time, the relationship between the motor-operated hand-side connector position and hand installation adaptor position must be as shown in Fig. 2.1.10. (The hand installation adaptor is at the position after returning to origin (+180°).)

(5) After the hand has been installed, secure the curl cable onto the hand-side connector and fore arm-side connector as shown in Fig. 2.1.9.
Using the nylon clamp furnished with the hand, secure the straight section of the curl cable to the fore arm together with the pulley cover (R) shown in Fig. 2.1.9 by means of the M3 × 6 truss screw securing the pulley cover (R).

CAUTION

When the hand is used, the curl cable may be caught between the hand installation adaptor and fore arm during operation of the robot depending on the positions of the wrist pitch (J4 axis) and wrist roll (J5 axis). Avoid using the hand with the robot in such a position.

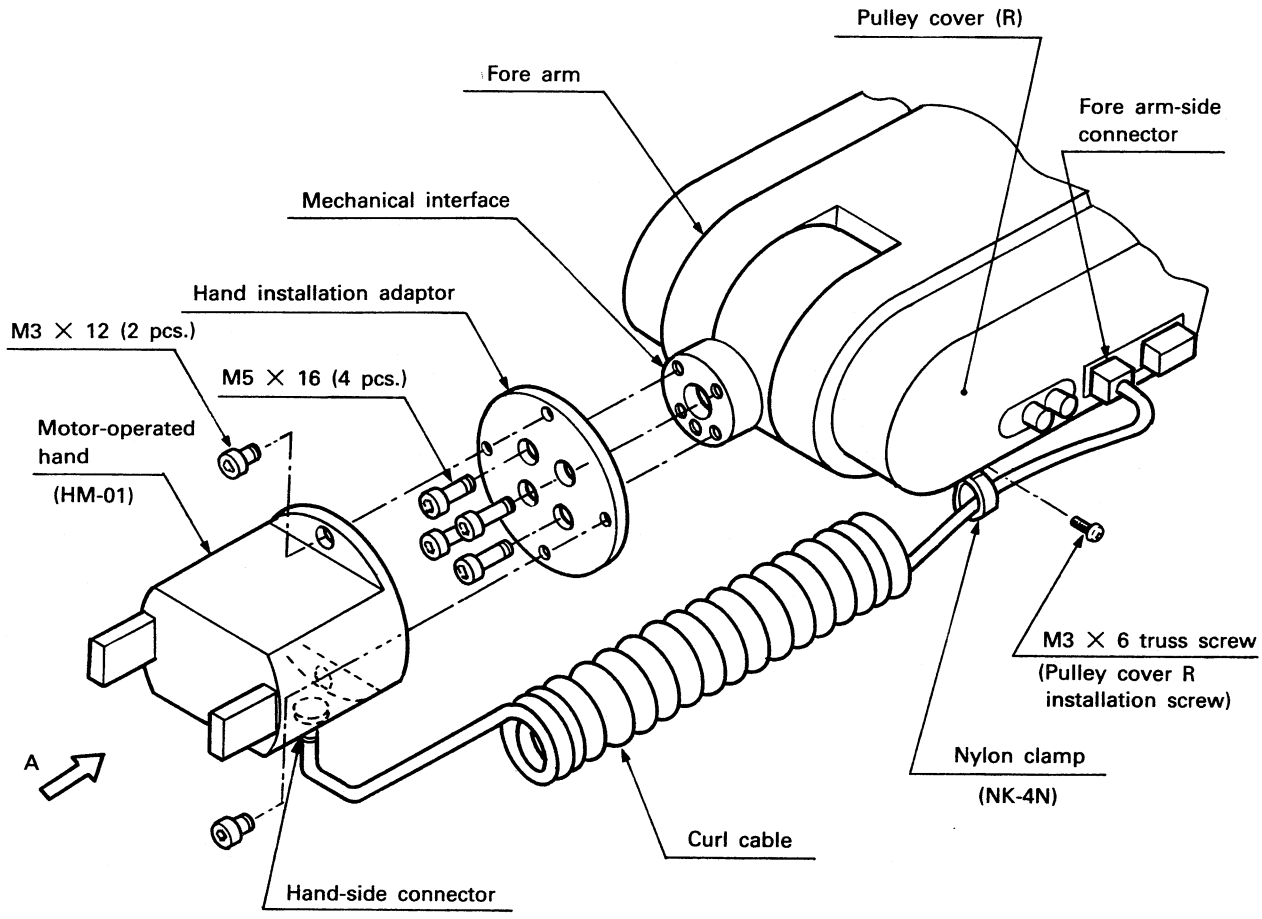
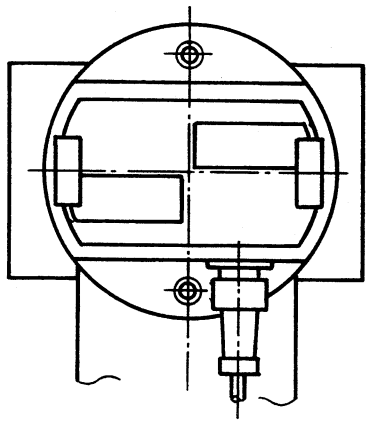


Fig. 2.1.9 Installation of the Motor-Operated Hand (Option)



View A

Fig. 2.1.10 Hand Installation Position Relationship

1.7 Installation of the Pneumatic Hand Set (Option)

The structure of the pneumatic hand set is shown in Fig. 2.1.11 and its components are indicated in Table 2.1.2.

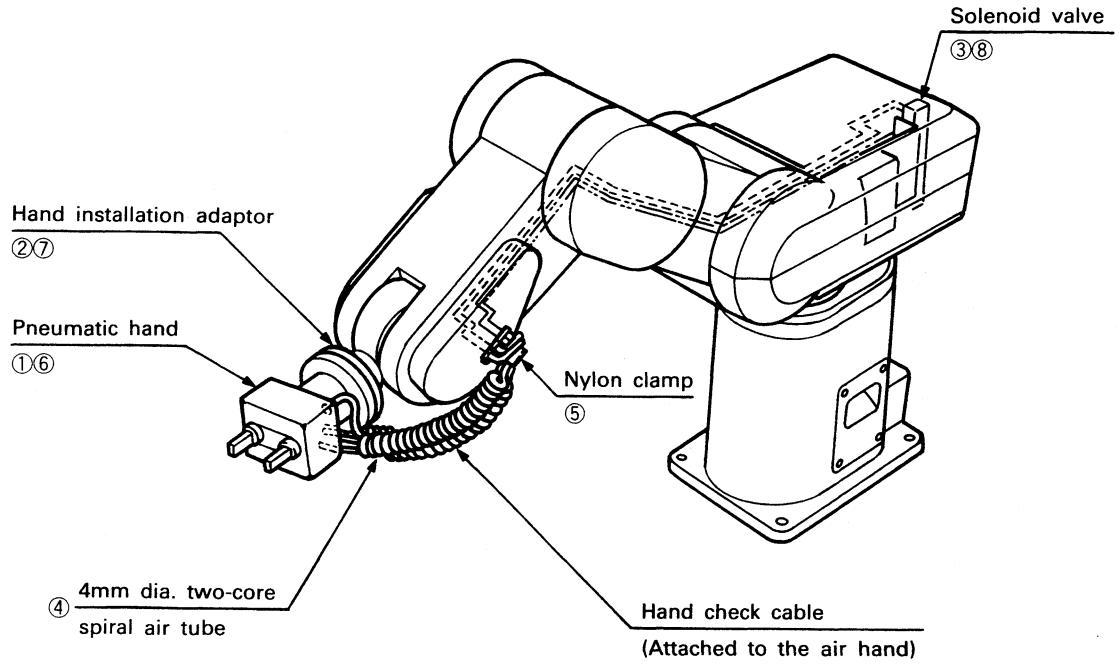


Fig. 2.1.11 Air Hand Set Structure

No.	Description	Type	Manufacturer	Remarks	Qty
①	Pneumatic hand	HM-51	Mitsubishi Electric	Including hand check cable	1
②	Hand installation adaptor	BU144D697H01	Mitsubishi Electric		1
③	Solenoid valve	110-4E1-32WY	Koganei	Mitsubishi Electric specifications	1
④	Spiral air tube	BU144D698H01	Mitsubishi Electric	4mm dia. × 2 cores	1
⑤	Nylon clamp	NK-6N	Kitagawa Kogyo		1
⑥	Hand installation bolt	M3 × 12 (socket head bolt)			4
⑦	Adaptor installation bolt	M5 × 16 (socket head bolt)			4
⑧	Solenoid installation bolt	M3 × 18 (socket head bolt)			2

Table 2.1.2 Air Hand Set Components

1.7.1 Installation of the pneumatic hand set

(1) Fig. 2.1.12 shows the way of installing the pneumatic hand (option).

(2) Set the wrist roll to the origin or perform jog operation to set it to the position after returning to origin (+180°).

(3) Install the hand installation adaptor to the mechanical interface of the robot with four M5 × 16 hexagon socket head bolts.

(4) Install the pneumatic hand to the hand installation adaptor with four M3 × 12 hexagon socket head bolts. At this time, the relationship between the pneumatic hand check cable outlet position and hand installation adaptor position must be as shown in Fig. 2.1.13. (The hand installation adaptor is at the position after returning to origin (+180°).)

(5) After the hand has been installed, secure the spiral air tube and hand check cable into the fittings coming from the pulley cover (R) and hand check connector as shown in Fig. 2.1.12. Using the nylon clamp furnished with the hand, secure the straight sections of the spiral air tube and hand check cable to the fore arm together with the pulley cover (R) shown in Fig. 2.1.12 by means of the M3 × 6 truss screw securing the pulley cover (R).

(6) Set the tool length to 108mm in accordance with Section 5.1, Vol. 2.

(7) Set the pins to the source type in accordance with Section 4.3 "Interface Setting" in APPENDICES.

CAUTION

When the hand is used, the spiral air tube and hand check cable may be caught between the hand installation adaptor and fore arm during operation of the robot depending on the positions of the wrist pitch (J4 axis) and wrist roll (J5 axis). Avoid using the hand with the robot in such a position.

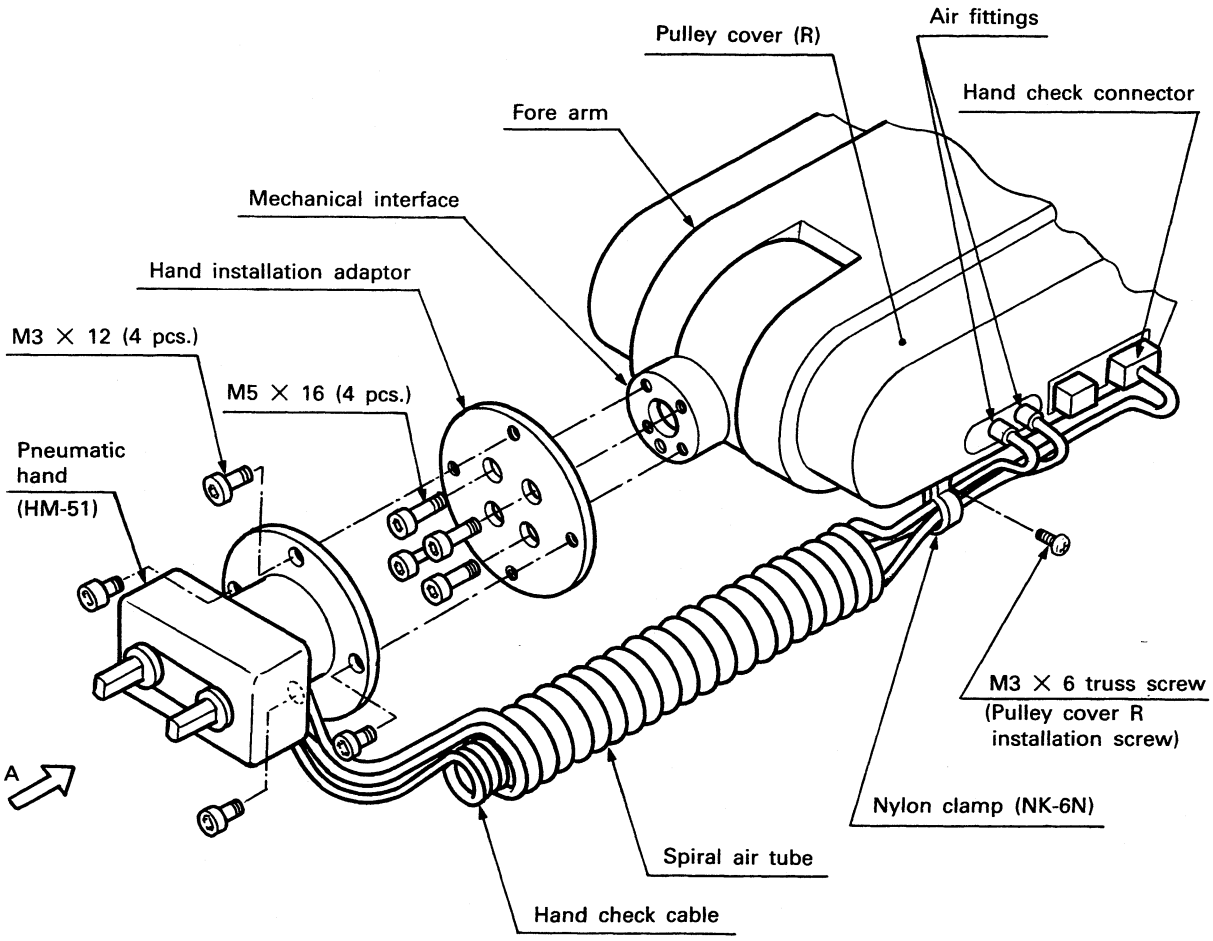
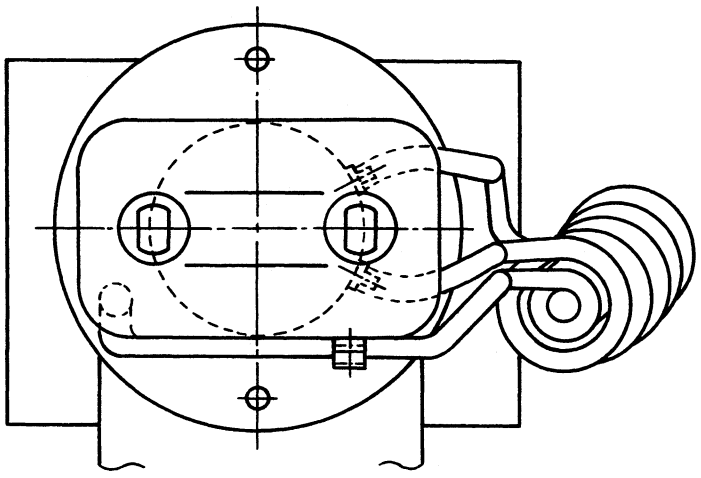


Fig. 2.1.12 Installation of the Pneumatic Hand (Option)



View A

Fig. 2.1.13 Hand Installation Position Relationship

1.7.2 Installation of the solenoid valve

(1) Fig. 2.1.14 shows the way of installing the solenoid valve (option).

(2) Remove the four M3 × 10 screws ① and remove the shoulder cover (upper).

(3) Remove the M3 × 6 hexagon socket head bolt ② to free the air tube.

(4) Install the solenoid valve by threading the two M3 × 18 hexagon socket head bolts ③ into the screw hole from which the bolt has been removed at step (3) and the screw hole below it.

(5) Disconnect the "HAND. M" connector (black) and connect the solenoid valve connector.

(6) Fit the air tube freed at step (3) into the solenoid valve fittings as shown in Fig. 2.1.14.

(7) Reinstall the shoulder cover (upper) in the reverse procedure of step (2), using care not to get the cable caught.

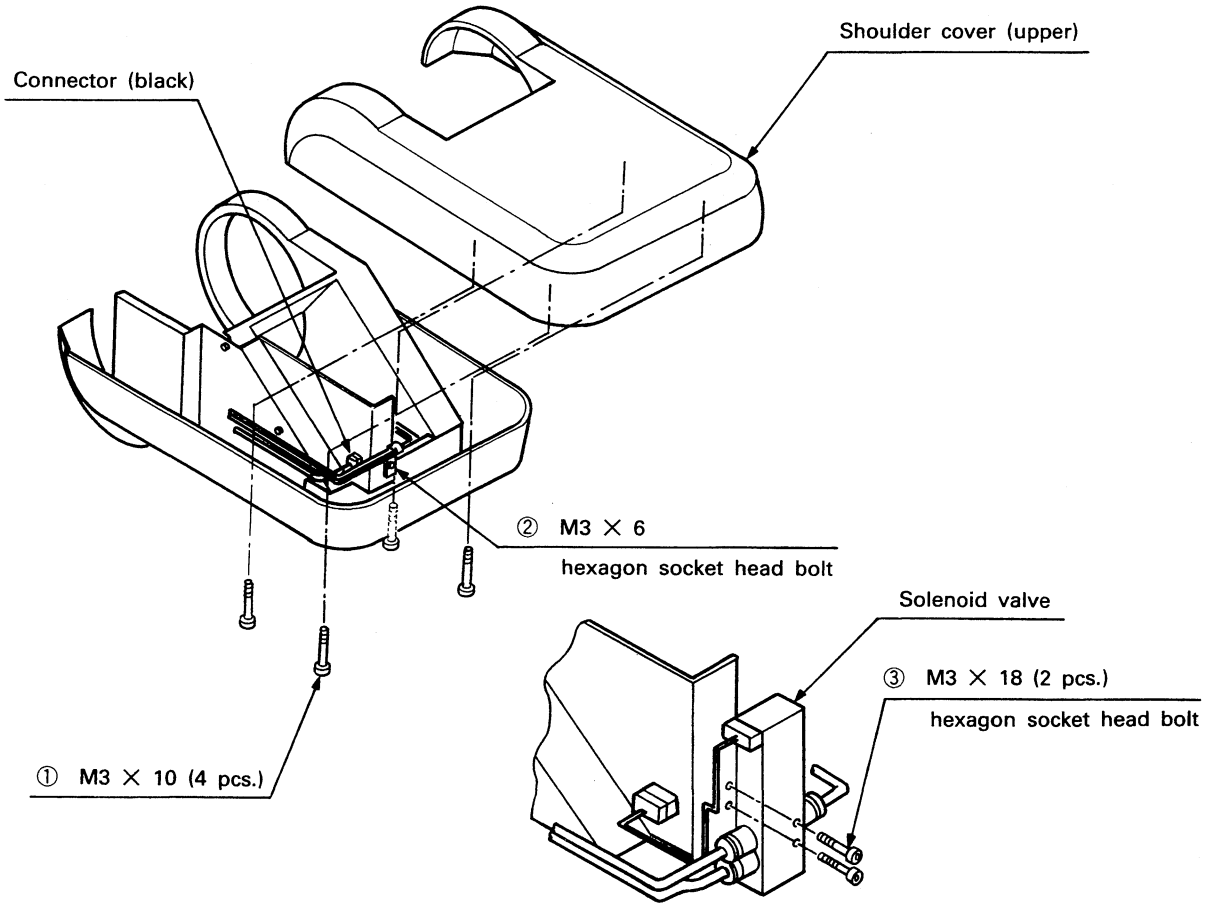


Fig. 2.1.14 Installation of the Solenoid Valve (Option)

1.7.3 Checking opening and closing of the hand

(1) Set the hand select switch of the drive unit to the AC position in accordance with "Section 2.1.3, Vol. 2".

(2) Connect the air line to the air piping port (quick fitting for 4mm dia. air tube) shown in Fig. 2.1.15. The supplied air must be clean.

(3) The RV-M2 may only be used with a single solenoid. At power off or emergency stop, therefore, the air is supplied from the port "B" of the solenoid valve. Connect the air tube taking the hand opening/closing operation at power off into consideration.

(4) By pushing the hand open [◀O▶] key on the teaching box, the air is supplied from the port "B" of the solenoid valve. Since the hand opening/closing operation at power off have been taken into consideration, the actual hand operation may be reverse to the teaching box key control.
By moving bit 3 of SW1 in accordance with "Section 2.1.2, Vol. 2", the teaching box key control can be matched with the hand opening/closing operation.

CAUTION

The supplied air must be clean.

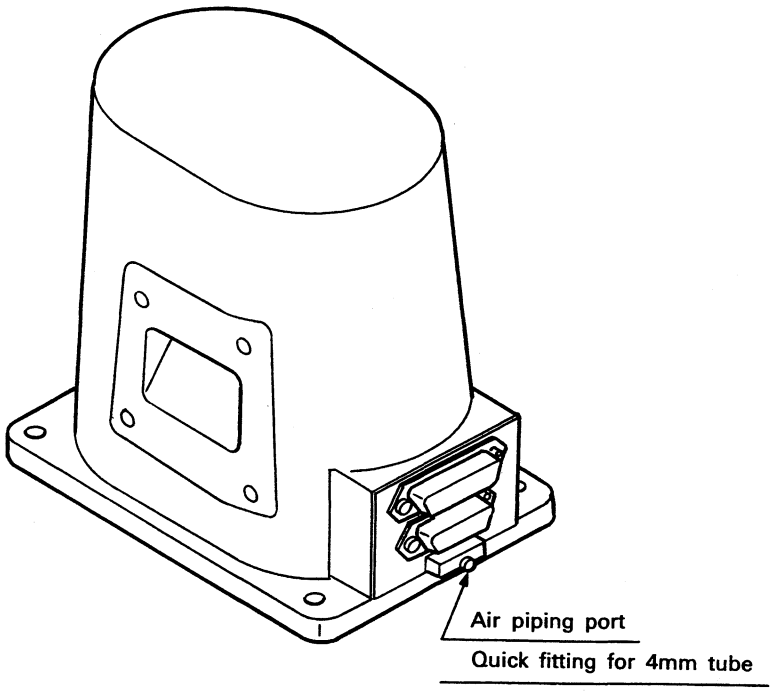


Fig. 2.1.15 Air Piping Port

1.8 Installation of the Teaching Box (Option)

When the teaching box (option) is used, plug the teaching box cable connector into the corresponding connector in the rear panel of the drive unit (lower connector on the I/O card) before powering up. At this time, raise the spring latches to secure the connector in position. (See Fig. 2.1.16.)
The connector may only be plugged or unplugged when the power of the drive unit is off.

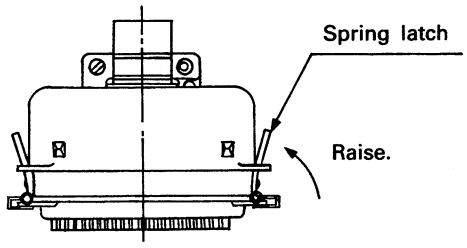


Fig. 2.1.16 Securing of the Connector

1.9 Installation of the Emergency Stop Switch

To ensure safety, the emergency stop switch must be installed at a place where it is readily accessible.

(1) Connect the emergency stop switch using the "EXT. EMG. STOP" terminals in the rear panel of the drive unit.

(2) Remove the jumper furnished and connect the COM terminal and NC terminal (normally closed) of the emergency stop switch to the terminals indicated in step (1). (The jumper must be kept for future use.)

(3) Electrical ratings of the emergency stop are 12VDC, 110mA. The wire used should be selected by fully considering its environmental resistance and strength.

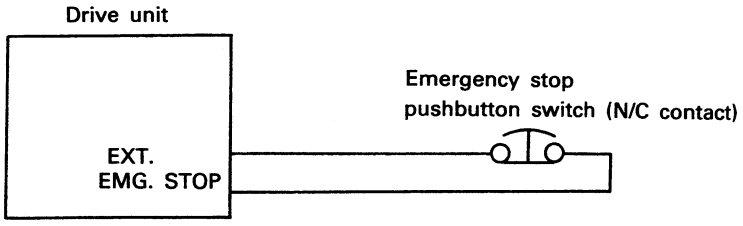


Fig. 2.17 Installation of the Emergency Stop Switch

2. BASIC FUNCTIONS OF THE SYSTEM COMPONENTS

2.1 Drive Unit

2.1.1 Functions of front control switches and LEDs

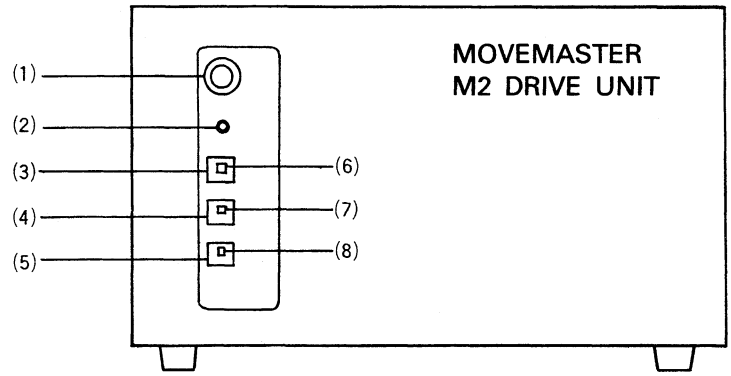


Fig. 2.2.1 Drive Unit Front Panel

(1) EMG. STOP (Emergency stop switch, red)
Emergency stop pushbutton switch with a locking mechanism. By pressing this button, the servo system is cut off and the brakes are applied to bring the robot to an immediate stop. At the same time, the error LED flickers, the buzzer beeps (if the buzzer has been set to ON), and "LED3" in the drive unit side door is lit. To reset the emergency stop, turn the switch clockwise and pull toward you.

(2) POWER (Power indicator LED, yellow)
Lit when the drive unit is on. Remains off when the fuse in the rear panel is blown.

(3) START (Start switch, green)
Starts the program or restarts it from a stop state. When the program is run, the execute LED (later described) is lit. This switch is invalid when the teaching box is valid or during "operation using external signals" mentioned later.

(4) STOP (Stop switch, red)
Stops the program and decelerates the robot to a stop. When the program is being run, the wait LED (described later) is lit. This switch is always valid.

(5) RESET (Reset switch, white)
Used to reset the program at a stop or reset an error condition if the error has occurred. This returns the program to its beginning and switches off the wait LED (described later) or switches off the error LED (described later). This switch is invalid when the teaching box is valid or during "operation using external signals" which will be mentioned below.

(6) EXECUTE (Execute LED, green)
Lit while the program is being run or a command is being executed by the personal computer or teaching box. Extinguished when the program or command is complete or the aforementioned stop switch is pressed.

(7) WAIT (Wait LED, orange)
Lit while the program is at a stop. Extinguished by pressing the above described start or stop switch.

(8) ERROR (Error LED, red)
Flickers or lit at occurrence of any error. To turn off this LED, remove the cause of the error and press the above mentioned reset switch. (See Section 10, Vol. 2.)

2.1.2 Functions of side setting switches and LEDs

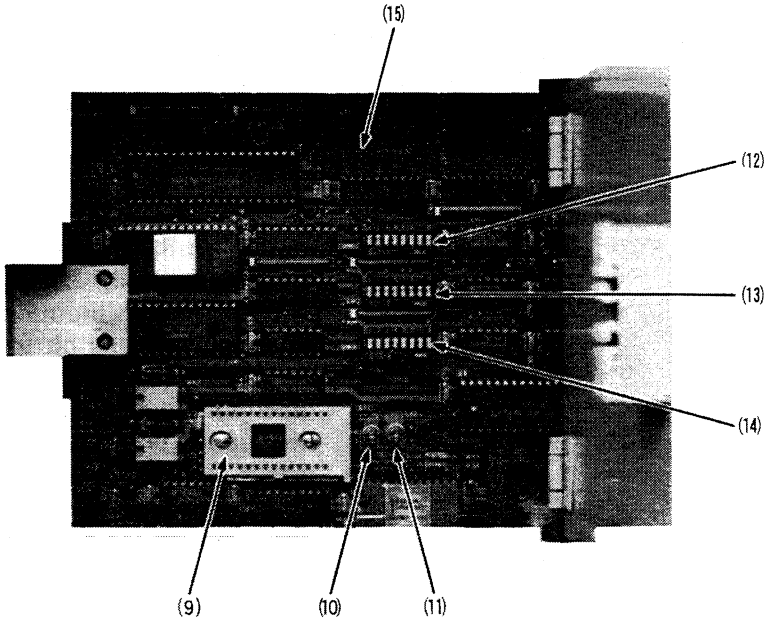


Fig. 2.2.2 Inside View of the Drive Unit Side Door

(9) SOC2 (EPROM socket for program and position data)
Used to install EPROM which allows the user-written program and position data to be transferred. The EPROM can be installed to or removed from the socket by moving the lever located under the socket. When inserting the EPROM, make sure that the indentation is positioned on the left.

(10) ST1 (Control mode select switch)
Used to select the control mode of the drive unit. (Upper position: drive unit mode; lower position: personal computer mode)

(11) ST2 (EPROM/RAM transfer switch)
Used to select whether or not data on the EPROM installed to the aforementioned SOC2 is automatically transferred to the drive unit memory (RAM) when the power is switched on. (Upper position: transferred; lower position: not transferred) Set the switch to the upper position to run a program with the data of the installed EPROM every time the power has been switched on. Set to the lower position to run a program with the data of the battery-backed RAM. If the power is accidentally switched on with the switch at the upper position, the RAM data is cleared by the EPROM data (the RAM data is also cleared when the socket is empty).

(12) SW1 (Function select DIP switch; bits numbered 1 through 8 from left to right)

Bit 1: Selects the terminator for data transmission from the drive unit through the RS232C interface. (Upper position: CR + LF; lower position: CR only) Set this bit to the upper position when the Mitsubishi MULTI16 series or MAXY is used.

Bit 2: Selects whether the RAM data is initialized or not at power on. (Upper position: not initialized; lower position: initialized) Usually set this bit to the upper position. When the memory back-up battery has been changed, set this bit to the lower position before the system is first switched on. Then set this bit to the upper position. When the power is switched on with this bit at the upper position, the RAM data is checked. If this check results in error mode I, it indicates that the RAM data is not being properly retained due to a battery fault, etc. (When the RAM data has been initialized, the position and program data is erased, the counter is reset to 0, and the initial value of the tool length is set to 123mm. The origin attitude compensation data is also erased.)

Bit 3: Selects forward or reverse direction of the hand opening/closing operation. This allows the hand opening/closing operation to be performed as commanded or reversely. (Upper position: reverse; lower position: forward)

Bit 4: Selects whether the origin setting reference position is to be set or not. (Upper position: to be set; lower position: not set) Set this bit to the upper position and set the reference position immediately after the power of the robot-drive unit system is switched on or if the drive unit used with the robot has been changed thereafter. (See Section 4.3, Vol. 2.) Set this bit also to the upper position when reading the reference position data from the EPROM to the RAM.

Bit 5: Selects whether the program start, stop and reset operations are to be performed by the drive unit front panel switches or external I/O signals. (Upper position: external signals; lower position: panel switches) When either position has been selected, the start and reset operations at the other position are made invalid with the exception of the stop operation.

Bit 6: Selects whether or not the robot brakes are to be released by the ENT key on the teaching box. (Upper position: brake release valid; lower position: invalid) Usually set this bit to the lower position. (See Section 2.2.4, Vol. 2.)

Bit 7: Selects the coordinate system definition method of the wrist roll (J5 axis) at power on. (Upper position: general angle method; lower position: articulated angle method)

Bit 8: Selects whether the buzzer is to be switched on or not when any error has occurred. (Upper position: on; lower position: off)

(13) SW2 (RS232C communication format setting switch)
See Section 2, "INTERFACE WITH PERSONAL COMPUTER (RS232C)", Vol. 5.

(14) SW3 (RS232C baud rate setting switch)
See Section 2, "INTERFACE WITH PERSONAL COMPUTER (RS232C)", Vol. 5.

(15) LEDs 1 to 7 (red), 8 to 13 (green) (Hardware error displays)
LEDs 1 to 7 are lit to indicate the cause of the corresponding hardware error (error mode I):

LED 1 ····· Servo system fault (first LED from top left)

LED 2 ····· Open or disconnected motor signal cable (second LED from top left)

LED 3 ····· Drive unit emergency stop input (third LED from top left)

LED 4 ····· Teaching box emergency stop input (fourth LED from top left)

LED 5 ····· Backup battery fault (fifth LED from top left)

LED 6 ····· Open or disconnected Locomotive unit signal cable (sixth LED from top left)

LED 7 ····· Hardware limit switch input (seventh LED from top left)

LED 8 ····· First axis fault (first LED from bottom left)

LED 9 ····· Second axis fault (second LED from bottom left)

LED 10 ···· Third axis fault (third LED from bottom left)

LED 11 ···· Fourth axis fault (fourth LED from bottom left)

LED 12 ···· Fifth axis fault (fifth LED from bottom left)

LED 13 ···· Sixth axis (Locomotive unit) fault (sixth LED from bottom left)

When the servo system fault has occurred, the teaching box also indicates the faulty axis and its error cause. (See Section 3.7, Vol. 4.)

2.1.3 Functions of rear panel switches, connectors and terminal block

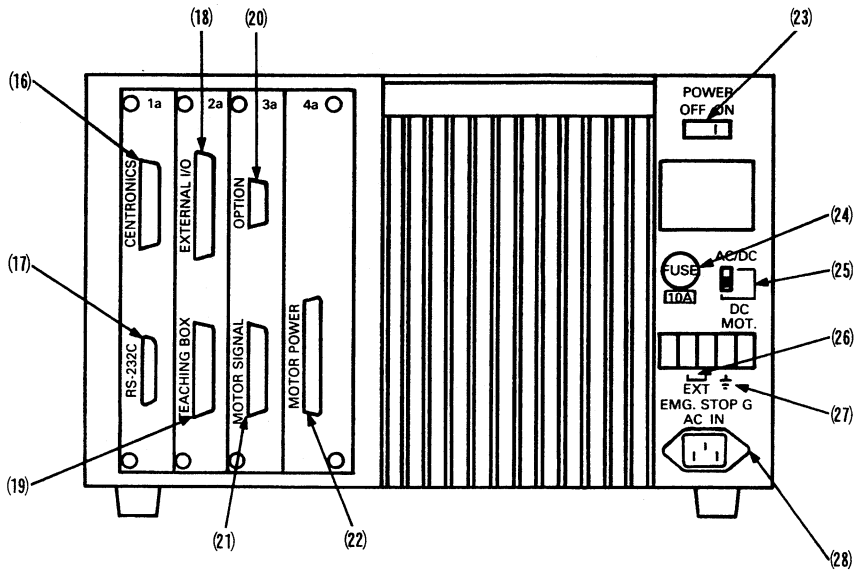


Fig. 2.2.3 Drive Unit Rear Panel

(16) CENTRONICS (Centronics connector)
For connection of the personal computer and drive unit.

(17) RS232C (RS232C connector)
For connection of the personal computer and drive unit.

(18) EXTERNAL I/O (External I/O equipment connector)
For connection of the external I/O equipment, e.g. programmable controller, and drive unit.

(19) TEACHING BOX (Teaching box connector)
For connection of the teaching box (option) and drive unit.

(20) OPTION (Locomotive unit signal cable connector)
For connection of the drive unit and Locomotive unit (option) signal line to control the Locomotive unit.

(21) MOTOR SIGNAL (Motor signal cable connector)
For connection of the drive unit and robot signal line.

(22) MOTOR POWER (Motor power cable connector)
For connection of the drive unit and robot power line.

(23) POWER (Power switch)
Power on/off switch for the drive unit.

(24) FUSE (Fuse assembly)
10A fuse for 250VAC power fitted in the fuse holder for the drive unit.

(25) AC/DC, DC MOT. (Hand select switch)
Set in accordance with the hand (option) installed to the robot. Set this switch to the "DC MOT." (lower) position to use the motor-operated hand. Set to the "AC/DC" (upper) position to use the air hand (100VAC solenoid). This switch must be set with special care because incorrect setting will damage the hand and internal circuits. After set, this switch should be secured by the vertically sliding fitment.

(26) EXT EMG. STOP (External emergency stop input terminal)
For connection of the emergency stop switch (normally closed contact) installed externally. Before using the system, remove the metal jumper which has been connected to this terminal at the factory.

(27) G (Frame ground)
Grounding terminal of the drive unit.

(28) AC IN (AC power inlet)
Inlet for plugging the drive unit power cord. (120, 220, 230 or 240VAC depending on the line voltage in your country) The ground terminal inside the inlet is internally connected with the above indicated frame ground.

2.2 Teaching Box

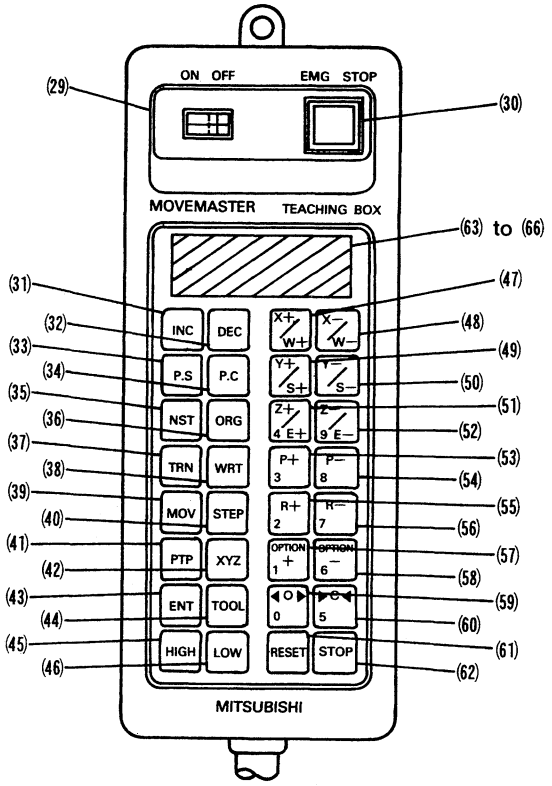


Fig. 2.2.4 Teaching Box

2.2.1 Functions of the switches

(29) ON/OFF switch
Selects whether the teaching box keys are made valid or invalid. Set to ON to operate the robot from the teaching box. Set to OFF to control the robot in program operation mode or with commands from the personal computer. Any incorrect keyed data can be canceled by setting the switch to OFF. In program operation mode, the robot cannot be operated from the teaching box if this switch is set to ON.

(30) Emergency stop (EMG. STOP) switch
Pushbutton switch to bring the robot to an emergency stop. (The signal is internally latched when this switch is pressed.) Whether the teaching box is ON or OFF, pressing this button brings the robot to an immediate stop and flickers the error LED. (Error mode I) "LED4" inside the drive unit side door also comes on.

2.2.2 Key functions

The left-hand side keys are mainly used to select mode and jog, and the right-hand side keys are mostly used to perform jog operation and enter numerals. To perform jog operation, hold down any of the jog select keys, [PTP], [XYZ] or [TOOL] on the left-hand side and press the appropriate jog key on the right-hand side. For a relationship between each jog key and the robot operation, see Section 3.1.5 Basic operations, Vol. 1. The [XYZ] and [TOOL] jog operations are invalid without carrying out origin setting described later.

(31) [INC] (+ [ENT])
Moves the robot to the predefined position of the position number greater than the current one in articulated interpolation. To move the robot sequentially, repeat the keystrokes. (See the "IP" command.)

(32) [DEC] (+ [ENT])
Moves the robot to the predefined position of the position number smaller than the current one in articulated interpolation. To move the robot sequentially, repeat the keystrokes. (See the "DP" command.)

(33) [P. S] (+ [Numeral] + [ENT])
Defines the coordinates of the current position of the robot into the position of the specified number. If a single number has been assigned to two different positions, the latter position is valid. (See the "HE" command.) Error may occur if any axis is located near the boundary of the operation space.

(34) [P. C] (+ [Numeral] + [ENT])
Deletes the defined position of the specified number. (See the "PC" command.)

(35) [NST] (+ [ENT])
Returns the robot to origin. (See the "NT" command.)

(36) [ORG] (+ [ENT])
Moves the robot to the origin-setting reference position. (See the "OG" command.)

(37) [TRN] (+ [9][9][9] + [ENT])
Transfers the contents (program and position data) of the user EPROM in the drive unit side door to the drive unit RAM. (See the "TR" command.)

(38) [WRT] (+ [9][9][9] + [ENT])
Writes the program and position data written on the drive unit RAM to the user EPROM in the drive unit side door. (See the "WR" command.)

(39) [MOV] (+ [Numeral] + [ENT])
Moves the end of the hand to the specified position in articulated interpolation. (See the "MO" command.) The moving speed is equivalent to SP8.

(40) [STEP] (+ [Numeral] + [ENT])
Runs the program step by step starting with the specified line number. To run the program sequentially from one step to another, repeat the keystrokes. At this time, the numeral need not be entered. Any error occurring during step-by-step operation results in error mode II and the line number at which the error has occurred is displayed by the indicator LED which will be described later.

(41) [PTP]
Selects the articulated jog operation. To perform the jog operation, hold down this key and press the required jog key. If this key is released, the robot continues the jog operation while the jog key is pressed.

(42) [XYZ]
Selects the cartesian coordinate jog operation. To perform the jog operation, hold down this key and press the required jog key. If this key is released, the robot continues the jog operation while the jog key is pressed.

(43) [ENT]
Terminates any of the keyed commands in 31 to 40 and effects the corresponding operation.

(44) [TOOL]
Selects the tool jog operation. To perform the jog operation, hold down this key and press the required jog key. If this key is released, the robot continues the jog operation while the jog key is pressed.

(45) [HIGH]
Sets the jog operation to high speed. Once this key is pressed, the succeeding jog operation is set to high speed.

(46) [LOW]
Sets the jog operation to low speed. Once this key is pressed, the succeeding jog operation is set to low speed. The jog operation speed is set to low when the ON/OFF switch of the teaching box is moved from OFF to ON.

(47) [X+/W+]
Moves the end of the hand in the positive direction of the X axis (to the left toward the front of the robot) in the cartesian jog operation. Sweeps the waist in the positive direction (clockwise as viewed from the top of the robot) in the articulated jog operation.

(48) [X−/W−]
Moves the end of the hand in the negative direction of the X axis (to the right toward the front of the robot) in the cartesian jog operation. Sweeps the waist in the negative direction (counterclockwise as viewed from the top of the robot) in the articulated jog operation.

(49) [Y+/S+]
Moves the end of the hand in the positive direction of the Y axis (to the front of the robot) in the cartesian jog operation. Swivels the shoulder in the positive direction (upward) in the articulated jog operation.

(50) [Y−/S−]
Moves the end of the hand in the negative direction of the Y axis (to the rear of the robot) in the cartesian jog operation. Swivels the shoulder in the negative direction (downward) in the articulated jog operation.

(51) [Z+/E+ 4]
Moves the end of the hand in the positive direction of the Z axis (straight upward) in the cartesian jog operation. Turns the elbow in the positive direction (upward) in the articulated jog operation. Advances the hand in the tool jog operation. Also used to enter the numeral of "4".

(52) [Z−/E− 9]
Moves the end of the hand in the negative direction of the Z axis (straight downward) in the cartesian jog operation. Turns the elbow in the negative direction (downward) in the articulated jog operation. Retracts the hand in the tool jog operation. Also used to enter the numeral of "9".

(53) [P+ 3]
Turns the end of the hand in the positive direction (upward), while maintaining its current position determined by the "TL" command, in the cartesian jog operation. Bends the wrist (wrist pitch) in the positive direction (upward) in the articulated jog operation. Also used to enter the numeral of "3".

(54) [P− 8]
Turns the end of the hand in the negative direction (downward), while maintaining its current position determined by the "TL" command, in the cartesian jog operation. Bends the wrist (wrist pitch) in the negative direction (downward) in the articulated jog operation. Also used to enter the numeral of "8".

(55) [R+ 2]
Twists the wrist (wrist roll) in the positive direction (clockwise toward the hand installation surface). Also used to enter the numeral of "2".

(56) [R− 7]
Twists the wrist (wrist roll) in the negative direction (counterclockwise toward the hand installation surface). Also used to enter the numeral of "7".

(57) [OPTION+ 1]
Moves the optional axis in the positive direction. Also used to enter the numeral of "1".

(58) [OPTION+ 6]
Moves the optional axis in the negative direction. Also used to enter the numeral of "6".

(59) [◀O▶ 0]
Opens the hand gripper. Also used to enter the numeral of "0".

(60) [▶C◀ 5]
Closes the hand gripper. Also used to enter the numeral of "5".

(61) [RESET]
Resets the program at a stop and cancels the error status. This switch provides the same function as that of the reset switch on the drive unit front panel. When the ON/OFF switch of the teaching box is ON, the teaching box reset switch is valid.

(62) [STOP]
Stops the program and decelerates the robot to a stop. This switch provides the same function as that of the stop switch on the drive unit front panel and is always valid independently of the ON/OFF switch of the teaching box.

2.2.3 Functions of the indicator LED

The 4-digit LED shows the following information:

(63) Position number
Indicated in three digits when the [INC], [DEC], [P. S], [P. C] or [MOV] key is being used.

(64) Program line number
Indicated in four digits when the [STEP] key is being used or when the program is running. A faulty line number is also indicated if an error has occurred during program operation.

(65) Teaching box status indicator (first digit from the left)
"⌊⌋" indicates that processing invoked by pressing the ENT key is in progress or has ended.
"⌊" indicates that processing invoked by pressing the ENT key cannot be performed.

(66) Axis number and cause of servo system fault
If a servo system fault has occurred, the faulty axis and error cause are displayed. (See Section 3.7, Vol. 4.)

2.2.4 Releasing the brakes

When the system power is off or when error mode I has occurred, the J2 and J3 axes of the robot cannot be moved from outside the robot because brakes are applied to those axes. However, the following procedure allows the brakes to be released and the axes moved from the teaching box. This procedure is useful when servicing the robot or positioning it for packing.

(1) After switching on the power, press the emergency stop switch of the teaching box to enter error mode I.

(2) Set bit 6 of the switch SW1 in the drive unit side door to the upper (ON) position.

(3) Press the [ENT] key of the teaching box to release the brakes of the J2 and J3 axes. At this time, the robot arm must be supported by your hands so that it will not drop by its own weight. The brakes of the J2 and J3 axes are released at the same time. The brakes are reapplied by releasing the [ENT] key.

(4) After the required job is complete, return bit 6 of SW1 to the lower (OFF) position.

2.2.5 Keys and corresponding intelligent commands

There are the following relationships between the following teaching box keys and intelligent commands executed from the personal computer:

[INC]	⟷	"IP"	[DEC]	⟷	"DP"
[P.S]	⟷	"HE"	[P.C]	⟷	"PC"
[NST]	⟷	"NT"	[ORG]	⟷	"OG"
[TRN]	⟷	"TR"	[WRT]	⟷	"WR"
[MOV]	⟷	"MO"	[RESET]	⟷	"RS"
[◀O▶]	⟷	"GO"	[▶C◀]	⟷	"GC"

3. PRE-START UP INFORMATION

This section gives an overview of operation and programming as pre-start up information.

3.1 System Configuration

The Movemaster system may be configured in two different ways as described below.

3.1.1 System centering around a personal computer

In this system, the Movemaster is connected with a personal computer and operated from the personal computer. The personal computer invokes the robot axis motions by the intelligent commands provided in the Movemaster. Accordingly, the personal computer acts as a brain that causes the robot to perform a variety of tasks including assembly and experimentation.
When used with a whole range of peripheral devices (e.g. printer, X-Y plotter, external storage and sensors), the Movemaster can easily configure an expandable, enhanced system. The system will also become highly flexible since all robotic motions are effected by the program written with the personal computer.
Possible application areas include training and research programs, preliminary assessment of a robotic system before making a decision to invest in it, and laboratory automation. This configuration corresponds to the personal computer control mode to be described later in this manual.

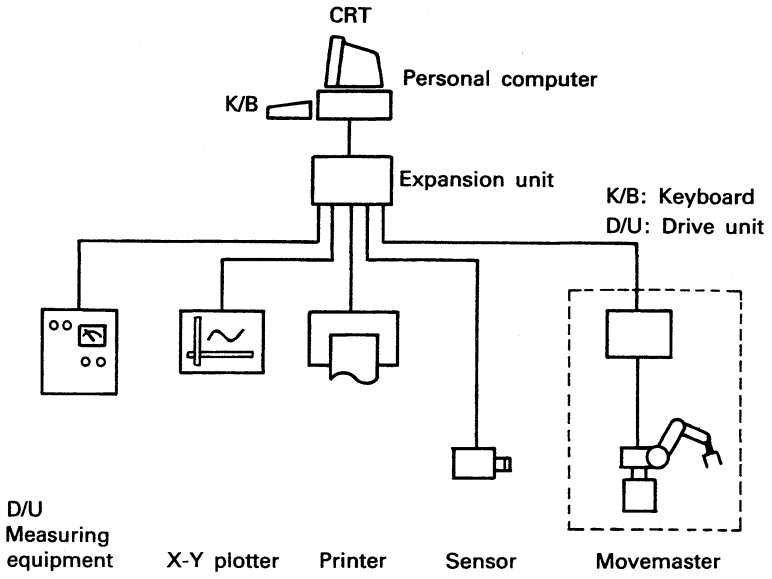


Fig. 2.3.1 Typical System Centering Around the Personal Computer

3.1.2 System centering around the drive unit

In this system, the drive unit is used to drive the Movemaster and the personal computer is only used for programming purposes. The program written with the personal computer is transferred to the drive unit for running the robot later. This means that you need not install a personal computer on the actual production floor. Signal exchanges between the robot and peripheral equipment such as limit switches, relays, LEDs, and programmable controllers are accomplished through the external I/O port in the drive unit. The program stored in the built-in EPROM can be easily changed by just exchanging the existing EPROM for a new one. Application areas include the production lines and inspection stations in the plants.
The configuration corresponds to the drive unit control mode to be described later in this manual.

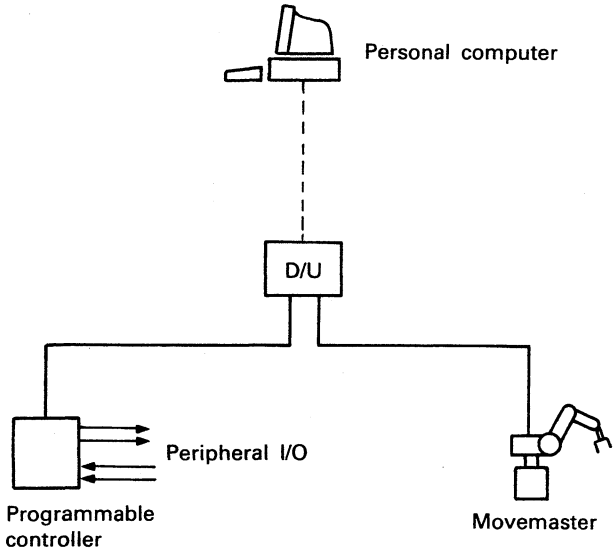


Fig. 2.3.2 Typical System Centering Around the Drive Unit

3.2 Robot-Computer Link

The drive unit makes available two types of interfaces for the link between the Movemaster and a personal computer. The following paragraphs outline the features of each interface. Study the personal computer and use the appropriate interface according to your personal computer and application. For further details, see Sections 1 or 2 "INTERFACE WITH THE PERSONAL COMPUTER, Vol. 5".

3.2.1 Centronics interface

This is originally the parallel standard for printers established by Centronics Corporation. Most printers and X-Y plotters currently in use support this standard. The personal computer sends 8 bits simultaneously, or in parallel, and the dedicated signal lines control the flow of data.
Though restricted to small distances of 1 to 2 meters, the parallel transmission ensures a transmission speed faster by far and requires no special settings, thus allowing for ease of applications.
The Movemaster has the interface equivalent to that used in the printer, i.e. the data transfer is only one-way from the personal computer to the robot. Also, part of the intelligent commands (those requesting for read of data on the robot side, including WH, PR and LR) cannot be used.
Data communication is done by the LPRINT statement in BASIC.

3.2.2 RS232C interface

The RS232C interface was originally the standard for data communication equipment using telephone lines and has evolved into the serial data transmission standard for the computer and its peripheral equipment.
Since data is sent along a single wire, or channel, one bit at a time, it takes longer than in the parallel transmission if the baud rate is low. Settings on the robot must also match those on the personal computer and not all personal computers can be employed.
Its bidirectional data transfer capability, however, enables the personal computer to read the robot's internal data. Serial also permits a transmission distance longer than parallel, as long as 3 to 15 meters, and allows configuration even when the Centronics port of the personal computer has been occupied by a printer or other peripheral device.
In BASIC, data communication is done by the OPEN, PRINT #, and LINE INPUT # statements.

3.3 Control Modes

The Movemaster has the following two control modes, personal computer mode and drive unit mode:

3.3.1 Personal computer mode

[Setting procedure]
Set the toggle switch (ST1) in the drive unit side door to the lower position.

[Explanation]
This mode allows the personal computer to execute the intelligent commands directly, write and transfer a program, and start the program transferred to the drive unit RAM (mode corresponding to Section 3.1.1, System centering around a personal computer). The operation in this mode is divided into the following three phases just as in the general BASIC.
In the following operations, the ON/OFF switch of the teaching box must be in the OFF position.

(1) Direct execution
This phase directly executes the intelligent commands of the Movemaster. For example, to move the robot to a previously taught point (position 1) using the command "MO" (move), the character string:

"MO 1" (Move to position 1)

is sent in ASCII. (Any lower-case letter will be converted internally into a corresponding upper-case letter.) This corresponds to:

LPRINT "MO 1"

for the Centronics interface, and:

PRINT #1, "MO 1"

for the RS232C interface (space may be omitted).
The commands sequentially sent in this phase are executed one by one and they do not form into a program stored in the drive unit.

(2) Program generation
The personal computer in this phase generates a program using the Movemaster commands. The program is stored in the drive unit RAM.
For example, to write a program for the robotic motion effected above, the character string:

"10 MO 1"

is sent in ASCII; where the first number, "10", represents the Movemaster program line number, which identifies the order of storage in memory like those used in the general BASIC. The program is executed in order of line numbers. Therefore, a line number must be assigned to the beginning of each line when writing a program. (Line numbers 1 to 3584 may be used. Any line number greater than 3584 will result in an error.) The Centronics equivalent to the above is:

LPRINT "10 MO 1"

while the RS232C equivalent is:

PRINT #1, "10 MO 1"

where the space may be omitted.
At this time, the transferred program is not executed and is only stored into the RAM of the drive unit.

(3) Program execution

In this phase, the program stored in the drive unit RAM is executed. The program is started by sending the command "RN" which corresponds to "RUN", the program start command in BASIC. The Centronics equivalent to this command is:

```
LPRINT "RN",
```

while the RS232C equivalent is:

```
PRINT #1, "RN".
```

Some of the typical programs are shown below.

Example 1: Direct execution (Centronics)

```
100 LPRINT "NT"          ; Origin setting (nesting).
110 LPRINT "SP 7"        ; Sets speed to 7.
120 LPRINT "MO 10, O"    ; Moves to position 10 with hand opened.
130 LPRINT "GC"          ; Closes hand (grip close).
140 LPRINT "MO 11, C"    ; Moves to position 11 with hand closed.
150 END                  ; Ends BASIC program.
RUN                      ; Runs BASIC program
OK
```

In the above example, hitting "RUN" causes each line (numbers 100 to 150) of the BASIC program to be executed, which in turn causes each of the Movemaster commands to be directly executed sequentially (with resultant robot motions).

Example 2: Program generation to execution (Centronics)

```
100 LPRINT "10 NT"
110 LPRINT "12 SP 7"
120 LPRINT "14 MO 10, O"
130 LPRINT "16 GC"
140 LPRINT "18 MO 11, C"
150 LPRINT "20 ED"       ; Ends Movemaster program.
160 END                  ; Ends BASIC program.
RUN                      ; Runs BASIC program
OK
LPRINT "RN"              ; Runs Movemaster program.
```

In the above example, hitting "RUN" causes each line (numbers 100 to 160) of the BASIC program to be executed, which in turn causes the Movemaster program (line numbers 10 to 20) to be transferred to the drive unit. Note that, at this time, the robot does not start its motion.
When "RN" is then transferred, it results in the robotic motion.

3.3.2 Drive unit mode

[Setting procedure]
Set the toggle switch (ST1) in the drive unit side door to the upper position.

[Explanation]
This mode allows the the program stored in the drive unit EPROM or battery-backed RAM to be executed (corresponding to Section 3.1.2, System centering around the drive unit). In this mode, the front control switches of the drive unit or the external I/O signals (only in external mode) can be used to start, stop and reset the program. Any command sent from the personal computer is not honored.

4. POWERING-UP TO ORIGIN SETTING

After the robot has been installed properly, perform the following procedures.

4.1 Checking the Side Setting Switches

Before switching the system power on, check that the side setting switches are set as indicated below in accordance with Section 2.1.2 Functions of side setting switches and LEDs, Vol. 2:

SW1: Bit 2 is at the upper position (RAM data not initialized).
SW1: Bit 4 is at the upper position (origin attitude setting)
ST1: Lower position (personal computer mode)
ST2: Lower position (data not transferred to EPROM/RAM.)

4.2 Powering Up

Power up the robot by setting the power switch on the drive unit rear panel to ON. The power indicator LED on the front panel of the drive unit then lights up.

4.3 Origin Attitude Setting

This operation must be performed before the customer controls the robot for the first time in order to compensate for the mechanical variations of the robot and use the robot with high accuracy. It must also be performed when the drive unit used with the robot has been changed. Perform origin attitude setting in the following procedure using the teaching box:

1) Set bit 4 of SW1 in the drive unit side door to the upper position. (Or check the above.)

2) Set the ON/OFF switch of the teaching box to ON.

3) Press the [NST] and [ENT] keys of the teaching box in that order. The robot starts the operation indicated in Section 3.1.6 Origin setting, Vol 1. If there is a possibility of hitting any obstacle around, the robot must be moved to a safe attitude beforehand by jog operation. The "EXECUTE" LED on the drive unit front panel is kept lit during the operation. Wait until the "EXECUTE" LED is extinguished.

4) Perform jog operation (PTP) from the teaching box so that all the match marks (triangle marks) of the J1 to J5 axes applied to the robot correspond to each other. (See Fig. 2.4.1.)

5) After positioning is complete, press the [P. S], [O] and [ENT] keys of the teaching box in that order.

6) Set bit 4 of SW1 to the lower position.

7) Press the [NST] and [ENT] keys on the teaching box successively to return the robot to the origin again.

By the above procedure, the mechanical compensation data unique to your robot is stored into the battery-backed RAM of the drive unit used with the robot. The operation in the above step 7) corresponds to the origin setting operation described in Section 4.4.

REMARKS

The compensation data stored in the RAM can be written and stored into the EPROM. To transfer the data from the EPROM to the RAM again, set bit 4 of SW1 to the upper position and press the [TRN] key of the teaching box or provide the "TR" command from the personal computer.

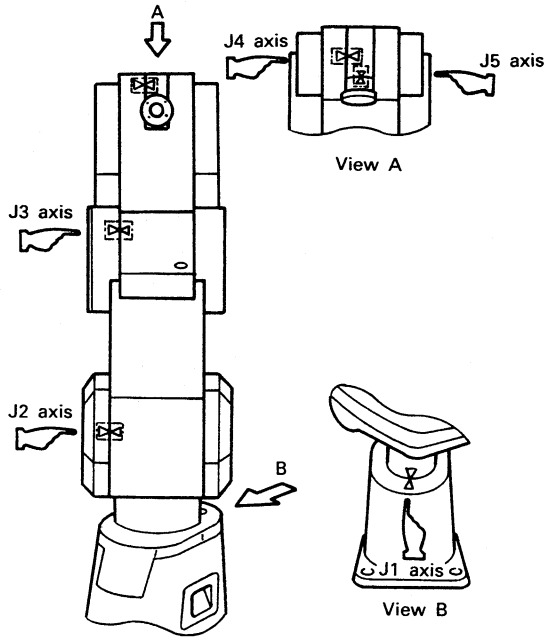


Fig. 2.4.1 Origin Attitude Setting

4.4 Origin Setting

Every time the power has been switched on, the robot must be returned to origin as described below. This operation is done to match the mechanical origin of the robot with that of the drive unit control system and is required only once after the power is switched on. (Check that bit 4 of the drive unit side setting switch SW1 is at the lower position.)

Using the teaching box

1) Set the ON/OFF switch of the teaching box to ON.

2) Press the [NST] and [ENT] keys in that order.

Using intelligent commands through personal computer

1) Set the ON/OFF switch of the teaching box to OFF.

2) Execute the "NT" command in direct mode.
   LPRINT "NT" (Centronics)
   PRINT #1, "NT" (RS232C)

At this time, the axes operate in the following order as indicated in Section 3.1.6 Origin setting, Vol.1:

1) The J2, J3 and J4 axes move to their respective mechanical origins.

2) Then, the J1 and J5 axes move to their respective mechanical origins.

Therefore, the robot must be moved to a safe position beforehand by jog operation to prevent it from striking any surrounding object depending on the attitude before origin setting. Hand control is also started together with the origin setting. Care must therefore be taken so that the hand may not drop the object it is holding. (The open/close setting can be changed with bit 3 of SW1.)
After the origin setting is complete, check that the robot axes stop at their corresponding match marks indicated in Section 4.3 "Origin attitude setting", Vol. 2. (The origin attitude setting should be done once more if any axis does not match its match mark.)

REMARKS

An error will occur if the origin setting operation is performed without setting the origin attitude.

5. POSITION SETTING

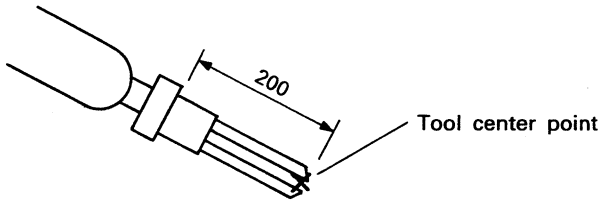
After the origin setting is complete, set positions in the following procedure:

5.1 Setting the Tool Length

To define positions using the Movemaster, the position of the end of the hand (tool center point) must first be established. This procedure is not necessary when the optional motor-operated hand is used, since the position represents a point about 123mm away from the hand installation surface, which is defaulted to when the system is initialized.
However, the end position (tool center point) must newly be defined if the hand other than the above type is used or an additional tool is attached to the end of the motor-operated hand to execute a command, e.g. pallet command, in the cartesian coordinate system. This is because all position data in the Movemaster is represented by the "position and attitude" of the end of the hand in the cartesian coordinate system. To set the tool length, use the "TL" command through the personal computer. After setting the tool length, teach the robot through the desired positions. (See Command "TL", Vol. 3.)

Example: Tool length of 200mm

```
LPRINT "TL 200" (Centronics)
PRINT #1, "TL 200" (RS232C)
```



REMARKS

When positions have been taught for a new tool length, be sure to check that the tool length for teaching is the same as that for moving before moving the robot to those positions. If they are different, the robot may interfere with any surrounding object. (See Command "WT", Vol. 3.)
Once set, the tool length is stored in the battery-backed RAM unless the RAM is initialized.

Example:

Teaching:	**Position 1 set for the tool length of 200mm.**
Programming:	**1 TL 200 2 NT 3 MO 1**
Direct execution:	**TL 200 ↓ MO 1 ↓**

5.2 Defining, Checking, Changing and Deleting the Positions

Typical operations from the teaching box are as follows. Set the teaching box ON/OFF switch to ON.

(1) Defining positions

Define three positions as follows:

1) Press the appropriate jog key(s) to move the end of the arm to an appropriate position. (Press the PTP, XYZ or TOOL jog select key at the same time.)

2) To define this as position "10" press the following keys:

[P. S] [1] [0] [ENT]

This sets position 10. At this time, the open/close position of the hand is also defined.

3) In the same way, define positions 11 and 12 by repeating steps 1) and 2) above.

(2) Checking positions

Check that the above defined positions are correct.

1) To check position 10, press the following keys:

[MOV] [1] [0] [ENT]

If the position is correct, the end of the arm moves to the point mentioned above. The hand is opened/closed before the arm moves.

2) In the same manner, check positions 11 and 12.

(3) Changing positions

Change or redefine the already defined positions.

1) Move the arm end to a position other than position 10 and press the following keys:

[P. S] [1] [0] [ENT]

This clears the old position data and redefines position 10.

2) In the same procedure, redefine positions 11 and 12.

(4) Deleting positions

Delete unnecessary positions as follows:

1) To delete position 10 data, press the following keys:

[P. C] [1] [0] [ENT]

This clears position 10 and makes it available for new definition.

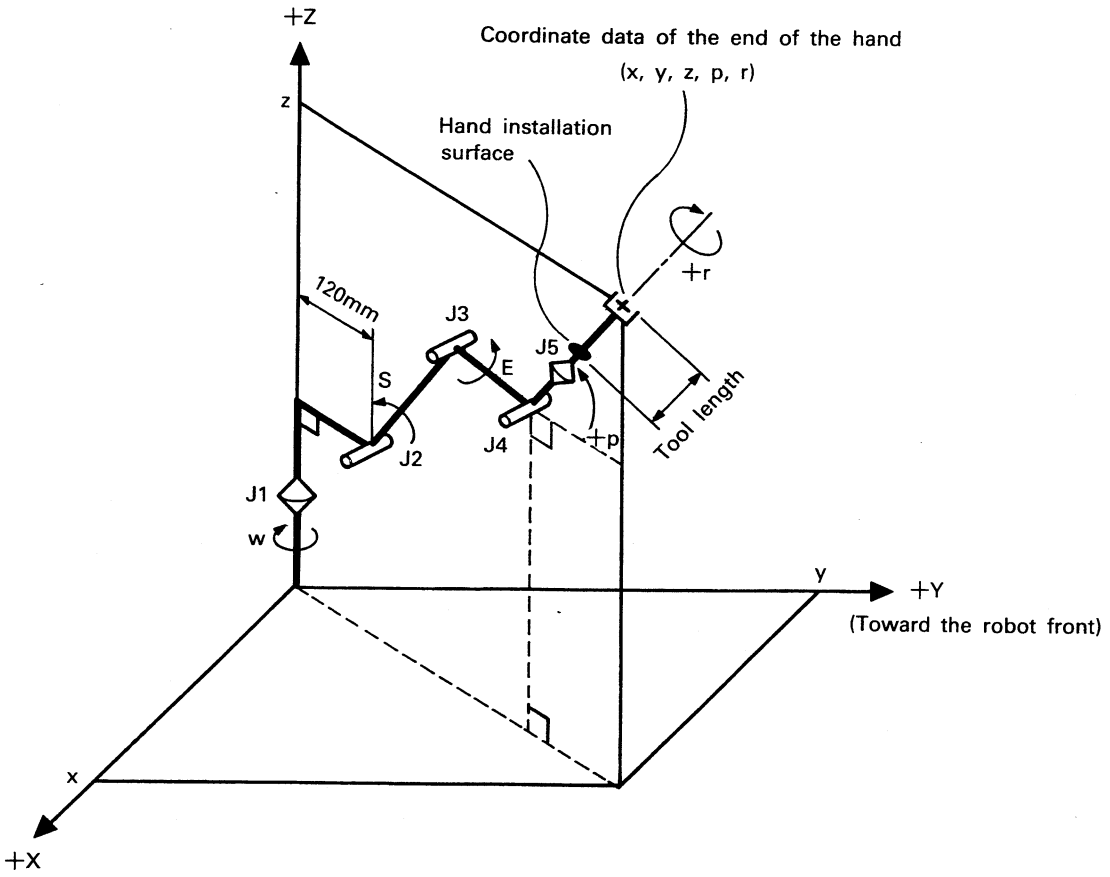
2) To check that position 10 has been deleted, press the following keys:

[MOV] [1] [0] [ENT]

If the position data has been properly deleted, the teaching box status indicator LED shows "□" which indicates that the function invoked cannot be performed.

5.3 Coordinate System of the Robot

The coordinate system of the Movemaster is defined as follows. All positions taught are represented as "position and attitude" of the end of the hand in this coordinate system. PTP, XYZ and TOOL jog operations are also performed according to this coordinate system. As to the coordinate system of the wrist roll (J5 axis), either of two representation systems (general angle system, articulated angle system) may be selected. For more information, see Section 8.3 "Selection of Wrist Roll (J5 Axis) Coordinate System" in APPENDICES.



where, x, y, z = position coordinates of the end of the hand (mm)
p = hand bending angle with respect to the X-Y plane (degrees)
r = hand twisting angle (degrees)

The coordinate data of the positions taught and the current coordinate data of the end of the hand can be read to the personal computer through the RS232C interface by using the PR (Position Read) and WH (where) commands, respectively. (See Commands PR and WH.)

6. PROGRAM GENERATION AND EXECUTION (PERSONAL COMPUTER MODE)

The following procedures are required to generate a program in personal computer mode by using the predefined positions and to execute it.
Start the procedures after setting the teaching box ON/OFF switch to OFF.

6.1 Generating and Transferring a Program

In personal computer mode described earlier, write a simple program using the three positions, 10, 11 and 12. The following lists the program sequence, where the numbers at the beginning indicate the Movemaster program line numbers. For more information on the commands, see Vol. 3.

Movemaster program

```
10  NT                  ; Origin setting.
12  SP  7               ; Sets speed to 7.
14  MO  10, O           ; Moves to position 10 with
                          hand opened.
16  MO  11, C           ; Moves to position 11 with
                          hand closed.
18  MO  12, C           ; Moves to position 12 with
                          hand closed.
20  TI  30              ; Stops operation for 3
                          seconds.
22  GT  14              ; Jumps to line number 14.
```

With the Centronics interface, execute the following BASIC program to transfer the above program to the drive unit. Note that the numbers at the beginning show the BASIC line numbers.

BASIC program

```
100  LPRINT  "10  NT"
110  LPRINT  "12  SP  7"
120  LPRINT  "14  MO  10, O"
130  LPRINT  "16  MO  11, C"
140  LPRINT  "18  MO  12, C"
150  LPRINT  "20  TI  30"
160  LPRINT  "22  GT  14"
170  END                  ; Ends BASIC program.
RUN ↓                     ; Runs BASIC program.
OK
```

For the RS232C interface, execute the following BASIC program. For details of the settings, see Section 2 "INTERFACE WITH THE PERSONAL COMPUTER (RS232C)," Vol. 5.

```
100  OPEN
     "COM1:9600,E,7,2"
     AS  #1                    ; For Mitsubishi MAXY.
110  PRINT  #1,"10  NT"
120  PRINT  #1,"12  SP  7"
130  PRINT  #1,"14  MO  10, O"
140  PRINT  #1,"16  MO  11, C"
150  PRINT  #1,"18  MO  12, C"
160  PRINT  #1,"20  TI  30"
170  PRINT  #1,"22  GT  14"
180  END                       ; Ends BASIC program.
RUN ↓                          ; Runs BASIC program.
OK
```

These sequences of operations cause the program to be transferred to the drive unit memory RAM.

6.2 Executing the Program

6.2.1 Step execution

The generated program may be executed, line by line, by operating the teaching box keys for verification in the following procedure:

1) Set the teaching box ON/OFF switch to ON.
2) To execute the program starting with line number 10, press the following keys:

   [STEP] [1] [0] [ENT]

   The command "NT" on line number 10 is executed.
3) After the command "NT" has been executed, the teaching box LED shows the subsequent program line number (in this case, "0012"). To execute line number 12, press:

   [STEP] [ENT]

   This causes the command "SP7" on line number 12 to be executed.
4) Repeat step 3) to check the program, line by line. (The line number need not be entered.)

   The step execution speed is internally set to SP8 or less.

6.2.2 Starting the program

In personal computer mode, the generated program can be initiated from the personal computer in the following procedure:

1) Set the teaching box ON/OFF switch to OFF.
2) Execute the command "RN" to directly run the program.

   Centronics ····· LPRINT "RUN"

   RS232C····· OPEN "COM1 ;9600,E,7,2" AS #1
   PRINT #1, "RN"
   CLOSE

6.2.3 Stopping and restarting the program

The program currently being run can be stopped and restarted from the front panel of the drive unit.

Stopping: Press the [STOP] switch to decelerate the robot to a stop. (This may also be effected by the [STOP] key on the teaching box.)

Restarting: Press the [START] switch to resume the robot operation from where it had stopped.

By setting the teaching box ON/OFF switch to ON during a stop, operation can be effected from the teaching box. To restart the program, set the ON/OFF switch to OFF and then press the [START] switch. Note also that no command can be executed from the personal computer even in personal computer mode while the program is at a stop. To execute any command, perform the following reset operation.

6.2.4 Stopping and resetting the program

The program currently being run can be stopped and reset from the front panel of the drive unit. When reset, the program returns to its beginning. If the program is terminated by the command "ED", it ends normally.

Stopping: Press the [STOP] switch to decelerate the robot to a stop.

Resetting: Press the [RESET] switch to reset the program. (This may also be effected by the [RESET] key on the teaching box when the teaching box is ON.)

To restart the program after resetting in personal computer mode, the command "RN" must be used. At this time, the [START] switch does not start the program. Note also that the general-purpose I/O outputs are not reset.

7. WRITING THE PROGRAM/POSITION DATA TO EPROM

The program and position data written in the drive unit memory RAM can be stored into the EPROM in the following procedure:

7.1 Inserting the Erased EPROM

Insert a new EPROM or the one erased by the EPROM eraser (recommended EPROM: Mitsubishi M5L27512K with access time of 250ns, write voltage of 12.5V) into the user socket of the drive unit. Make sure of the correct installation direction of the EPROM; the indentation must be positioned on the left.

7.2 Writing Data into EPROM

Write the contents of the drive unit RAM to the EPROM installed to the socket in the following procedure:

Using the teaching box

1) Set the teaching box ON/OFF switch to ON.

2) Press the [WRT] [9] [9] [9] [ENT] keys.

Using intelligent commands through the personal computer

1) Set the teaching box ON/OFF switch to OFF.

2) Execute the command "WR" in direct execution mode.

```
LPRINT "WR"        (Centronics)
PRINT #1, "WR"     (RS232C)
```

While the data is being written into the EPROM, the EXECUTE LED on the drive unit front control panel is lit green. The LED goes out in about 3 minutes 40 seconds after the data has been written correctly. Do not remove the EPROM from the socket until the LED goes off. Error mode II results if the data is not written correctly (due to EPROM fault or write error). At this time, reset the error and try another EPROM.

7.3 Precautions for Storage of EPROM

After the data has been written correctly to the EPROM, apply an ultraviolet rays shielding seal to the glass window of that EPROM to ensure data integrity. When the EPROM is to be stored off the socket, take necessary preventive measures against electrostatic charge.

8. OPERATION USING THE EPROM DATA (DRIVE UNIT MODE)

This section gives a procedure to operate the robot in "drive unit mode" using the program and position data stored in the EPROM. The following procedure assumes that the power is not yet switched on.

8.1 Inserting the EPROM

Insert the EPROM, storing the required program and position data, into the user socket of the drive unit. Ensure that the indentation of the EPROM is located on the left.

8.2 Setting the Side Setting Switches

Before switching on the power, set the following switches as indicated below in accordance with Section 2.1.2 Functions of side setting switches and LEDs:

ST1: Upper position (drive unit mode)
ST2: Upper position (EPROM data transferred to RAM)

8.3 Switching the Power On

Set the power switch on the drive unit rear panel to ON. The EPROM data is then transferred to the drive unit memory RAM according to the switch setting in Section 8.2. When the power is switched on with ST2 in the lower position, the program then existing in the RAM is made valid.

8.4 Executing the Program

Run the transferred program in the following procedure:

8.4.1 Step execution

The transferred program may be executed, line by line, with the teaching box keys for verification. The detailed procedure is the same as that in personal computer mode. (See Section 6.2.1 Step execution.)

8.4.2 Starting the program

In drive unit mode, the program may be initiated from the drive unit front panel:

8.4.3 Stopping and restarting the program

The program currently being run can be stopped and restarted from the drive unit front panel. The procedure is the same as that in personal computer mode. (See Section 6.2.3 Stopping and restarting the program.)

8.4.4 Stopping and resetting the program

The stopping and resetting procedures are similar to those in personal computer mode. (See Section 6.2.4 "Stopping and resetting the program.")

Stopping: Press the [STOP] switch to decelerate the robot to a stop.

Resetting: Press the [RESET] switch to reset the program.

By pressing the [START] switch after resetting, the program is executed starting at the first line. Note that the general-purpose I/O outputs are not reset.

9. OPERATION USING EXTERNAL SIGNALS

This section outlines a procedure to run the program using the external signal lines with an external programmable controller, etc. For further details, see Section 3 INTERFACE WITH EXTERNAL I/O EQUIPMENT, Vol. 5.

9.1 Setting the Switches

Set the following switches as indicated below in accordance with Section 2.1.2 Functions of side setting switches and LEDs (the teaching box ON/OFF switch should be in the OFF position):

ST1 : Upper position (drive unit mode)
SW1: Set bit 5 to upper position (external signals enabled)

By switching on the power after the above settings, the external signals are enabled for the execution of the program in drive unit mode. After these settings, the drive unit front panel switches are made invalid with the exception of the stop switches.

9.2 Executing the Program

9.2.1 Starting the program

The program is initiated by a $\overline{\text{START}}$ signal input.

9.2.2 Stopping and restarting the program

Stopping : The program is stopped by a $\overline{\text{STOP}}$ signal input. (This may also be effected by the [STOP] key on the teaching box.)

Restarting: The program is restarted by a $\overline{\text{START}}$ signal input after the $\overline{\text{STOP}}$ signal.

9.2.3 Stopping and resetting the program

Stopping : The program is stopped by a $\overline{\text{STOP}}$ signal input. (This may also be effected by the [STOP] key on the teaching box.)

Restarting: The program is reset by a $\overline{\text{RESET}}$ signal input after the $\overline{\text{STOP}}$ signal. (This may also be effected by the [RESET] key on the teaching box when the teaching box is ON.)

10. ERROR CONDITIONS

This section deals with various error conditions which may occur during the operation of the Movemaster, including the warning indicators, conditions at error occurrence, possible causes and remedial actions.

10.1 Error Mode I

Error mode I is concerned mainly with hardware errors. The cause of the error may be identified by the lit LED inside the side door of the drive unit.

〈Warning indicators〉
The error indicator LED (red) on the drive unit front panel flickers at intervals of 0.5 seconds. The buzzer, if set to on, also sounds in phase with the on timing of the LED. Further, an error signal is output from the dedicated signal line of the external I/O equipment connector.

〈Condition〉
The current to the motors for all axes (including the hand) is immediately cut off (servo off) and the brakes are applied to the J2 and J3 axes to bring the robot to an immediate stop. If the error has occurred during program execution, the teaching box LED indicates the faulty line number.

〈Possible causes〉
(1) LED1 lit: Excessive servo system errors, e.g. excessive error, overspeed, overcurrent, overload, overvoltage/open cable, hardware limit exceeded.
(2) LED2 lit: Open or disconnected signal cable between the robot and drive unit.
(3) LED3 lit: Drive unit emergency stop input, e.g. the front panel emergency stop switch pressed or the rear panel emergency stop input (N/C contact) actuated. Emergency stop is also effected when the temperature in the drive unit panel has exceeded the specified value.
(4) LED4 lit: Teaching box emergency stop input, e.g. the teaching box emergency stop switch pressed or the teaching box connector plugged or unplugged while the power is on.
(5) LED5 lit: Battery backup fault. This error does not occur when RAM initialization has been selected (i.e. bit 2 of side setting switch SW1 is in the lower position).
(6) LED6 lit: Open or disconnected signal cable between the Locomotive unit (option) and drive unit. This error is only indicated when the Locomotive unit is in use.
(7) LED7 lit: Hardware limit switch pressed by the robot operating over the allowed range after origin setting.
For the errors displayed by LEDs 1 and 7, the faulty axes may be indicated by the green LEDs (8 to 13) in the side door. (The axis numbers, 1 to 6, are indicated by the LEDs 8 to 13, respectively.) When the teaching box has been connected, more information on the error cause of each axis may be identified according to Section 3.7, Vol 4.

〈Remedial action〉
After switching off the power and eliminating the cause of the error, switch on the power and return the robot to the origin. If the cause of the error is other than an open/disconnected cable, battery fault, and hardware limit error, the error condition can be canceled by the following resetting operation without switching off the power:

Using the drive unit (with teaching box OFF)

Press the [RESET] switch on the front panel.

Using intelligent command through the personal computer (personal computer mode)

Set the teaching box ON/OFF switch to OFF and execute the "RS" command in direct mode.

LPRINT "RS" (Centronics)
PRINT #1, "RS" (RS232C)

Using the dedicated signal line of I/O connector (external mode)

Enter the $\overline{\text{RESET}}$ signal.

Using the teaching box

Set the teaching box ON/OFF switch to ON and press the [RESET] key.

The above reset operation switches on the servo and returns to the ordinary servo lock state. While the servo is off due to the error, the robot position is under proper control. Hence, the origin setting done before the error occurrence is still valid and the robot need not be returned to origin after resetting. Since the reset operation resets the running program, the program is executed from its first line number when restarted. The general-purpose I/O outputs are not reset by the reset operation.

REMARKS

If the error has occurred after origin setting, the reset operation cannot be effected when any of the robot axes has exceeded its allowed operating range. In this case, switch the power off and return the robot to origin.

10.2 Error Mode II

Error mode II is concerned mostly with software errors. No indicator is provided for error mode II to let the operator know the cause of the error.

〈Warning indicators〉
The error indicator LED (red) on the drive unit front panel is illuminated. The buzzer, if set to on, also sounds continuously. Further, an error signal is output from the dedicated signal line of the external I/O equipment connector just as in error mode I.

〈Condition〉
The system waits for the error to be reset. If the error has occurred during program execution, the program stops at the faulty line and its number is indicated on the teaching box LED.

〈Possible causes〉
(1) Command transfer error by the personal computer, e.g. an undefined command, input format error, or transmission error.

(2) Command is unexecutable, e.g. any parameter set value exceeding the specified range, move command to an undefined position.

(3) EPROM write error, e.g. EPROM unerased or EPROM fault.

(4) A move instruction has been given before origin setting.

〈Remedial action〉
Perform any of the following reset operations:

Using the drive unit (with teaching box OFF)

Press the [RESET] switch on the front panel.

Using intelligent command through the personal computer (personal computer mode)

Set the teaching box ON/OFF switch to OFF and execute the "RS" command in direct mode.

LPRINT "RS" (Centronics)
PRINT # 1, "RS" (RS232C)

Using the dedicated signal line of I/O connector (external mode)

Enter the $\overline{\text{RESET}}$ signal.

Using the teaching box

Set the teaching box ON/OFF switch to ON and press the [RESET] key.

The error indicator LED goes out after the above reset operation. Since the reset operation resets the running program, the program is executed from its first line number when restarted. The general-purpose I/O outputs are not reset by the reset operation.

1. SPECIFICATIONS
2. OPERATION
3. DESCRIPTION OF THE COMMANDS
4. MAINTENANCE AND INSPECTION
5. APPENDICES

CONTENTS (DESCRIPTION OF THE COMMANDS)

Note: Commands marked ※ may only be executed directly (cannot be programmed with a line number).

3. DESCRIPTION OF THE COMMANDS

1. COMMAND OVERVIEW

This section gives an overview of a wealth of commands provided for the Movemaster as classified according to their functions. For details of the commands, see Section 2 EXPLANATION OF THE COMMANDS.

(1) Position/motion control instructions (26 instructions)
These commands are concerned with the positions and motions of the robot. They include those defining, replacing, assigning, and calculating the position data as well as those effecting the articulated and linear interpolations and continuous-path motions. Also included are the speed-setting, origin setting and palletizing commands.

(2) Program control instructions (23 instructions)
These commands control the flow of the program. They include those concerning the subroutines, repetitive loops, and conditional jumps. Also included are the counter operation and the interrupt and logical operations by means of external signals.

(3) Hand control instructions (4 instructions)
These commands control the hand. Commands are also available for the motor-operated hand (option) that set the gripping force and open/close time of the grip.

(4) I/O control instructions (6 instructions)
These commands are concerned with the input and output of data through the general-purpose I/O ports. Both for the inputs and outputs, the data can be exchanged synchronously or asynchronously and processing is possible in bits or parallel.

(5) RS232C read instructions (8 instructions)
These commands allow the personal computer to read data from the robot's memory. The data that can be read include the position data, program data, counter data, external input data, error mode, current position, limit pulses and established tool length.

(6) Miscellaneous (4 instructions)
Grouped into this category are the error reset command, read/write commands of the user program and position data, and the command governing writing of comments.

2. EXPLANATION OF THE COMMANDS

All commands are explained in the following format in alphabetical order. Commands marked ※ can only be executed directly from a personal computer and cannot be programmed.

[Function] ..... Gives a brief description of the function invoked by the command.

[Input Format] ..... Shows the arrangement of the command entry. 〈 〉 indicates the command parameter and [ ] the parameter that can be omitted.

[Sample Input] ..... Shows a typical command entry.

[Explanation] ..... Explains the detailed function or functions to be invoked by the command and gives some precautions for use.

[Sample Program] ..... Gives a typical program with exact meaning of each line and/or some footnotes.

AN (And)

[Function]

ANDs the specified value and the internal register value.

[Input Format]

AN 〈Value to be operated〉

Where, −32768 ≦ Value to be operated (decimal) ≦ +32767
&8000 ≦ Value to be operated (hexadecimal) ≦ &7FFF

[Sample Input]

AN &800F

[Explanation]

(1) Specify the value 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 the relevant instruction. (See the EQ, NE, LG, SM, CL, DR, OR and XO commands.)

(3) Execution of the AN command after the input command (ID, IN) allows receiving of only the required bits of the parallel input data fetched from the external device.

[Sample Program]

```
10 LPRINT "100 ID"          ; Enters data from external input port.
20 LPRINT "110 AN &000F"    ; Receives four lower bits only.
30 LPRINT "120 CL 12"       ; Loads above data into counter 12.
40 LPRINT "130 EQ 8,200"    ; Jumps to line number 200 if above data is equal
                              to 8.
50 LPRINT "140 ED"          ; Ends program.
60 LPRINT "200 MO 99"       ; Moves to position 99.
```

CL (Counter Load)

[Function]

Sets the internal register value to the specified counter.

[Input Format]

CL 〈Counter number〉

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

CL 3

[Explanation]

(1) Sets the data fetched from the input port to the specified counter. Hence, the CL command must be executed after the input command (see ID, IN).

(2) Since the data from the input port is treated as signed, the data set to the counter is signed. (Between −32768 and +32767)

(3) Used to specify the number of job sequences and the counter value at palletizing from an external device, such as a programmable controller. At this time, any of the logical operation commands (see AN, OR, XO) may be used as necessary.

(4) Execution of the CL command after the CP command allows the counter data to be transferred.

(5) The counter value can be changed, compared or read by the relevant command. (See IC, DC, SC, CP and CR commands.)

[Sample Program]

```
10 LPRINT "ID"        ; Fetches data from external input port.
20 LPRINT "CL 25"     ; Sets above data to counter 25.
30 LPRINT "CP 11"     ; Sets data of counter 11 to internal register.
40 LPRINT "CL 21"     ; Sets data of internal register to counter 21.
```

CP (Compare Counter)

[Function]

Loads the value in the specified counter into the internal register.

[Input Format]

CP 〈Counter number〉

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

CP 20

[Explanation]

(1) To be executed before a conditional jump instruction (see EQ, NE, LG, SM) is executed if the value in the specified counter is used as a condition for the jump. The conditional jump instruction causes a jump to occur when a certain condition is met involving comparison with the value of the internal register loaded by the CP command.

(2) The value in the internal register remains intact if the value of the specified counter has changed after execution of the CP command. When the conditional jump is specified under the condition specifying comparison with the counter value, the CP command must be executed after the counter value has changed.

(3) The input instructions (see ID, IN) use the same internal register, meaning that the old contents of the internal register are lost when any input instruction is executed.

(4) The contents of the counter can be changed or read by the relevant instruction. (See SC, IC, DC, CR, CL, AN, OR and XO commands.)

[Sample Program]

```
10 LPRINT "100 IC 21"
20 LPRINT "110 CP 21"
30 LPRINT "120 EQ 255,500"
40 LPRINT "130 GT 100"
50 LPRINT "500 SC 21,0"
60 LPRINT "510 GT 100"
```

※ In the above example, line number 100 causes counter 21 to be incremented by 1. Line numbers 110 and 120 compare the contents of the counter with the value 255, and if they equal 255, the program jumps to line number 500, whereby the counter is initialized (reset to 0). If they do not equal 255, line number 130 causes the program to return to line number 100.

CR (Counter Read)

[Function]

Reads the contents of the specified counter (using RS232C).

[Input Format]

CR 〈Counter number〉

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

CR 75

[Explanation]

(1) Outputs the contents of the specified counter from the RS232C port.

(2) The output format is in ASCII coded decimal.

(3) The terminator of the output data is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, therefore, it is necessary to handle the entire data stream up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

(4) If an undefined counter is read, the initial value of 0 is returned. The contents of the counter are battery backed after the power is switched off.

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 INPUT "COUNTER NO. =" ; N
30 INPUT "COUNTER DATA =" ; D
40 PRINT#1, "SC" + STR$ (N) + ", " + STR$ (D)
50 PRINT#1, "CR" + STR$ (N)
60 LINE INPUT#1, A$
70 PRINT A$
80 END
   RUN ↓
   COUNTER NO. = ? 10 ↓
   COUNTER DATA = ? 255 ↓
   255
```

※ In the above example, wherein the Mitsubishi MAXY is used, line number 10 opens the RS232C communication file and line numbers 20 and 30 enter the desired counter number and data. Line number 40 uses the "SC" command to define the counter number and data, and line number 50 transmits the "CR" command and counter number. Line number 60 then uses the LINE INPUT # statement to store the received data in A$ and line number 70 outputs the data to the display screen.

DA (Disable Act)

[Function]

Disables an interrupt from being executed by a signal through the specified bit of the external input terminal.

[Input Format]

DA ⟨Bit number⟩

Where, 0 ≦ Bit number ≦ 17 (bits 16 and 17 are hand check inputs)

[Sample Input]

DA 3

[Explanation]

(1) Clears the interruptible state of the bit defined by the interrupt enable instruction (see the EA command).

(2) After the DA command has been executed, no interrupt is executed by the interrupt signal input to the specified bit during operation of the robot. Note that execution of the DA command does not affect the interruptible states of the other bits.

(3) To inhibit repeated interrupts by a single signal, either low or high, the DA command must be executed at the beginning of a line to which the program jumps after the interrupt has taken place.

[Sample Program]

See EA.

DC (Decrement Counter)

[Function]

Subtracts 1 from the value in the specified counter.

[Input Format]

DC 〈Counter number〉

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

DC 35

[Explanation]

(1) Error mode II results if the counter value drops below −32768.

(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 instruction. (See commands SC, IC, CP, CR, CL, AN, OR and XO.)

[Sample Program]

```
10 LPRINT "SC 21,15"
20 LPRINT "DC 21"
```

※ In the above example, line number 10 sets value 15 to counter 21 and line number 20 causes the counter to be decremented by 1.

DL※ (Delete Line)

[Function]

Deletes the specified line(s) in the program.

[Input Format]

DL 〈Line number (a)〉 [, 〈Line number (b)〉]

Where, 1 ≦ Line number (a), (b) ≦ 3584 Line number (a) ≦ Line number (b)

[Sample Input]

DL 200, 300

[Explanation]

(1) Deletes all lines in the program between line numbers (a) and (b). (Line number (b) included.)

(2) If line number (b) is not specified, line number (a) is only deleted.

[Sample Program]

```
10 LPRINT "100 MO 10"      ; Moves to position 10.
20 LPRINT "110 MO 12"      ; Moves to position 12.
30 LPRINT "120 MO 15"      ; Moves to position 15.
40 LPRINT "130 MO 17"      ; Moves to position 17.
50 LPRINT "140 MO 20"      ; Moves to position 20.
60 LPRINT "DL 130"         ; Deletes line 130.
```

DP (Decrement Position)

[Function]

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

[Input Format]

DP

[Sample Input]

DP

[Explanation]

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

(2) Error mode II takes place if there is no predefined position which is smaller in position number than the current position.

[Sample Program]

```
10 LPRINT "MO 3"        ; Moves to position 3.
20 LPRINT "MO 4"        ; Moves to position 4.
30 LPRINT "MO 5"        ; Moves to position 5.
40 LPRINT "DP"          ; Moves to position 4.
```

DR (Data Read)

[Function]

Reads the hand check input state and the values of the internal register.

[Input Format]

DR

[Sample Input]

DR

[Explanation]

(1) Outputs the hand check input state and the values of the internal register from the RS232C port. Allows the external input port data and hand open/close confirmation input to be read from the RS232C port when executed after the input instruction (ID, IN).

(2) The output format is in ASCII coded hexadecimal, which is headed by "&H." One byte that follows corresponds to the hand check input and the other four bytes correspond to the external input port data. (For further information, see Section 3.4 "Example of Connection to I/O Circuits" in APPENDICES.)

(3) The terminator of the output data is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, therefore, it is necessary to handle the entire data stream up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT#1, "ID"
30 PRINT#1, "DR"
40 LINE INPUT#1, A$
50 PRINT "INPUT DATA =" ;A$
60 END
   RUN ↓
   INPUT DATA = &H10FB2
```

※ In the above example, wherein the Mitsubishi MAXY is used, line number 10 opens the RS232C communication file and line number 20 loads the data of the input port into the internal register using the direct input command "ID". Line number 30 then transmits the command "DR". Line number 40 uses the LINE INPUT # statement to store the received data in A$ and line number 50 outputs the data to the display screen.

DS (Draw Straight)

[Function]

Moves the end of the hand to a position away from the current one covering the distance specified in the X-, Y- and Z-axis directions. (Linear interpolation)

[Input Format]

DS [⟨Travel distance in X⟩], [⟨Travel distance in Y⟩], [⟨Travel distance in Z⟩]

[Sample Input]

DS 50.5, −28.2, 30.0

[Explanation]

(1) The least input increment for the distance of travel is 0.1mm (e.g. specify 20.1 for 20.1mm).

(2) The attitude of the hand, including the open/close status of the gripper, remains the same before and after the movement.

(3) Error mode II occurs before or during movement if the destination or travel path exceeds the operating space of the robot. Especially the wrist roll axis must be kept from exceeding its operating space since it is compensated for during the movement.

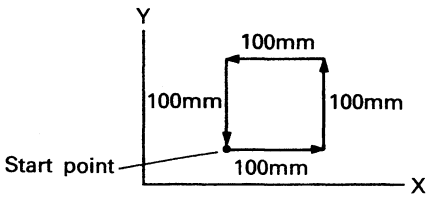
(4) The default distance of travel is 0.

(5) The moving velocity during linear interpolation is determined by the SP or SD command. (End of the hand at constant speed)

(6) The position of the end of the hand is determined by the tool length currently established. (See the TL command.)

[Sample Program]

```
10 LPRINT "DS 100,0,0"
20 LPRINT "DS 0,100,0"
30 LPRINT "DS −100,0,0"
40 LPRINT "DS 0,−100,0"
```



※ In the above example, the end of the hand moves through the four corners of a square by linear interpolation and returns the start point finally.

DW (Draw)

[Function]

Moves the end of the hand to a position away from the current one covering the distance specified in the X-, Y- and Z-axis directions. (Articulated interpolation)

[Input Format]

DW [⟨Travel distance in X⟩], [⟨Travel distance in Y⟩], [⟨Travel distance in Z⟩]

[Sample Input]

DW 10.5, 20.5, −30.5

[Explanation]

(1) The least input increment for the distance of travel is 0.1mm (e.g. specify 20.1 for 20.1mm).

(2) The attitude of the hand, including the open/close status of the gripper, remains the same before and after the movement. Error mode II occurs before start of the movement if the distance of travel exceeds the operating space of the robot.

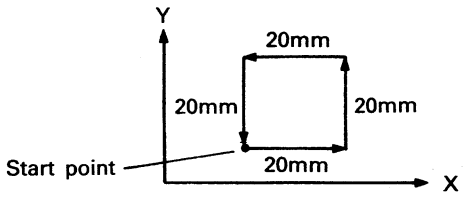
(3) The default distance of travel is 0.

(4) Since the motion is based on articulated interpolation, the combined motion of the end of the hand generates an arc when a longer distance of travel is involved.

(5) The position of the end of the hand is determined by the tool length currently established. (See the TL command.)

[Sample Program]

```
10 LPRINT "DW 20,0,0"
20 LPRINT "DW 0,20,0"
30 LPRINT "DW −20,0,0"
40 LPRINT "DW 0,−20,0"
```



※ In the above example, the end of the hand moves through the four corners of a square by articulated interpolation and returns to the start point finally.

EA (Enable Act)

[Function]

Enables an interrupt to be executed by a signal through the specified bit of the external input terminal and specifies the destination line to which the program jumps when the interrupt occurs.

[Input Format]

EA 〈+ or −〉 〈Bit number〉 , 〈Line number〉

Where, 0 ≦ Bit number ≦ 17 (bits 16 and 17 are hand check inputs.)
1 ≦ Line number ≦ 3584

[Sample Input]

EA +7, 1024

[Explanation]

(1) Causes an interrupt to be executed by an external input signal while the robot is operating. When the specified signal is input during operation after execution of the EA command, the robot is decelerated to a stop and the program jumps to the specified line. Error mode II takes place if the specified line does not exist.

(2) The jump takes place when the specified input bit is switched on (for parameter +) or off (for parameter −).

(3) Two or more bits may be specified at one time. If there are more than one inputs received, the bit with a greater bit number has precedence over the other bits.

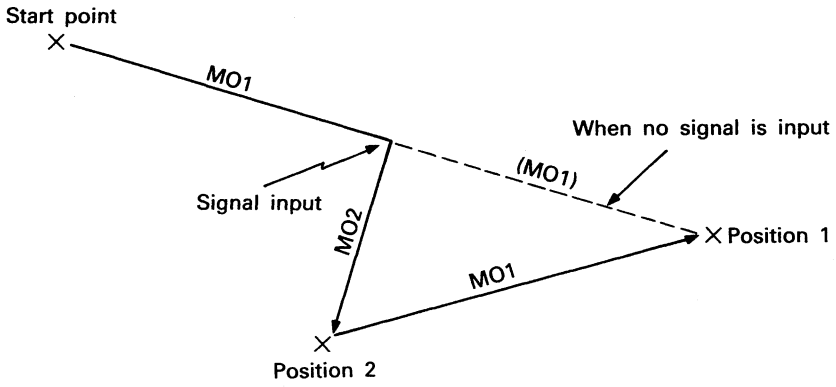
(4) Once the EA command is executed, the interruptible state is retained until the interrupt disable instruction (DA), program end instruction (ED) or reset instruction (RS) is executed.

(5) The EA command is invalid during jog operation from the teaching box and during movement effected by the move joint instruction (MJ) or nest instruction (NT).

[Sample Program]

```
10 LPRINT "100 EA +5, 600"
20 LPRINT "110 MO 1"
30 LPRINT "120 ED"
40 LPRINT "600 DA 5"
50 LPRINT "610 MO 2"
60 LPRINT "620 GT 110"
```

※ In the above example, line number 100 declares an interrupt causing the program to jump to line number 600 when bit 5 turns on and line number 110 moves the robot to position 1. When the specified signal is input during this motion, therefore, the robot is decelerated to a stop and then the program jumps to line number 600, where the interrupt is disabled. Line number 610 then moves the robot to position 2 and line number 620 causes the program to jump to line number 110. The robot is then moved to position 1 again.



ED (End)

[Function]

Ends the program.

[Input Format]

ED

[Sample Input]

ED

[Explanation]

(1) Marks the end of a program.

(2) Required at the end of a program unless the program commands are directly executed from the personal computer. (The ED command is not required, however, when the program forms a closed loop.)

[Sample Program]

```
10 LPRINT "100 SP 3"        ; Sets speed to 3.
20 LPRINT "110 MO 3"        ; Moves to position 3.
30 LPRINT "120 MO 5"        ; Moves to position 5.
40 LPRINT "130 ED"          ; Ends the program.
```

EQ (Equal)

[Function]

Causes a jump to occur if the internal register value equals the specified value when compared.

[Input Format]

EQ 〈Compared value〉 , 〈Jump destination line number〉

Where, −32768 ≦ Compared value (decimal) ≦ +32767
or &8000 ≦ Compared value (hexadecimal) ≦ &7FFF
1 ≦ Jump destination line number ≦ 3584

[Sample Input]

EQ 128, 1024 or EQ &80, 1024

[Explanation]

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

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

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

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

[Sample Program]

```
10 LPRINT "100 ID"          ; Fetches data from external input port.
20 LPRINT "110 EQ 100, 130" ; Jumps to line number 130 if input data equals
                              100.
30 LPRINT "120 ED"          ; Ends the program if above condition is not met.
40 LPRINT "130 MO 7"        ; Moves to position 7.
```

ER※ (Error Read)

[Function]

Reads the error mode status (using RS232C).

[Input Format]

ER

[Sample Input]

ER

[Explanation]

(1) Outputs the error condition of the robot from the RS232C port.

(2) Outputs 0 to indicate that no error exists, 1 to indicate error mode Ⅰ, and 2 to indicate error mode Ⅱ in ASCII.

(3) Allows the error condition to be read when the robot is operating without fault or when error mode Ⅰ or Ⅱ has occurred.

(4) The terminator of the data is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, therefore, it is necessary to handle the entire data stream up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

(5) Useful to transfer a sequence of data from the personal computer to the robot while checking for an error.

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT#1, "MO 1"
30 GOSUB 100
40 PRINT#1, "MO 2"
50 GOSUB 100
      ⟆
100 PRINT#1, "ER"
110 LINE INPUT#1, A$
120 IF A$ = "0" THEN RETURN
130 PRINT "ERROR OCCURRENCE"
140 END
```

※ In the above example, wherein the Mitsubishi MAXY is used, line number 10 opens the RS232C communication file and line number 20 and beyond transfer the commands to the robot. At this time, every time any of the commands is transferred, subroutine 100 is called to check for an error.

GC (Grip Close)

[Function]

Closes the grip of the hand.

[Input Format]

GC

[Sample Input]

GC

[Explanation]

(1) Motor-operated hand:
Causes the grip of the hand to be closed by the gripping force waveform defined by the GP command.The "retained gripping force" is only valid among the GP command parameters if the GC command is used repeatedly.

(2) Air hand:
Causes the solenoid valve to be energized to close the hand (or to attract the workpiece).

(3) A certain period of time is required before the motion of the robot becomes stabilized as its hand closes to hold the workpiece. This may make it necessary to introduce a time delay using the TI command before and after the GC command. The execution time of the GC command is determined by the parameter "starting gripping force retention time" for the GP command. (See the GP command.)

(4) The hand open/close condition depends on the setting of bit 3 of switch SW1 in the drive unit side door. (Ordinary open/close operation in the lower position. Reverse open/close operation in the upper position.)

[Sample Program]

```
10 LPRINT "100 MO 10, O"      ; Moves to position 10 (with hand open).
20 LPRINT "110 TI 5"          ; Sets 0.5sec. timer.
30 LPRINT "120 GC"            ; Closes hand (to hold workpiece).
40 LPRINT "130 TI 5"          ; Sets 0.5sec. timer.
50 LPRINT "140 MO 15, C"      ; Moves to position 15 (with hand closed).
```

GF (Grip Flag)

[Function]

Defines the open/close state of the grip of the hand (used in conjunction with the PD command).

[Input Format]

GF <Switch>

where switch = 0 or 1

[Sample Input]

GF 1

[Explanation]

(1) Defines the open or close state of the hand grip used in conjunction with the PD command which defines the coordinates of the specified position. The PD command takes precedence if the hand open/close state has been specified by the PD command.

(2) The grip is open when the switch is set to 0 and closed when the switch is 1. The switch is set to 0 (grip open) when the power is turned on. The hand open/close state depends on the setting of bit 3 of switch SW1 in the drive unit side door. (Ordinary open/close operation in the lower position. Reverse open/close operation in the upper position.)

(3) Once made, the switch setting is valid until a new value is set.

[Sample Program]

```
10 LPRINT "GF 1"                    ; Sets grip flag.
20 LPRINT "PD 10,0,550,450,-50,0" ; Defines position 10 as hand closed state.
```

GO (Grip Open)

[Function]

Opens the grip of the hand.

[Input Format]

GO

[Sample Input]

GO

[Explanation]

(1) Motor-operated hand:
Causes the grip of the hand to be opened by the gripping force waveform defined by the GP command.
The "retained gripping force" is only valid among the GP command parameters if the GO command is used repeatedly.

(2) Air hand:
Causes the solenoid valve to be energized to open the hand (or to release the workpiece).

(3) A certain period of time is required before the motion of the robot becomes stabilized as its hand opens to release the workpiece. This may make it necessary to introduce a time delay using the TI command before and after the GO command. The execution time of the GO command is determined by the parameter "starting gripping force retention time" for the GP command. (See the GP command.)

(4) The hand open/close condition depends on the setting of bit 3 of switch SW1 in the drive unit side door. (Ordinary open/close operation in the lower position. Reverse open/close operation in the upper position.)

[Sample Program]

```
10 LPRINT "100 MO 10, O"      ; Moves to position 10 (with hand closed).
20 LPRINT "110 TI 5"          ; Sets 0.5sec. timer.
30 LPRINT "120 GO"            ; Opens hand (to release workpiece).
40 LPRINT "130 TI 5"          ; Sets 0.5sec. timer.
50 LPRINT "140 MO 15, O"      ; Moves to position 15 (with hand open).
```

GP (Grip Pressure)

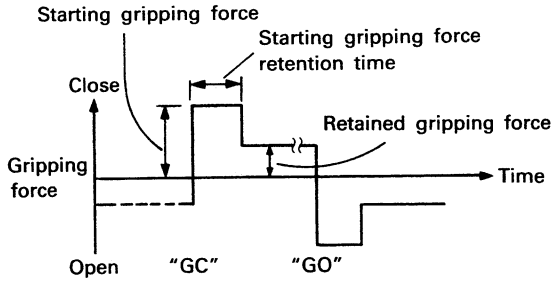
[Function]

Defines the gripping force to be applied when the motor-operated hand is closed and opened.

[Input Format]

GP ⟨Starting gripping force⟩ , ⟨Retained gripping force⟩ , ⟨Starting gripping force retention time⟩

where 0 ≦ Starting/retained gripping force ≦ 63 (invalid)
0 ≦ Starting gripping force retention time ≦ 99



[Sample Input]

GP 32, 32, 5

[Explanation]

(1) Sets the gripping force of the motor-operated hand (option) as it changes with time. (See the GO and GC commands.)

(2) The starting and retained gripping forces are 63 at their maximum and 0 at their minimum. The starting gripping force retention time is the parameter × 0.1 seconds (max. 9.9 seconds). Define the parameters optimum for the workpiece to be held. The parameter setting, once made, remains valid until a new value is set.

(3) The default parameter values are "GP 63, 63, 3".

(4) When the air hand is used, the parameters, starting and retained gripping forces, are invalid.

(5) The robot motion stops during the starting gripping force retention time.

[Sample Program]

```
10 LPRINT "GP 10, 6, 10"        ; Sets grip pressure.
20 LPRINT "GC"                  ; Closes hand in above settings.
```

GS (Go Sub)

[Function]

Permits the instruction sequence to jump to the subroutine which starts with the specified line number.

[Input Format]

GS 〈 Line number〉

where 1 ≦ Line number ≦ 3584

[Sample Input]

GS 1024

[Explanation]

(1) Permits the instruction sequence to jump to the specified line number. The RT command is used to return subsequently to the main program after the subroutine has been completed. If the specified line does not exist, error mode II occurs at the time of GS execution.

(2) Subroutines are written and stored separately from the main program and must be terminated by the RT command.

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

[Sample Program]

```
10 LPRINT "20 GS 100"          ; Jumps to subroutine beginning at line number
       ⎩                         100.

200 LPRINT "90 ED"             ; Ends program.
210 LPRINT "100 MO 11"         ; Moves to position 11.
220 LPRINT "110 MO 12"         ; Moves to position 12.
230 LPRINT "120 MO 13"         ; Moves to position 13.
240 LPRINT "130 RT"            ; Ends subroutine.
```

Subroutine

GT (Go To)

[Function]

Permits the program sequence to jump to the specified line number unconditionally.

[Input Format]

GT ⟨Line number⟩

where ≦ 1 Line number ≦ 3584

[Sample Input]

GT 1024

[Explanation]

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

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

[Sample Program]

```
10 LPRINT "20 MO 1"          ; Moves to position 1.
20 LPRINT "30 GT 100"        ; Jumps to line number 100 unconditionally.
      ⎩
200 LPRINT "100 MO 12"       ; Moves to position 12.
210 LPRINT "110 MO 15"       ; Moves to position 15.
```

HE (Here)

[Function]

Defines the coordinates of the current position by assigning a position number to it.

[Input Format]

HE ⟨Position number⟩

where ≦ 1 Position number ≦ 999

[Sample Input]

HE 5

[Explanation]

(1) The coordinates of the current position are calculated on the basis of the currently established tool length (see the TL command). In the initial condition, the tool length is 123mm, a point 123mm away from the center of 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 is also stored as the position data.

(4) Error mode II may occur if the HE command is executed when any of the axes of the robot is positioned near the boundary of its operational space. In such cases, avoid such attitude of the robot.

(5) Error mode II takes place if the HE command is executed before the robot returns to the origin.

[Sample Program]

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

HO (Home)

[Function]

Establishes the reference position for origin setting.

[Input Format]

HO

[Sample Input]

HO

[Explanation]

(1) Establishes the reference position for origin setting. Execute the HO command after matching the match marks on each axis of the robot using the teaching box. (See Section 4.3 "Origin Attitude Setting", Vol. 2. OPERATION.)

(2) The HO command must be executed again when the drive unit used with the robot has been changed, any robot part has been replaced, or the robot has been adjusted.

(3) The HO command cannot be executed when bit 4 of switch SW1 in the drive unit side door is in the lower (OFF) position.

[Sample Program]

```
10 LPRINT "HO"
```

IC (Increment Counter)

[Function]

Adds 1 to the value in the specified counter.

[Input Format]

IC <Counter number>

where ≦ 1 Counter number ≦ 99

[Sample Input]

IC 25

[Explanation]

(1) Error mode II occurs if the counter value exceeds 32767.

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

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

[Sample Program]

```
10 LPRINT "SC 21,15"
20 LPRINT "IC 21"
```

※ In the above example, line number 10 sets value 15 in counter 21 and line number 20 causes the counter to be incremented by 1.

ID (Input Direct)

[Function]

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

[Input Format]

ID

[Sample Input]

ID

[Explanation]

(1) Causes a signal (parallel data) from the external device, e.g. a programmable controller, to be fetched unconditionally from the input port, and at the same time, data to be fetched from the hand check input.

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

(3) For information on connections, see Section 3 "INTERFACE WITH EXTERNAL I/O EQUIPMENT," Vol. 5 APPENDICES.

[Sample Program]

```
10 LPRINT "100 ID"          : Fetches data from external input port.
20 LPRINT "110 EQ 100,130"  ; Jumps to line number 130 if input data equals
                              100.
30 LPRINT "120 ED"          ; Ends program if above condition is not met.
40 LPRINT "130 MO 7"        ; Moves to position 7.
```

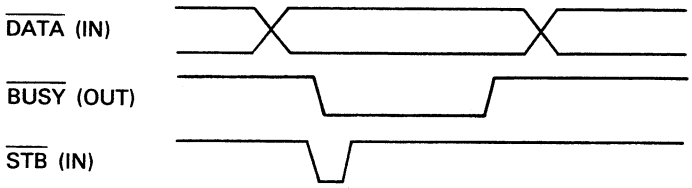
IN (Input)

[Function]

Fetches data synchronously from the external input port and hand check input (using the control signal lines).

[Input Format]

IN



[Sample Input]

IN

[Explanation]

(1) Causes a signal (parallel data) from the external device, e.g. a programmable controller, to be fetched synchronously from the input port, and simultaneously, data to be fetched from the hand check input. At this time, the control signals ($\overline{STB}$ and $\overline{BUSY}$ signals) must have been connected to the external device.

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

(3) For information on connections, see Section 3 "INTERFACE WITH EXTERNAL I/O EQUIPMENT," Vol. 5 APPENDICES.

[Sample Program]

```
10 LPRINT "100 IN"          : Fetches data from external input port.
20 LPRINT "110 EQ 100,130"  ; Jumps to line number 130 if input data equals
                              100.
30 LPRINT "120 ED"          ; Ends program if above condition is not met.
40 LPRINT "130 MO 7"        ; Moves to position 7.
```

IP (Increment Position)

[Function]

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

[Input Format]

IP

[Sample Input]

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) Error mode II takes place if there is no predefined position which is greater in position number than the current position.

[Sample Program]

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

LG (If Larger)

[Function]

Causes a jump to occur if the internal register value is greater than the specified value when compared.

[Input Format]

LG ⟨Compared value⟩ , ⟨Jump destination line number⟩

where $-32768 \leqq$ Compared value (decimal) $\leqq +32767$
or &8000 $\leqq$ Compared value (hexadecimal) $\leqq$ &7FFF
$1 \leqq$ Jump destination line number $\leqq 3584$

[Sample Input]

LG 128, 1024 or LG &80, 1024

[Explanation]

(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 greater than the compared value (i.e. when the condition is met), the program jumps to the specified line number. Otherwise (i.e. when the condition is not met), the program continues in sequence. Error mode II occurs at a jump time if the specified line number does not exist.

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

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

[Sample Program]

```
10 LPRINT "100 ID"              ; Fetches data from external input port.
20 LPRINT "110 LG 100, 130"     ; Jumps to line number 130 if input data is greater
                                  than 100.
30 LPRINT "120 ED"              ; Ends the program if above condition is not met.
40 LPRINT "130 MO 7"            ; Moves to position 7.
```

LR※ (Line Read)

[Function]

Reads the program on the specified line number (using RS232C).

[Input Format]

LR ⟨Line number⟩

where ≦1 Line number ≦ 3584

[Sample Input]

LR 512

[Explanation]

(1) Outputs the program on the specified line number from the RS232C port.

(2) The data is output in ASCII.

(3) The terminator of the output data is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, therefore, it is necessary to handle the entire data stream up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

(4) When an undefined line number has been read, only 0DH is returned.

(5) The parameter entered in hexadecimal using "&" is output in decimal equivalent (e.g. "OD & FF" is converted into "OD 255").

[Sample Program]

```
10 OPEN "COM1:9600, E, 7, 2" AS #
1
20 INPUT "START LINE =" ;S
30 INPUT "END LINE =" ;E
40 FOR I=S TO E
50 PRINT #1, "LR" STR$(I)
60 LINE INPUT #1, A$
70 IF A$="" THEN 90
80 PRINT I;:PRINT A$
90 NEXT
100 END
    RUN ↓
    START LINE =?1 ↓
    END LINE =?8 ↓
```

```
1 NT
2 MO 1
4 DW +10.0, +20.0, -30.0
6 MO 2
8 ED
```

※ In the above example, wherein the Mitsubishi MAXY is used, line number 10 opens the RS232C communication file and line numbers 20 and 30 enter the starting and ending line numbers for a complete listing of programs between the two lines. Line number 50 transmits the command "LR" and the line numbers represented by STR$(I). Line number 60 then uses the LINE INPUT # statement to store the received data in A$ and line number 70 verifies the presence of the data. If there is data, line number 80 causes its contents to be shown on the display screen together with the corresponding line numbers. If there is no data, the program jumps to line number 90 and reexecutes line numbers 40 and onward.

MA (Move Approach)

[Function]

Moves the end of the hand from the current position to a position away from the specified position in increments as specified for another position. (Articulated interpolation)

[Input Format]

MA ⟨Position number (a)⟩ , ⟨Position number (b)⟩ [, ⟨O or C⟩]

where ≦ 1 Position number (a), (b) ≦ 999

[Sample Input]

MA, 2, 3, C

[Explanation]

(1) Moves the end of the hand from the current position to a position away from position (a) in increments as specified for position (b). It does not change the coordinates of positions (a) and (b). (See the SF command.)
(Each coordinate of position (b) is temporarily added to the corresponding coordinate of position (a).)

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

(3) If the calculated incremental dimensions exceed the robot's operational space, error mode II occurs before the robot moves.

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

(5) The position of the end of the hand is determined by the tool length currently established. (See the TL command.)

[Sample Program]

```
10 LPRINT "HE 1"
20 LPRINT "PD 5, 0, 0, 30, 0, 0"
30 LPRINT "MA 1, 5, O"
```

※ In the above example, the end of the hand is moved with the hand opened from position 1 to a position in 30 mm increments only in the Z-axis direction. The values of coordinates of positions 1 and 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 or C⟩]

where 1 ≦ Position number (a), (b) ≦ 999
| Position number (a) − Position number (b) | ≦ 99

[Sample Input]

MC, 101, 200

[Explanation]

(1) Moves the robot at a given speed from the current position via position (a) to position (b) by linear interpolation, moving continuously past the predefined intermediate points between positions (a) and (b).

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

(3) When the hand open/close setting (O for open, C for close) has been done, hand open/close control is executed before the movement is effected. When the hand open/close setting has not been made, the open/close state of the hand before the robot starts moving remains valid throughout the entire movement.

(4) Since the robot does not accelerate or decelerate during its moving through the intermediate points, avoid movement path that involves a great change in orientation of any of the axes of motion; otherwise, error mode I may occur.

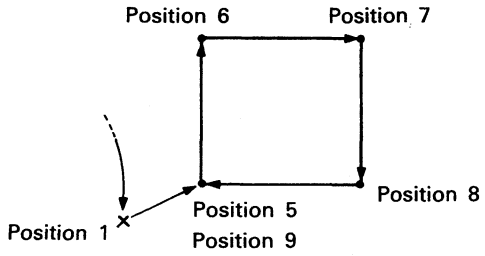
(5) The speed of travel during linear interpolation is determined by the SP or SD command. (End of the hand at constant speed)

(6) Error mode II takes place if specified positions (a) and (b) have not been defined or defy the physical movement of the robot, or if the difference between the position numbers (a) and (b) exceeds 99.

(7) Error mode II also occurs during movement if the movement path goes beyond the robot's operational space.

[Sample Program]

```
10 LPRINT "SP 10"
20 LPRINT "MO 1"
30 LPRINT "MC 5, 9"
```



※ In the above example, line number 10 sets speed to 10 and line number 20 moves the robot to position 1 by articulated interpolation. Line number 30 then moves the robot continuously from position 5 to position 9 by linear interpolation (at constant speed). (Without acceleration/deceleration midway through the operation)

MJ※ (Move Joint)

[Function]

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

[Input Format]

MJ[<Waist turning angle>], [<Shoulder turning angle>], [<Elbow turning angle>] , [<Pitch angle, Roll angle>]

[Sample Input]

MJ 10, 20, −30, 40, −50

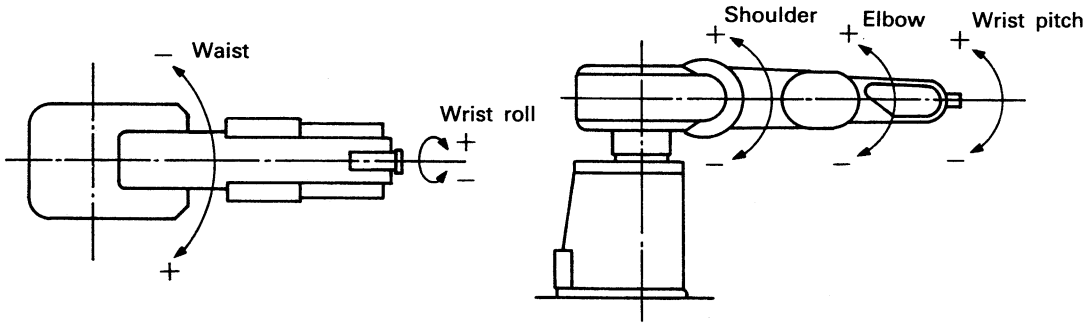
[Explanation]

(1) The least input increment of the turning angle is 0.1°, e.g. specify 15.2 for 15.2°.

(2) The open/close state of the hand does not change before and after the movement. Error mode II occurs before the axis 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 axis of motion are as follows:



[Sample Program]

```
10 LPRINT "MJ +90, 0, 0, 0, 0"
20 LPRINT "MJ 0, −30, 0, 0, 0"
30 LPRINT "MJ 0, 0, 0, +20, 0"
```

※ In the above example, line number sweeps the waist 90° in the positive direction, line number 20 swivels the shoulder 30° in the negative direction, and line number 30 bends the wrist 20° in the positive direction.

MO (Move)

[Function]

Moves the end of the hand to the specified position. (Articulated interpolation)

[Input Format]

MO <Position number> [, <O or C>]

where ≦1 Position number ≦ 999

[Sample Input]

MO, 2, C

[Explanation]

(1) Moves the end of the hand to the coordinates of the specified position by articulated interpolation. The position of the end of the hand is determined by the tool length currently established. (See the TL command.)

(2) If the open/close state of the hand has been specified (O: open; C: closed), the end of the hand moves after executing the hand control instruction. If it has not been specified, the definition of the specified position is executed.

(3) Error mode II takes place if the specified position has not been predefined or the movement exceeds the robot's operational space.

[Sample Program]

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

MP※ (Move Position)

[Function]

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

[Input Format]

MP [<X-axis coordinate>] , [<Y-axis coordinate>] , [<Z-axis coordinate>] , [<Pitch angle>] , [<Roll angle>]

[Sample Input]

MP 0, 380, 300, −70, −40

[Explanation]

(1) The least input increment of the coordinate values is 0.1mm or 0.1° (e.g. specify 20.1 for 20.1mm).

(2) Error mode II occurs if the specified coordinates exceed the robot's operational space.

(3) Any coordinate value defaults to 0.

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

(5) The position of the end of the hand depends on the current tool length. (See the TL command.)

[Sample Program]

```
10 LPRINT "PD 1, 0, 550, 450, −50,
0"
20 LPRINT "MO 1"
30 LPRINT "MP 0, 550, 430, −50, 0"
```

※ In the above example, the end of the hand first moves to position 1, then goes 20mm down in the Z-axis direction according to line number 30, with the open/close state of the hand remaining unchanged.

MS (Move Straight)

[Function]

Moves the end of the hand to the specified position number through the specified number of intermediate points on a straight line. (Linear interpolation)

[Input Format]

MS ⟨Position number⟩ [, ⟨O or C⟩]

where ≦ 1 Position number ≦ 999

[Sample Input]

MS, 2, C

[Explanation]

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

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

(3) If the open/close state of the hand has been specified (O: open; C: closed), the end of the hand moves after executing the hand control instruction. If it has not been specified, the definition of the specified position is executed.

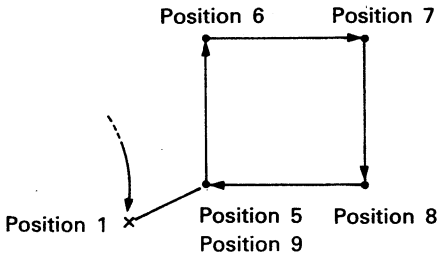
(4) Error mode II takes place if the specified position has not been predefined.

(5) The speed of travel during linear interpolation is determined by the SP or SD command. (End of the hand at constant speed) Error mode II will occur if the maximum speed of any axis is exceeded depending on the attitude.

(6) Use the MC command to move the end of the hand continuously through several positions by linear interpolation without acceleration/deceleration.

[Sample Program]

```
10 LPRINT "SP 10"
20 LPRINT "MO 1"
30 LPRINT "MS 5"
40 LPRINT "MS 6"
50 LPRINT "MS 7"
60 LPRINT "MS 8"
70 LPRINT "MS 9"
```



※ In the above example, line number 10 sets speed to 10 and line number 20 moves the end of the hand to position 1 by articulated interpolation. Line numbers 30 through 70 then move the end of the hand sequentially from position 5 to position 9 by linear interpolation (at constant speed). (Acceleration/deceleration made midway through the operation)

MT (Move Tool)

[Function]

Moves the end of the hand from the current position to a position away from the specified position in increments as specified in the tool direction. (Articulated interpolation)

[Input Format]

MT <Position number> , [<Travel distance>] [, <O or C>]

where ≦1 Position number ≦ 999

[Sample Input]

MT, 5, +70, O

[Explanation]

(1) The least input increment of the distance of travel is 0.1mm (e.g. specify 150.5 for 150.5mm).

(2) When the distance of travel entered is positive, the end of the hand advances along the specified distance in the tool direction. When the distance of travel is negative, the end of the hand retracts along the specified distance in the tool direction. (The default distance is 0.)

(3) If the open/close state of the hand has been specified (O: open; C: closed), the end of the hand moves after executing the hand control instruction. If it has not been specified, the definition of the specified position is executed.

(4) Error mode II takes place before the end of the hand starts moving if the destination exceeds the robot's operational space. It also occurs if the specified position has not been predefined or is out of the operational space.

(5) The position of the end of the hand is determined by the tool length currently established. (See the TL command.)

[Sample Program]

```
10 LPRINT "HE 1"
20 LPRINT "MT 1, +30, C"
```

※ In the above example, line number 20 defines the reference position as the current position. As a result, the end of the hand advances 30mm from the current position in the tool direction with the hand closed.

NE (If Not Equal)

[Function]

Causes a jump to occur if the internal register value does not equal the specified value when compared.

[Input Format]

NE ⟨Compared value⟩ , ⟨Jump destination line number⟩

where −32768 ≦ Compared value (decimal) ≦ +32767
or &8000 ≦ Compared value (hexadecimal) ≦ &7FFF
1 ≦ Jump destination line number ≦ 3584

[Sample Input]

NE 128, 1024 or NE &80, 1024

[Explanation]

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

(2) If the internal comparison register value does not equal the compared value (i.e. when the condition is met), the program jumps to the specified line number. Otherwise (i.e. when the condition is not met), the program continues in sequence. Error mode II occurs at a jump time if the specified line number does not exist.

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

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

[Sample Program]

```
10 LPRINT "100 ID"          ; Fetches data from external input port.
20 LPRINT "110 NE 100, 130" ; Jumps to line number 130 if input data does not
                              equal 100.
30 LPRINT "120 ED"          ; Ends the program if above condition is not met.
40 LPRINT "130 MO 7"        ; Moves to position 7.
```

NR (Nest Read)

[Function]

Reads the number of limit pulses of any axis after origin setting (using RS232C).

[Input Format]

NR <Axis number>

where ≦1 axis number ≦ 5

[Sample Input]

NR 5

[Explanation]

(1) Outputs the number of limit pulses of the specified axis from the RS232C port after origin setting. The limit pulses are required to check the limit switch installation state of each axis during maintenance and inspection.

(2) The output format is as indicated below and the data is all output in ASCII code.
Output format: Number of limit pulses for the specified axis 0D (hexadecimal)

(3) The limit pulse checking operation may also be performed from the teaching box. For more information, see Section 3.8 in MAINTENANCE AND INSPECTION.

(4) The number of limit pulses indicated is the result of the most recent origin setting. Therefore, the number of limit pulses read by effecting the origin setting again may be slightly different from the previous value.

(5) Error mode II occurs if the axis number exceeds the specified range.

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS #1
20 INPUT "AXIS NO. =" ;J
30 PRINT #1, "NR" +STR$(J)
40 LINE INPUT# 1, A$
50 PRINT A$
60 END
RUN ↓
250
```

※ In the above example, wherein the Mitsubishi MAXY is used, line number 10 opens the RS232C communication file and line number 20 enters the axis number to be read. Line number 30 then transmits command "NR" and the above axis number. Line number 40 uses the LINE INPUT# statement to store the received data in A$ and line number 50 outputs the data to the display screen.

NT (Nest)

[Function]

Returns the robot to mechanical origin.

[Input Format]

NT

[Sample Input]

NT

[Explanation]

(1) Used to return the robot to mechanical origin immediately after the power is switched on. Must be executed before teaching or the program is executed.

(2) The function of the NT command depends on the position of bit 4 of switch SW1 in the drive unit side door. Usually set bit 4 to the lower position. (See Section 4.3 "Origin Attitude Setting", Vol. 2, OPERATION.)

(3) Origin setting of the J2, J3 and J4 axes is first executed, which is followed by origin setting of the J1 and J5 axes. If the arm can interfere with the object surrounding the robot, use the teaching box to move it to a safe location before attempting to return the robot to origin.

(4) When the hand holds a workpiece, care must be taken to prevent personal injury because the hand opens as soon as the origin setting operation is initiated. The hand open/close condition depends on the setting of bit 3 of switch SW1 in the drive unit side door. (Set to the lower position for ordinary opening/closing and to the upper position for reverse opening/closing.)

(5) Ordinarily, origin setting may be executed only once after the power is switched on. If, after origin setting, the robot has been stopped by emergency stop operation during re-execution, it returns to the power-on state and the origin setting operation must therefore be performed again.

[Sample Program]

```
10 LPRINT "NT"          ; Executes origin setting.
20 LPRINT "MO 1"        ; Moves to position 1.
```

NW※ (New)

[Function]

Deletes all program and position data.

[Input Format]

NW

[Sample Input]

NW

[Explanation]

(1) Deletes all programs and positions on the drive unit RAM.

(2) Does not delete the reference position data for origin setting.

(3) Use the DL or PC command to delete programs and positions individually.

(4) When the NW command is executed, the counter values, internal register value, established tool length, set speed, hand grip pressure value and pallet set data remain intact (unchanged).

[Sample Program]

```
10 LPRINT "NW"          ; Deletes all programs and positions.
```

NX (Next)

[Function]

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

[Input Format]

NX

[Sample Input]

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) Error mode II 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 ⟨+ or −⟩ ⟨Bit number⟩

where ≦ 0 Bit number ≦ 15

[Sample Input]

OB +1

[Explanation]

(1) Set + to switch on the specified bit and − to switch off the specified bit. Append the + or − sign in front of the bit number.

(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, OD or OT command.

(3) If no bit number is specified, it defaults to bit 0.

[Sample Program]

```
10 LPRINT "OD&FFFF"          ; Outputs data in hexadecimal (&FFFF).
20 LPRINT "OB -0"
```

※ In the above example, line number 10 sets all external bits (bits 0 to 15) to ON and line number 20 only sets bit 0 to OFF.

OD (Output Direct)

[Function]

Outputs the specified data unconditionally through the output port.

[Input Format]

OD ⟨Output data⟩

where $-32768 \leqq$ Output data (decimal) $\leqq +32767$
$\&8000 \leqq$ Output data (hexadecimal) $\leqq \&7FFF$

[Sample Input]

OD 7

[Explanation]

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

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

(3) For information on connections, see Section 3 "INTERFACE WITH EXTERNAL I/O EQUIPMENT", Vol. 5. APPENDICES.

[Sample Program]

```
10 LPRINT "OD &FFFF"          ; Sets all external output port bits (16 bits) to ON.
```

OG (Origin)

[Function]

Moves the robot to the mechanical reference position for origin setting. (Articulated interpolation)

[Input Format]

OG

[Sample Input]

OG

[Explanation]

(1) Moves the robot to its mechanical origin which has been defined by the NT command or the NST ENT operation from teaching box.

(2) After the OG command has been executed, the attitude of the robot is the same as that after origin setting operation, which means that the match marks of each robot axis match with each other.

(3) Error mode II occurs if the OG command is executed before origin setting.

[Sample Program]

```
10 LPRINT "NT"          ; Executes origin setting.
20 LPRINT "MO 2"        ; Moves to position 2.
30 LPRINT "OG"          ; Returns to origin setting position.
```

OR (Or)

[Function]

ORs the specified data and the internal register data.

[Input Format]

OR <Data to be operated>

where −32768 ≦ Data to be operated (decimal) ≦ +32767
&8000 ≦ Data to be operated (hexadecimal) ≦ &7FFF

[Sample Input]

OR &8001

[Explanation]

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

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

(3) Execution of the OR command after the input command (see ID, IN) allows 1 to be set to the required bits of the parallel input data fetched from the external device.

[Sample Program]

```
10 LPRINT "100 ID"              ; Enters data from external input port.
20 LPRINT "110 OR &FFF0"        ; Sets 1 to all bits other than four lower bits.
30 LPRINT "120 EQ &FFFF, 200"   ; Jumps to line number 200 if above data is 1 in all
                                  bits.
40 LPRINT "130 ED"              ; Ends program.
50 LPRINT "200 MO 10"           ; Moves to position 10.
```

OT (Output)

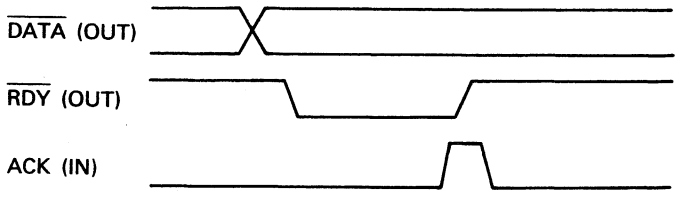
[Function]

Outputs the specified data synchronously through the output port (using the control signal lines).

[Input Format]

OT <Output data>

where $-32768 \leqq$ Output data (decimal) $\leqq +32767$
$\&8000 \leqq$ Output data (hexadecimal) $\leqq \&7FFF$



[Sample Input]

OT 7

[Explanation]

(1) Outputs a signal (parallel data) synchronously through the output port to external equipment such as a programmable controller. At this time, the control signal lines (RDY and ACK signals) must have been connected to the external equipment. The data output to the external equipment is retained.

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

(3) For information on connections, see Section 3 "INTERFACE WITH EXTERNAL I/O EQUIPMENT", Vol 5. APPENDICES.

[Sample Program]

```
10 LPRINT "OT &FFFF"          ; Sets all external output port bits (16 bits) to ON.
```

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⟩

where $1 \leqq$ Pallet number $\leqq 9$
$1 \leqq$ Number of column grid points $\leqq 32767$
$1 \leqq$ Number of row grid points $\leqq 32767$

[Sample Input]

PA 3, 20, 30

[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 determined by the directions of the terminating positions, respectively. (See the PT command.)

[Sample Program]

```
10 LPRINT "PA 5, 20, 30"
20 LPRINT "SC 51, 15"
30 LPRINT "SC 52, 25"
40 LPRINT "PT 5"
50 LPRINT "MO 5"
```

※ In the above example, line number 10 defines pallet 5 as the pallet having 20 × 30 grid points. Line numbers 20, 30 and 40 then identify the coordinates of one of the grid points (15, 25) as position 5 and line number 50 moves the robot to that position.

PC※ (Position Clear)

[Function]

Clears the data of the specified position(s).

[Input Format]

PC 〈Position number (a)〉 [, 〈Position number (b)〉]

where 1 ≦ Position number (a), (b) ≦ 999, Position number (a) ≦ Position number (b)

[Sample Input]

PC 5, 8

[Explanation]

(1) Deletes all position data between positions (a) and (b). (Position (b) included)

(2) Only clears the data of position (a) if position (b) is not specified.

(3) Note that the reference position data for origin setting is cleared if any position number is not specified or 0 (zero) is defined, with bit 4 of switch SW1 in the drive unit side door located in the upper position. At this time, the robot must be set to the origin setting reference position once again by the HO command or by the [P. S] [0] operation from the teaching box. (See Section 4.3 "Origin Attitude Setting," Vol. 2 OPERATION.)

[Sample Program]

```
10 LPRINT "MO 10"        ; Moves to position 10.
20 LPRINT "MO 11"        ; Moves to position 11.
30 LPRINT "MO 12"        ; Moves to position 12.
40 LPRINT "PC 11"        ; Deletes position 11.
50 LPRINT "DP"           ; Moves to position 10.
```

PD※ (Position Define)

[Function]

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

[Input Format]

PD 〈Position number〉 , [〈X-axis coordinate〉] , [〈Y-axis coordinate〉] , [〈Z-axis coordinate〉] , [〈Pitch angle〉] , [〈Roll angle〉] [, 〈O or C〉]

where, 1 ≦ Position number ≦ 999

[Sample Input]

PD 10, 0, 380, 300, −70, −40, C

[Explanation]

(1) The least input increment of the coordinate values is 0.1mm or 0.1° (e.g. specify 20.1 for 20.1mm).

(2) No error occurs if the defined coordinates exceed the robot's operational space. This allows you to define a position representing an incremental movement when the command is used in combination with the other command, e.g. SF, MA.

(3) Any coordinate value defaults to 0.

(4) When the open/close state of the hand (O: Open; C: Close) is not specified, it is defined by the grip flag command. (See command GF.)

(5) The PD command can only define a robot position when the coordinates of the end of the hand as determined by the tool command are located ahead of the Z-axis (i.e. the direction the robot faces).

[Sample Program]

```
100 LPRINT "PD 10, 0, 550, 430, −50, −40, C"
110 LPRINT "PD 20, 0, 0, 20, 0, 0"
120 LPRINT "SF 10, 20"
130 LPRINT "MO 10"
```

※ In the above example, line number 100 defines position 10, at which the hand is closed, while line number 110 defines position 20. Line number 120 then redefines position 10 as a position being shifted 20mm in the Z axis as determined by position 20. Line number 130 moves the robot to the new position 10.

PL (Position Load)

[Function]

Assigns the coordinates of the specified position to another specified position.

[Input Format]

PL 〈Position number (a)〉 , 〈Position number (b)〉

where 1 ≦ Position number (a), (b) ≦ 999

[Sample Input]

PL 5, 7

[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 position of the hand at position (b) to that at position (a).

(3) Error mode II occurs if position (b) is not defined.

[Sample Program]

```
10 LPRINT "HE 2"
20 LPRINT "PL 3, 2"
```

※ In the above example, the current coordinates and hand open/close state are defined as position 2 and the data of position 2 is then copied to position 3.

PR (Position Read)

[Function]

Reads the coordinates of the specified position and the open/close state of the hand (using RS232C).

[Input Format]

PR 〈Position number〉

where 1≦ Position number ≦ 999

[Sample Input]

PR 5

[Explanation]

(1) Causes the coordinates of the specified position and the open/close state of the hand to be output from the RS232C port.

(2) The data is output in ASCII as indicated below. The least output increment is 0.1mm or 0.1° (e.g. 20.1 is displayed for 20.1mm).
Output format: X-axis coordinate, Y-axis coordinate, Z-axis coordinate, pitch angle, roll angle, O or C 0D (Hex)
In the above format, the hand open state is output as O and the closed state as C.

(3) The delimiter of the data is a comma (,: Hex. 2C) and the terminator is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, it is necessary to handle the entire data string up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

(4) When an undefined position is read, 0's are returned (0, 0, 0, 0, 0).

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 INPUT "POSITION NO. =" ;P
30 PRINT#1, "PR" + STR$(P)
40 LINE INPUT#1, A$
50 PRINT A$
60 END
   RUN↓
   POSITION NO. =? 15↓
   +10.0, +380.0, +300.0, −70.0, −40.0, C
```

※ In the above example, which uses Mitsubishi's MAXY, line number 10 opens the RS232C communication file and line number 20 enters the required position number. Line number 30 transmits the command "PR" and the entered position number. Line number 40 then uses the LINE INPUT # statement to store the received data into A$ and line number 50 outputs the data to the display screen.

PT (Pallet)

[Function]

Calculates the coordinates of a grid point on the specified pallet and identifies the coordinates as the position number corresponding to the specified pallet number.

[Input Format]

PT 〈Pallet number〉

where 1 ≦ Pallet number ≦ 9

[Sample Input]

PT 3

[Explanation]

(1) Calculates the coordinates of a grid point on the specified pallet and identifies the coordinates as the position number 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. The following is a listing of a combination of pallet positions and counters corresponding to each pallet number:

(Pallet 1)	Pallet reference position	Position 10
	Pallet column terminating position	Position 11
	Pallet row terminating position	Position 12
	Pallet corner position opposite to reference	Position 13
	Pallet column counter	Counter 11
	Pallet row counter	Counter 12
	PT1 position	Position 1
(Pallet 2)	Pallet reference position	Position 20
	Pallet column terminating position	Position 21
	Pallet row terminating position	Position 22
	Pallet corner position opposite to reference	Position 23
	Pallet column counter	Counter 21
	Pallet row counter	Counter 22
	PT2 position	Position 2

(Pallet 3)	Pallet reference position	Position 30
	Pallet column terminating position	Position 31
	Pallet row terminating position	Position 32
	Pallet corner position opposite to reference	Position 33
	Pallet column counter	Counter 31
	Pallet row counter	Counter 32
	PT3 position	Position 3
(Pallet 4)	Pallet reference position	Position 40
	Pallet column terminating position	Position 41
	Pallet row terminating position	Position 42
	Pallet corner position opposite to reference	Position 43
	Pallet column counter	Counter 41
	Pallet row counter	Counter 42
	PT4 position	Position 4
(Pallet 5)	Pallet reference position	Position 50
	Pallet column terminating position	Position 51
	Pallet row terminating position	Position 52
	Pallet corner position opposite to reference	Position 53
	Pallet column counter	Counter 51
	Pallet row counter	Counter 52
	PT5 position	Position 5
(Pallet 6)	Pallet reference position	Position 60
	Pallet column terminating position	Position 61
	Pallet row terminating position	Position 62
	Pallet corner position opposite to reference	Position 63
	Pallet column counter	Counter 61
	Pallet row counter	Counter 62
	PT6 position	Position 6
(Pallet 7)	Pallet reference position	Position 70
	Pallet column terminating position	Position 71
	Pallet row terminating position	Position 72
	Pallet corner position opposite to reference	Position 73
	Pallet column counter	Counter 71
	Pallet row counter	Counter 72
	PT7 position	Position 7
(Pallet 8)	Pallet reference position	Position 80
	Pallet column terminating position	Position 81
	Pallet row terminating position	Position 82
	Pallet corner position opposite to reference	Position 83
	Pallet column counter	Counter 81
	Pallet row counter	Counter 82
	PT8 position	Position 8

(Pallet 9)	Pallet reference position	Position 90
	Pallet column terminating position	Position 91
	Pallet row terminating position	Position 92
	Pallet corner position opposite to reference	Position 93
	Pallet column counter	Counter 91
	Pallet row counter	Counter 92
	PT9 position	Position 9

※ If the pallet positions (four corner points of the pallet) 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.

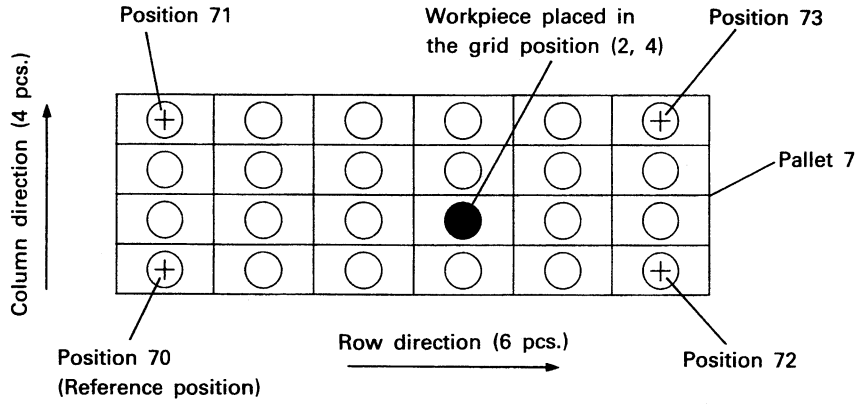
(3) Error mode II occurs if the pallet positions have not been defined and the pallet counters have not been set or have been set to have values exceeding those defined by the PA command. An error does not occur, however, even when the coordinates obtained for the grid point exceed the robot's operational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

(4) The open/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.

[Sample Program]

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.




```
   LPRINT "TL 200"
10 LPRINT "PA 7, 4, 6"
20 LPRINT "SC 71, 2"
30 LPRINT "SC 72, 4"
40 LPRINT "PT 7"
50 LPRINT "MO 7"
```

Step 1) Assume that pallet 7 is used. Define the tool length (in this case, 200mm) corresponding to the hand in use. Then, guide the arm through positions 70, 71, 72 and 73 at four corners of the pallet.

Step 2) Execute the pallet definition command (PA 7, 4, 6) to define the number of grid points in the column and row directions. (Line number 10)

Step 3) Define parameter 2 for counter 71 (column) and parameter 4 for counter 72 (row). These parameters correspond to the target grid point. (Line numbers 20 and 30)

Step 4) Execute the pallet calculation command (PT7). This allows the coordinates of the target grid point to be calculated and they are identified as position 7. The hand can now be moved to that position by MO7. (Line numbers 40 and 50)

PX (Position Exchange)

[Function]

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

[Input Format]

PX 〈Position number (a)〉 , 〈Position number (b)〉

where 1 ≦ Position number (a), (b) ≦ 999

[Sample Input]

PX 2, 3

[Explanation]

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

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

(3) Error mode II occurs if positions (a) and (b) have not been predefined.

[Sample Program]

```
10 LPRINT "HE 2"
20 LPRINT "MJ 20, 30, 0, 0, 0"
30 LPRINT "GO"
40 LPRINT "HE 3"
50 LPRINT "PX 2, 3"
```

※ In the above example, the coordinates and hand open/close state of position 2 are exchanged for those of position 3.

RC (Repeat Cycle)

[Function]

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

[Input Format]

RC ⟨Number of repeated cycles⟩

where 1 ≦ Number of repeated cycles ≦ 32767

[Sample Input]

RC 32

[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 cause the line number following NX to be subsequently executed.

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

[Sample Program]

```
10 LPRINT "20 MO 1"      ; Moves to position 1.
20 LPRINT "30 RC 3"      ; Repeats loop delimited by NX
                           three times.
30 LPRINT "40 MO 2"      ; Moves to position 2.   }
40 LPRINT "50 MO 3"      ; Moves to position 3.   } Loop
50 LPRINT "60 MO 4"      ; Moves to position 4.   }
60 LPRINT "70 NX"        ; Delimits the loop.
70 LPRINT "80 MO 5"      ; Moves to position 5.
```

RN※ (Run)

[Function]

Executes the specified part of instructions in a program.

[Input Format]

```
RN  [<Starting line number>] [ , <Ending line number>]
```

where 1 ≦ Starting line number, ending line number ≦ 3584

[Sample Input]

RN 20, 300

[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 LED display.

(4) If the starting line number is not specified, the program starts with the first line.

(5) When the RN command is executed, the contents of the counter remain intact (not initialized).

[Sample Program]

```
10 LPRINT "100 MO 10"
10 LPRINT "110 MO 12"
30 LPRINT "120 GC"
40 LPRINT "130 MO 17"
50 LPRINT "140 ED"
60 LPRINT "RN 100"          ; Runs program starting with line number 100.
```

RS※ (Reset)

[Function]

Resets the program and error condition.

[Input Format]

RS

[Sample Input]

RS

[Explanation]

(1) Resets an error condition in error mode Ⅰ (hardware error), switching servo from OFF to ON and causing the program to return to its beginning. If any of the axes has exceeded its software limit after origin setting, the error cannot be reset.

(2) Also resets an error condition in error mode Ⅱ (software error), causing the error to be reset and the program to return to its beginning.

(3) The counter values and outputs remain unchanged by resetting any error. (See Section 10 "ERROR CONDITIONS", Vol. 2 OPERATION.)

[Sample Program]

```
10 LPRINT "MO 1000"     ; Causes error mode Ⅱ (defined value greater than 999).
20 LPRINT "RS"          ; Resets error.
```

RT (Return)

[Function]

Completes a subroutine and returns to the main program.

[Input Format]

RT

[Sample Input]

RT

[Explanation]

(1) Completes the subroutine called by the "GS" command and returns to the main program.

(2) Error mode II occurs if the corresponding "GS" command is not specified.

[Sample Program]

See the GS command.

SC (Set Counter)

[Function]

Loads the specified value into the specified counter.

[Input Format]

SC 〈Counter number〉 , [〈Set value〉]

where 1 ≦ Counter number ≦ 99
−32768 ≦ Set value (decimal) ≦ +32767
or &8000 ≦ Set value (hexadecimal) ≦ &7FFF

[Sample Input]

SC 15, 123

[Explanation]

(1) All counters are factory-set to 0.

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

(3) The default set value is 0.

(4) The contents of the counter can be changed, compared or read by the relevant instruction. (See the IC, DC, CP, CR, CL, AN, OR and XO commands.)

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

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

[Sample Program]

```
10 LPRINT "SC 21, 15"
20 LPRINT "IC 21"
```

※ In the above example, line number 10 sets value 15 in counter 21 and line number 20 causes the counter to be incremented by 1.

SD (Speed Define)

[Function]

Defines the moving velocity and first order time constant of the end of the hand for linear interpolation.

[Input Format]

SD 〈Moving speed〉 [, 〈First order time constant〉]

where 0 < Moving speed ≦ 500.0 (mm/sec)
80 ≦ First order time constant ≦ 300 (msec)

[Sample Input]

SD 125.0, 100

[Explanation]

(1) The minimum input increment of the moving velocity is 0.1mm/second or 0.1 degree/second (e.g. specify 20.5 for 20.5mm/second). The minimum input increment of the first order time constant is 1 millisecond (with upper and lower limits).

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

(3) Setting a larger value to the first order time constant makes the robot operation slower and smoother during linear interpolation. When the robot turns a corner by the MC command, the SD command causes the path to be nearer to the inside. When the first order time constant is not specified, it defaults to 80msec.

(4) The velocity set by the SD command is valid until a new value is set by the SD or SP command.

(5) During linear interpolation, a certain moving speed set value of the SD command may cause error mode II in excess of the maximum speed of the corresponding axis depending on the robot attitude. In this case, set the moving speed of the SD command to a lower value.

(6) The moving velocity of the articulated interpolation command is not determined by the value of the SD command but by that of the SP command.

(7) At power on, the moving velocity is initialized to SP8, H.

[Sample Program]

```
10 LPRINT "SP 15"      ; Sets speed to 15.
20 LPRINT "MS 1"       ; Moves to position 1 by linear interpolation. (SP15)
30 LPRINT "SD 100"     ; Sets moving speed to 100mm/second.
40 LPRINT "MS 2"       ; Moves to position 2 by linear interpolation. (Linear
                         velocity 100mm/second)
50 LPRINT "MO 3"       ; Moves to position 3 by articulated interpolation
                         (SP15)
60 LPRINT "MS 4"       ; Moves to position 4 by linear interpolation. (Linear
                         velocity 100mm/second)
```

SF (Shift)

[Function]

Shifts the coordinates of the specified position in increments representing those of another specified position and redefines the new coordinates.

[Input Format]

SF 〈Position number (a)〉 , 〈Position number (b)〉

where 1 ≦ Position number (a), (b) ≦999

[Sample Input]

SF 10, 100

[Explanation]

(1) Adds each coordinate of position (b) to the corresponding coordinate of position (a). The hand open/close state at position (a) is not affected by the SF command.

(2) Error mode II occurs if positions (a) and/or (b) have not been predefined.

(3) Does not effect any robotic motion.

[Sample Program]

```
10 LPRINT "PD 5, 0, 0, 30, 0, 0"
20 LPRINT "HE 1"
30 LPRINT "SF 1, 5"
40 LPRINT "MO 1"
```

※ In the above example, Z-coordinate 30mm is added to the corresponding coordinate of position 1 and the new coordinates are defined as position 1. Then, the robot moves to that point.

SM (If Smaller)

[Function]

Causes a jump to occur if the internal register value is smaller than the specified value when compared.

[Input Format]

SM 〈Compared value〉 , 〈Jump destination line number〉

where −32768 ≦ Compared value (decimal) ≦ +32767
or &8000 ≦ Compared value (hexadecimal) ≦ &7FFF
1 ≦ Jump destination line number ≦ 3584

[Sample Input]

SM 128, 1024 or SM &80, 1024

[Explanation]

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

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

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

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

[Sample Program]

```
10 LPRINT "100 ID"              ; Fetches data from external input port.
20 LPRINT "110 SM 100, 130"     ; Jumps to line number 130 if input data is smaller
                                  than 100.
30 LPRINT "120 ED"              ; Ends the program if above condition is not met.
40 LPRINT "130 MO 7"            ; Moves to position 7.
```

SP (Speed)

[Function]

Sets the operating velocity and acceleration/deceleration time for the robot.

[Input Format]

SP 〈Speed level〉 [, 〈H or L〉]

where 0 ≦ Speed level ≦ 20

[Sample Input]

SP 15, H

[Explanation]

(1) Sets the operating velocity in 21 steps and the acceleration/deceleration time upon start and stop in 2 steps.

(2) The speed level is predetermined as a ratio to the maximum rpm of each axis for articulated interpolation and as a ratio to the maximum velocity of the end of the hand (222mm/second) or to the maximum position angular velocity (222!/second) for linear interpolation.

(3) The acceleration/deceleration time may be selected from among H, max. 0.2 seconds, or L, max. 0.4 seconds (for articulated interpolation only).

(4) When the velocity and acceleration/deceleration time have been defined for articulated interpolation, the acceleration/deceleration distance required for movement is predetermined. This means that the set speed may not be reached if the distance of travel is small.

(5) For linear interpolation, the end of the hand is moved at constant speed determined by the tool instruction. In this case, error mode II may result from any of the axes exceeding its maximum speed depending on the robot position. If the motion of the position angle (wrist pitch, roll) is greater than the motion of the distance (X, Y, Z), the robot moves in reference to the position angular velocity. (The speed of the end of the hand is made lower.) 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 SP8, H. (The most recent acceleration/deceleration time remains valid when it is not specified.)

(7) If the speed parameter is not specified, it defaults to 0.

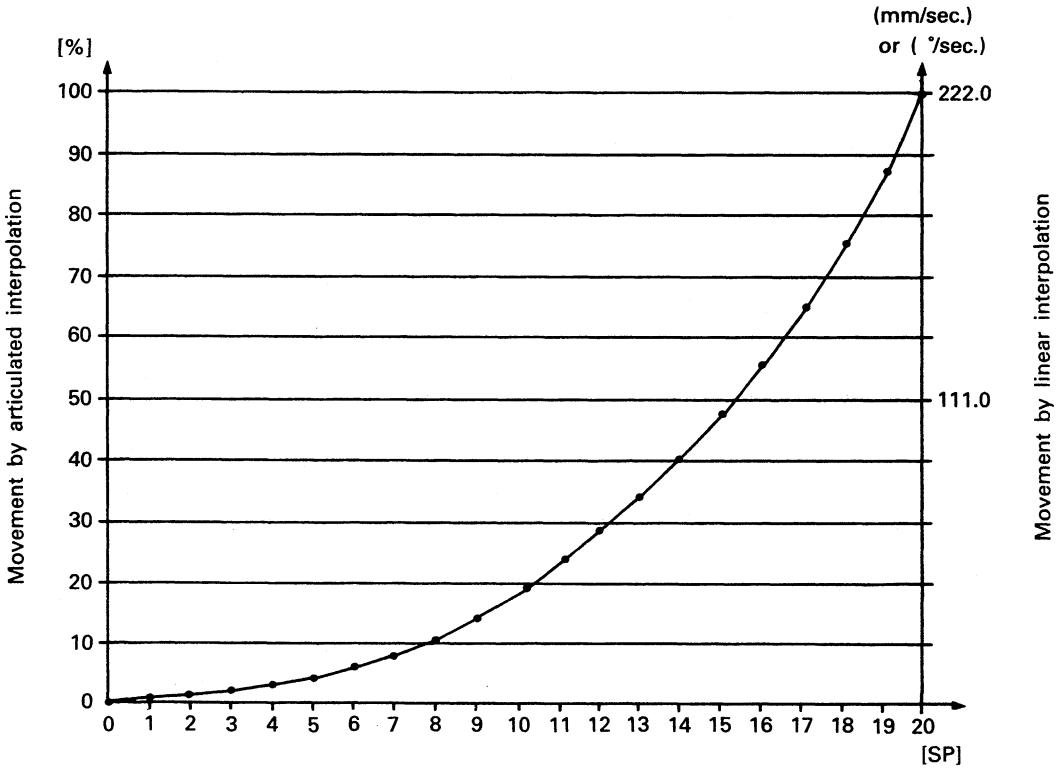
[Sample Program]

```
10 LPRINT "SP 8"          ; Sets speed to 8.
20 LPRINT "MO 5"          ; Moves to position 5. (Articulated interpolation)
30 LPRINT "SP 10"         ; Sets speed to 10.
40 LPRINT "MS 7"          ; Moves to position 7. (Linear interpolation)
```

Relationship between Speed Parameters and Velocity

SP	%	Velocity (mm/sec.) *1	SP	%	Velocity (mm/sec.) *1
0	0.1	0.2			
1	1.2	2.7	11	23.9	53.1
2	1.7	3.8	12	28.5	63.3
3	2.4	5.3	13	33.9	75.3
4	3.3	7.3	14	40.2	89.2
5	4.4	9.8	15	47.4	105.2
6	6.0	13.3	16	55.7	123.7
7	8.0	17.8	17	65.1	144.5
8	10.7	23.8	18	75.7	168.1
9	14.3	31.7	19	87.3	193.8
10	19.0	42.2	20	100.0	222.0

Note 1: The robot moves according to the angular velocity (°/sec.) if the motion of the position angle (wrist pitch, roll) ≧ that of the distance (X, Y, Z).



TB (Test Bit)

[Function]

Causes a jump (or no jump) to occur in accordance with the specified bitvalue in the internal counter.

[Input Format]

TB 〈+ or −〉 〈Bit number〉 , 〈Jump destination line number〉

where 0 ≦ Bit number ≦ 17
1 ≦ Jump destination line number ≦ 3584

[Sample Input]

TB +7, 1024

[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 comparison register is on (with a positive (+) sign) or off (with a negative (−) sign) (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 comparison register by executing the input instruction (see ID and IN) for the external input data or by executing the compare counter instruction (see CP) for the counter data. It is therefore necessary to execute either of the above instructions beforehand so that a conditional jump can occur.

[Sample Program]

```
10 LPRINT "100 ID"          ; Fetches data from external input port.
20 LPRINT "110 TB +1, 130"  ; Jumps to line number 130 if first bit of input data
                              is on.
30 LPRINT "120 ED"          ; Ends program if above condition is not met.
40 LPRINT "130 MO 7"        ; Moves to position 7.
```

TI (Timer)

[Function]

Halts the motion for the specified length of time.

[Input Format]

TI 〈Timer counter〉

where 0 ≦ Timer counter ≦ 32767

[Sample Input]

TI 20

[Explanation]

(1) Causes the robot to halt its motion for the following period of time:
Specified timer counter value×0.1 seconds (max. 3276.7 seconds)

(2) Used to introduce a time delay before and after the hand is opened and closed for gripping a workpiece.

(3) The default value is 0.

[Sample Program]

```
10 LPRINT "MO 1, O"      ; Moves to position 1 (with hand open).
20 LPRINT "TI 5"         ; Sets timer to 0.5 sec.
30 LPRINT "GC"           ; Closes hand (to grip workpiece).
40 LPRINT "TI 5"         ; Sets timer to 0.5 sec.
50 LPRINT "MO 2, C"      ; Moves to position 2 (with hand closed).
```

TL (Tool)

[Function]

Establishes the distance between the hand mounting surface and the end of the hand. (Tool center point)

[Input Format]

TL [<Tool length>]

where 0 ≦ Tool length ≦ +300.0(mm)

[Sample Input]

TL 145

[Explanation]

(1) The least input increment of the tool length is 0.1mm (e.g. specify 200.5 for 200.5mm).

(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 setting of the tool length is 123mm.)

(3) The default value is 0.

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

(5) 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]

```
10 LPRINT "TL 120"
20 LPRINT "HE 1"
30 LPRINT "TL 100"
40 LPRINT "MO 1"
```

※ In the above example, line number 30 changes the tool length and then line number 40 causes the end of the hand to advance 20mm in the tool direction.

TR※ (Transfer)

[Function]

Transfers the program and position data stored in EPROM to the drive unit.

[Input Format]

TR

[Sample Input]

TR

[Explanation]

(1) Causes the contents of the EPROM installed in the user socket inside the drive unit side door to be transferred to the RAM. After the TR command has been executed, the old program and position data stored in the RAM is all cleared.

(2) The "EXECUTE" LED on the drive unit front panel is instantaneously lit up when the data is transferred.

(3) When bit 4 of SW1 inside the drive unit side door is in the upper (ON) position, the TR command also causes the reference position data for origin setting stored in the EPROM to be transferred to the RAM. This data transfer is not made when bit 4 of SW 1 is in the lower (OFF) position (i.e. data in the RAM is valid).

[Sample Program]

```
10 LPRINT "TR"
```

WH (Where)

[Function]

Reads the coordinates of the current position and the open/close state of the hand (using RS232C).

[Input Format]

WH

[Sample Input]

WH

[Explanation]

(1) Causes the coordinates of the current position of the end of the hand, as determined by the tool length currently being established (by the TL command), and the hand open/close state to be output from the RS232C port.

(2) The data is output in ASCII as indicated below. The least output increment is 0.1mm or 0.1! (e.g. 20.1 is displayed for 20.1mm):
Output format: X-axis coordinate, Y-axis coordinate, Z-axis coordinate, pitch angle, roll angle, O or C 0D (Hex)
In the above format, the hand open state is output as O and the close state as C.

(3) The delimiter of the data is a comma (,: Hex. 2C) and the terminator is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, it is necessary to handle the entire data string up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

[Sample Program]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT#1, "WH"
30 LINE INPUT#1, A$
40 PRINT "CURRENT POSITION =" ;A$
50 END
   RUN ↓
   CURRENT POSITION = +10.0, +380.0, +300.0, −70.0, −40.0, C
```

※ In the above example, which uses Mitsubishi's MAXY, line number 10 opens the RS232C communication file and line number 20 transmits the "WH" command. Line number 30 then uses the LINE INPUT # statement to store the received data in A$ and line number 40 outputs the data to the display screen.

WT (What Tool)

[Function]

Reads the tool length currently being established (using RS232C).

[Input Format]

WT

[Sample Input]

WT

[Explanation]

(1) Causes the tool length currently being established (by the TL command) to be output from the RS232C port.

(2) The data is output in ASCII coded decimal. The least output increment is 0.1mm (e.g. 105.7 is displayed for 105.7mm).

(3) The terminator of the data is a carriage return (CR: Hex. 0D). If the data is to be received by a personal computer, it is necessary to handle the entire data string up to hex. 0D. The BASIC equivalent to this is the LINE INPUT # statement.

(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]

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT#1, "WT"
30 LINE INPUT#1, A$
40 PRINT "TOOL =" ;A$
50 END
RUN↓
TOOL = 105.7
```

※ In the above example, which uses Mitsubishi's MAXY, line number 10 opens the RS232C communication file and line number 20 transmits the "WT" command. Line number 30 then uses the LINE INPUT # statement to store the received data in A$ and line number 40 outputs the data to the display screen.

WR※ (Write)

[Function]

Writes the generated program and position data into EPROM.

[Input Format]

WR

[Sample Input]

WR

[Explanation]

(1) Causes the program and position data generated in the drive unit RAM to be written to the EPROM installed in the user socket inside the drive unit side door. The EPROM must have been erased before the WR command is executed.

(2) The "EXECUTE" LED on the drive unit front panel stays lit while the data is being written to the EPROM. It goes out as soon as all data has been written (which takes about 3 minutes 40 seconds).

(3) Error mode II occurs if the EPROM has not been erased or a write error takes place.

[Sample Program]

```
10 LPRINT "WR"
```

XO (Exclusive Or)

[Function]

EXCLUSIVE ORs the specified data and the internal register data.

[Input Format]

XO ⟨Data to be operated⟩

where −32768 ≦ Data to be operated (decimal) ≦ +32767
&8000 ≦ Data to be operated (hexadecimal) ≦ &7FFF

[Sample Input]

XO &000F

[Explanation]

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

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

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

[Sample Program]

```
10 LPRINT "100 ID"          ; Enters data from external input port.
20 LPRINT "110 AN &000F"    ; Fetches 4 lower bits only.
30 LPRINT "120 XO &000F"    ; Flips data of 4 lower bits to their opposite settings.
40 LPRINT "130 CL 21"       ; Sets above data to counter 21.
50 LPRINT "140 EQ 10,200"   ; Jumps to line number 200 if above data equals 10.
60 LPRINT "150 ED"          ; Ends program.
70 LPRINT "200 MO 99"       ; Moves to position 99.
```

' (Comment)

[Function]

Allows the programmer to write a comment.

[Input Format]

' [String of information consisting of up to 7 alphanumeric characters]

[Sample Input]

'MELCO

[Explanation]

(1) Allows the programmer to write a comment consisting of up to seven alphanumeric characters following the ' (apostrophe).

(2) Used to write the name and date on the generated program or to mark a subroutine. Comments are helpful in later understanding of the program as the LR (Line Read) command is used.

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

(4) If the number of characters exceeds 7, the whole excess is ignored.

[Sample Program]

```
10 LPRINT "1 'DATE"
20 LPRINT "2 '890311"
30 LPRINT "3 NT"
40 LPRINT "4 MO 1"
```

※ In the above example, line numbers 1 and 2 write the date when the program was written (March 11, 1989).

1. SPECIFICATIONS
2. OPERATION
3. DESCRIPTION OF THE COMMANDS
4. MAINTENANCE AND INSPECTION
5. APPENDICES

CONTENTS (MAINTENANCE AND INSPECTION)

4. MAINTENANCE AND INSPECTION

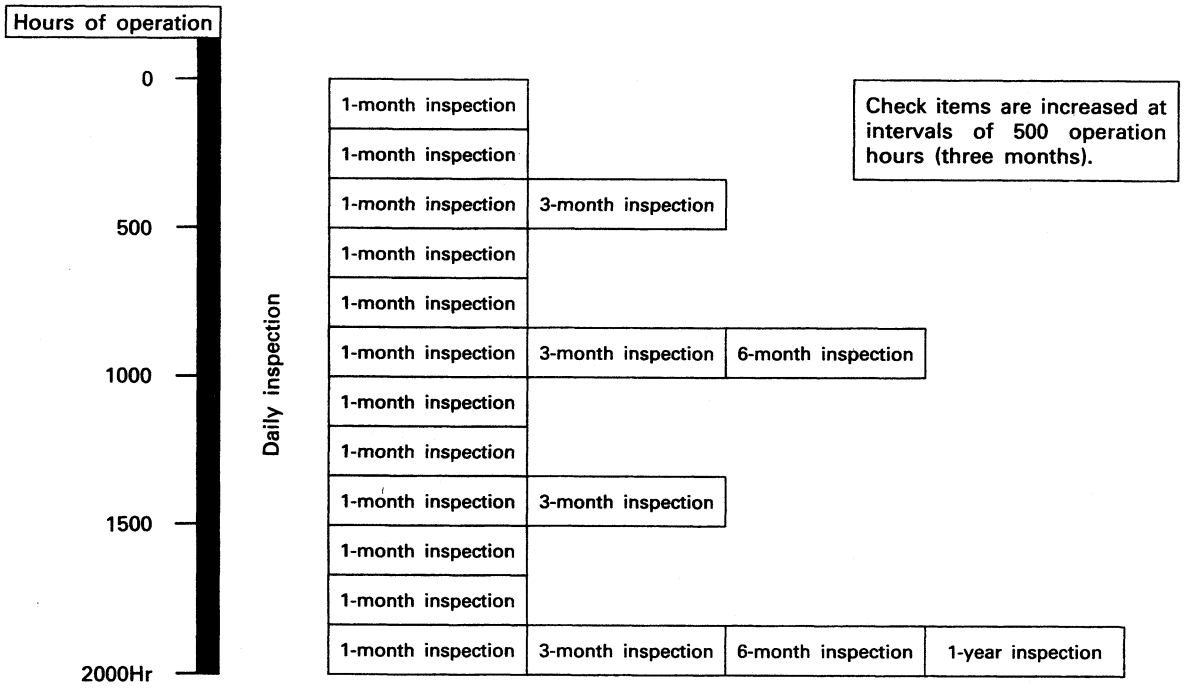
1. MAINTENANCE AND INSPECTION SCHEDULE

The maintenance and inspection procedures on the following pages are mandatory to ensure the best performance for the robot over a long period of time. Be sure to perform the daily as well as periodic inspection procedures.

The periodic inspection procedure must be performed independently of the daily inspection procedure.

Inspection Schedule

Check items should be added at intervals of 500 operation hours as indicated below:



- Guide to inspection intervals

1-shift:
8 hours/day × 20 days × 3 months = approx. 500 hours
10 hours/day × 20 days × 3 months = approx. 600 hours

2-shifts:
15 hours/day × 20 days × 3 months = approx. 1,000 hours

2. INSPECTION ITEMS

2.1 Daily Inspection

The daily inspection procedure must be performed before each day's operation, even with the 24-hour operation. Table 4.2.1 lists the daily inspection items and procedures.

Step	Inspection Item (Checking Method)	Remedial Action
Before powering up	Check the following items 1 to 5 before powering up.	
1.	Check that the power cable is securely connected. (Visual)	Connect securely.
2.	Check that the cable is securely connected between the robot and drive unit. (Visual)	Connect securely.
3.	Ensure that the robot and drive unit are free of cracking and contaminants. Make sure they are clear of obstacle (for origin setting). (Visual)	Change or take necessary action.
4.	Check that no grease leaks from the wrist.	Supply grease after cleaning.
5.	Check that the pneumatic line is without fault, e.g. air leakage, accumulated drain, hose breakage, air source fault.	Take necessary action, e.g. drain the line, repair air leakage, change parts.
Powering up	Power up, being attentive to possible robot motion.	
6.	Ensure that the robot operates without fault or unusual noise when the power is switched on. (Visual, etc.)	Take necessary action in accordance with Section 2.3 Troubleshooting.
Operation	Return the robot to origin and run your program.	
7.	Ensure that the point of axis motion is properly aligned. If it is out of alignment, check for the following: 1) Installation bolts left loose (Use a tool to retighten.) 2) Hand installation bolts left loose (Use a tool to retighten.) 3) Jigs out of alignment (Visual, etc.) 4) If the problem still persists, use procedures shown in Section 2.3 Troubleshooting.	Take necessary action according to Section 2.3 Troubleshooting. Retighten. Retighten. Correct position.
8.	Ensure that there is no unusual noise or vibration. (Visual, etc.)	Take necessary action according to Section 2.3 Troubleshooting.

Table 4.2.1 Daily Inspection Items

If no trouble is noted

Start automatic operation after checking that the peripherals operate without any fault.

If trouble is noted

Make checks according to Section 2.3 "Troubleshooting."

2.2 Periodic Inspection

Table 4.2.2 lists the inspection items and procedures at each inspection frequency. Part of the inspection and adjustment procedures as well as parts changing procedures is detailed in Section 3.

Step	Inspection Item	Inspection Procedure and Remedial Action	Cross-Reference Section
1-monthly inspection			
1.	Check that the hand installation bolts are tight.	Use a wrench to check tightness. Retighten if necessary.	
2.	After removing covers, check cables for scratches, damages, and contaminants.	Isolate and eliminate the cause. Change if damage is excessive.	Section 3.2
3.	Check that bolts and screws, both visible and invisible, are tight.	Use a wrench and screwdriver to check tightness. Retighten if necessary.	
3-monthly inspection			
11.	Check the detector of the position detection switch for dust or iron filings.	Clean.	
12.	Supply grease to the J5 gear.	See Section 3.5 "Supplying Grease."	Section 3.5
13.	After removing the pulley covers (L), (R), check the timing belt tension.	Adjust as required.	Section 3.4
6-monthly inspection			
31.	Check DC servo motor brushes for wear.	Replace the brushes if wear exceeds the limit.	Section 3.3
32.	Check timing belt teeth for wear.	Change the belt if teeth are broken or excessively worn.	Section 3.4
Yearly inspection	Contact our Service Division for yearly inspection because part of its items is highly technical.		
51.	Change DC servo motor brushes.	If brushes have not been changed within the past one year, change in accordance with Section 3.3.	Section 3.3.1 Section 3.3.2
52.	Change J4, J5 timing belts.	If the belt has not been changed within the past one year, change in accordance with Section 3.4.	Section 3.4.2 Section 3.4.3 Section 3.4.4
53.	Check origin limit switch operation (limit pulse).	Adjust as necessary.	Section 3.8
54.	Disassemble and change grease in the harmonic drive reduction gear for each axis.	Disassemble the harmonic drive reduction gear and check teeth for wear and damage. Then, change or apply grease.	Contact our Service Division.
55.	Check cables used in bent sections inside the robot.	Change the cable if its resistance value is 0.3 or higher.	

Table 4.2.2 Periodic Inspection Items

2.3 Troubleshooting

Table 4.2.3 lists troubleshooting procedures that may be used should the system develop any fault upon start-up or during robotic operation.

Symptom	Possible Cause	Remedy
Power remains off.	1) Power cord left unplugged from the outlet.	• Connect properly.
	2) Fuse blown.	• Change the fuse.
	3) Wrong source voltage.	• Use correct line voltage.
Robot inoperative.	1) ERROR LED is on or flickering.	• Check the command, etc. • Connect cables properly. • Check that the emergency stop switch is off.
	2) Robot arm in contact with mechanical stopper or peripheral.	• Keep the robot clear of obstacles.
	3) Motor power cable left unplugged. 4) Operation end.	• During the wrist pitch (J4) operation before origin setting, wrist roll operation end error may occur before the wrist pitch reaches its operation end. In this case, move the wrist roll in reverse direction and operate the wrist pitch.
Robot cannot return to origin.	1) Return to origin not invoked.	• Check the commands and teaching box key operation.
	2) Return to origin is not terminated.	• Check the limit switch operation. • Encoder Z-phase may be out of adjustment. Contact our Service Division. (See Section 3.8.)
Program not written from personal computer.	1) Teaching box select switch is in ON position.	• Set to OFF.
	2) ERROR LED on.	• Check the command input format. (See Section 10.2 "Error Mode II", Vol. 2 Operation.)
	3) Drive unit mode.	• Select personal computer mode. (Place drive unit ST1 in the lower position.)
	4) Reset switch not pressed after the stop switch has been pressed during execution of a program.	• Press the reset switch.
	5) Personal computer connection cable left disconnected.	• Connect the cable properly.
Stop during operation.	1) ERROR LED flickering.	• Connect or change the cables properly. • Check for excessive robot load. • Check for possible contact of arm with mechanical stopper or peripheral. If OK, proceed to "Poor repeatability."
	2) Line voltage drop	• Use the specified voltage range.
	3) Power failure including instantaneous	• Restart from origin setting.
	4) Emergency stop switch pressed accidentally.	• Switch power off, then on and start up from origin setting.
	5) Motor developing unusual smell	• Change the motor.
	6) Driving parts developing unusual noise or vibration	• Progress to "Unusual noise or vibration."

Table 4.2.3 Troubleshooting (Continue)

Symptom	Possible Cause	Remedy
Poor repeatability	1) Unusual noise developing 2) Robot not secured in position 3) Screws in drive transmission system left loose 4) Timing belt slack off 5) Unusual return-to-origin operation 6) Match marks not matching with each other after origin setting. 7) Hand installation screws left loose.	• Remove the noise source. • Secure the robot in position. • Tighten screws securely. • Adjust timing belt tension. (See Section 3.4.) • Go back to "Robot cannot return to origin." • Tighten screws securely.
Unusual noise or vibration	1) Robot installation screws left loose 2) Harmonic drive developing unusual noise or vibration increased suddenly.	• Tighten screws securely. • Harmonic drive may have been damaged. Contact our Service Division.
Motor-operated hand cannot be opened or closed.	1) Hand select switch in wrong position. 2) Wrong parameter defined for GP command. 3) Hand short of grease.	• Select hand select switch to either AC or DC in accordance with the type of the hand being used. • Correct parameter. • Pack worm gear area with grease. (Consult our Service Division.)
Air hand cannot be opened or closed.	1) Hand select switch in wrong position. 2) Air not supplied. 3) Wrong solenoid valve used.	• Select hand select switch to either AC or DC in accordance with the type of the hand being used. • Supply air. • Use solenoid valve of appropriate specifications (see p. 1-11, Vol. 1.)
No output provided.	1) External line voltage and/or polarity wrong. 2) Command OD, OT or OB used improperly. 3) Internal output transistor damaged due to short circuit of load.	• Use at correct voltage (12 to 24V DC) and polarity. • Use commands correctly. • Change I/O card IC.
Robot acted unpredictably.	1) ERROR LED flashing. 2) Arm sagged at power-on. 3) Given axis misoperated. 4) Robot cables not changed within a year.	• Check all points on the left and contact our Service Division.

Table 4.2.3 Troubleshooting

3. MAINTENANCE AND INSPECTION PROCEDURES

The following parts must be changed every 2,000 hours of the Movemaster RV-M2 operation. (It is the manufacturer's recommendation and more frequent replacement is necessary depending on the operating condition.)
The following pages contain the replacement procedures of these parts to aid the service personnel in changing parts at the customer's site. Read the procedures carefully to make yourselves familiarize with them.
The replacement work may be done by our Service Center at customer's expense.
Never attempt to disassemble parts not covered on the following pages.

Parts to be changed every 2,000 hours of operation:

1. DC servo motor brushes (all axes)
2. Timing belts (for wrist pitch, wrist roll)

- The types and quantities of these parts are listed in Section 4 "SERVICE PARTS", Vol. 4 MAINTENANCE AND INSPECTION.

WARNING

When performed, the replacement procedures may result in misaligned mechanical origin. The position data must therefore be reviewed after the procedures have been performed.

3.1 Construction of the Robot

Fig. 4.3.1 shows the general construction of the Movemaster RV-M2.

(1) Waist (J1 axis) sweep

- The waist (J1 axis) is driven by the J1 axis motor ① and harmonic drive reduction gear ② located inside the base.
- The J1 axis photoelectric switches ③ are installed inside the base rear cover.

(2) Shoulder (J2 axis) swivel

- The shoulder (J2 axis) is driven by the J2 axis motor ④ and J2 axis harmonic drive reduction gear ⑤ which are located in the upper arm.
- The J2 axis DC servo motor ④ is provided with an electromagnetic brake that prevents the shoulder from swiveling down by its own weight when the power is switched off.
- The J2 axis photoelectric switches ⑥ are installed in the shoulder cover.

(3) Elbow (J3 axis) extension

- The elbow (J3 axis) is driven by the J3 axis motor ⑦ and J3 axis harmonic drive reduction gear ⑧ which are located in the fore arm.
- As with the J2 axis, the J3 axis motor ⑦ is provided with the electromagnetic brake.
- The J3 axis photoelectric switches ⑨ are installed in the elbow cover.

(4) Wrist pitch (J4 axis)

- Rotation of the J4 axis motor ⑩ located in the fore arm is transmitted via the J4 axis timing belt ⑪ to the J4 axis harmonic drive reduction gear ⑫; as a result, the wrist housing and beyond are rotated.
- The J4 axis proximity switch ⑬ is installed in the pulley cover (R).

(5) Wrist roll (J5 axis)

- Rotation of the J5 axis motor ⑭ located in the fore arm is transmitted via the J5 axis timing belt ⑮ to the J5 axis harmonic drive reduction gear ⑯, which then rotates the mechanical interface via the bevel gear ⑰.
- The J5 axis proximity switches ⑱ are installed in the pulley cover (L).

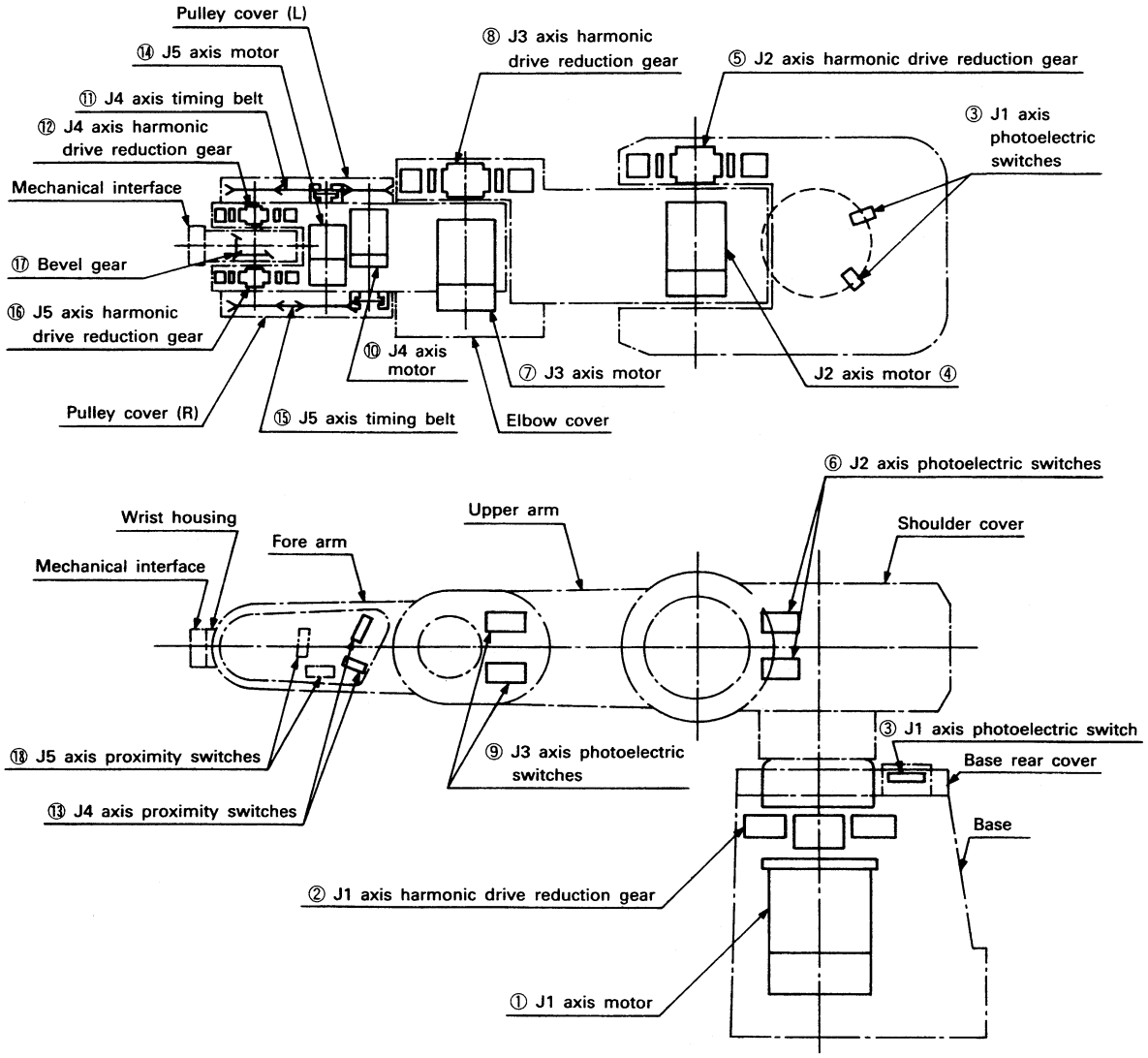


Fig. 4.3.1 General RV-M2 Construction

3.2 Removal of Covers

(1) Whenever the robot is to be serviced, remove the covers as shown in Fig. 4.3.2.

(2) Tables 4.3.1 and 4.3.2 list the covers and installation screws, respectively.

(3) Some covers may be hard to remove depending on the position of the robot. In such cases, move the robot as appropriate so that the covers can be removed easily.

(4) Reinstall the covers in reverse order of removal.

Key	Cover Name	Qty
[1]	Base side cover (R)	1
[2]	Base side cover (L)	1
[3]	Base rear cover	1
[4]	Base front cover	1
[5]	Shoulder cover, upper	1
[6]	Shoulder cover, lower	1
[7]	Upper arm cover	1
[8]	Elbow cover (R)	1
[9]	Card cover	1
[10]	Pulley cover (R)	1
[11]	Pulley cover (L)	1

Table 4.3.1 Cover Names

Key	Screw Name	Qty
ⓐ	Cross-recessed pan head screw M4 × 10	8
ⓑ	Hexagon socket head bolt M3 × 6	6
ⓒ	Hexagon socket head bolt M3 × 10	6
ⓓ	Cross-recessed truss head screw M3 × 6	11
ⓔ	Cross-recessed truss head screw M3 × 35	3

Table 4.3.2 Installation Screw List

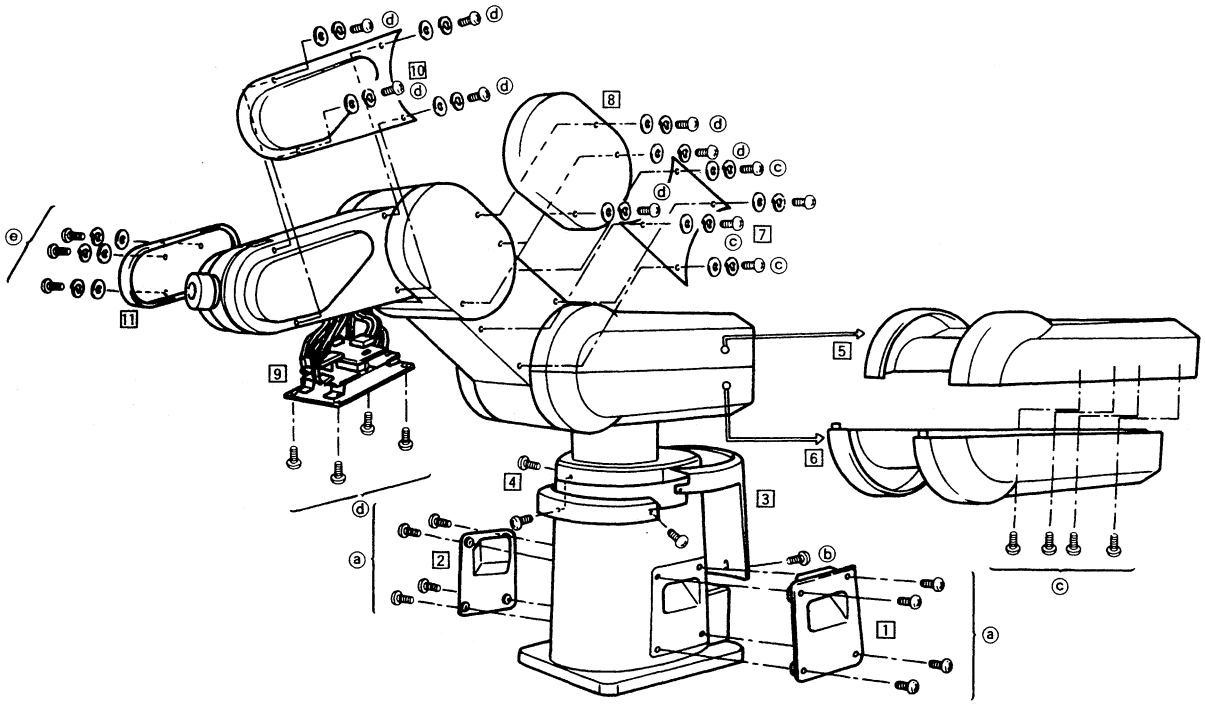


Fig. 4.3.2 Removal of Covers

3.3 Replacement of the Motor Brushes

The RV-M2 robot uses DC servo motors which require their brushes to be checked and changed at given intervals. The brush life varies greatly depending on the operating conditions. As a guide, check and change the motor brushes every 2,000 hours of operation.

3.3.1 Inspection and replacement of the waist drive motor brushes

A. Inspection

(1) Fig. 4.3.3 outlines the inspection and replacement procedures of the waist drive motor brushes.

(2) Remove both base side covers in accordance with Section 3.2 Removal of Covers.

(3) Using a flat-blade screwdriver, turn the heads of the two brush holder caps ① counterclockwise to remove the brush holder caps ①.

(4) Fig. 4.3.4 shows the shape of the brush and its wear limit length. Check the brush for wear and change before the wear limit is reached.

(5) If the brush is damaged, change it independently of its length.

(6) Check the brushes one location at a time. When they are reinstalled, make sure of the correct installation position and direction.

B. Replacement

(1) Remove all brushes in accordance with the procedure in A.

(2) Using a vacuum cleaner, remove all carbon particles from the brush holder pockets.

(3) Carbon particle residual can cause loss of insulation and rectification fault between grounding points. Be sure to remove all particles.

(4) Install new brushes.

(5) After the new brushes have been installed, move each axis of motion its maximum stroke at the maximum speed or run the robot for 30 minutes using the user program.

NOTES:

1) Before installing the new brushes, make sure they are free from grease, oil and moisture. Never handle the brushes with grease-stained hands or place them in locations subject to contamination with grease and oil.

2) When removing the carbon particles, never subject the motor to grease, oil and moisture.

3) The brush holder cap ① is provided with a O-ring. Do not attempt to remove the O-ring from the cap.

4) Tighten the brush holder cap ① to 4kgf-cm.

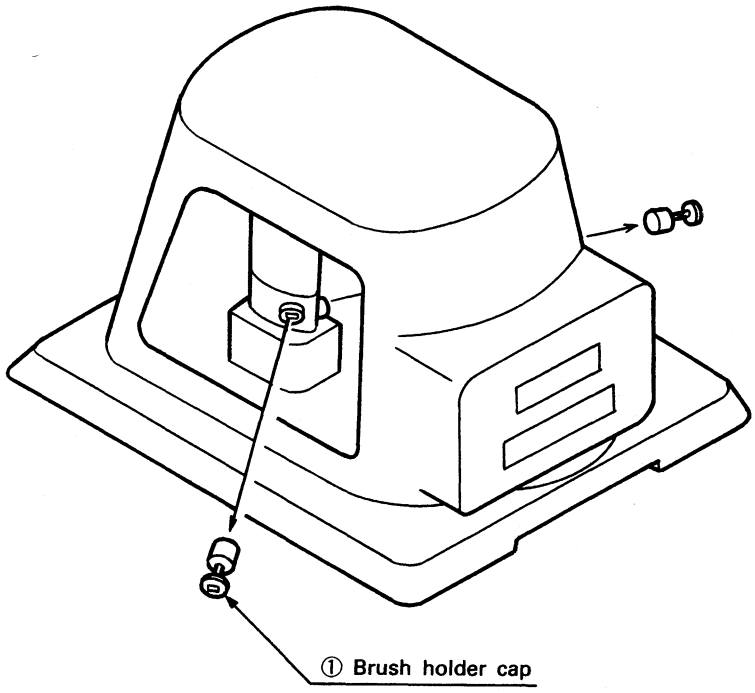


Fig. 4.3.3 Checking and Changing the Waist Drive Motor Brushes

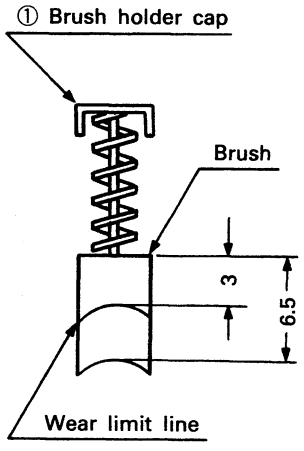


Fig. 4.3.4 Brush Shape

3.3.2 Inspection and replacement of the shoulder drive motor brushes

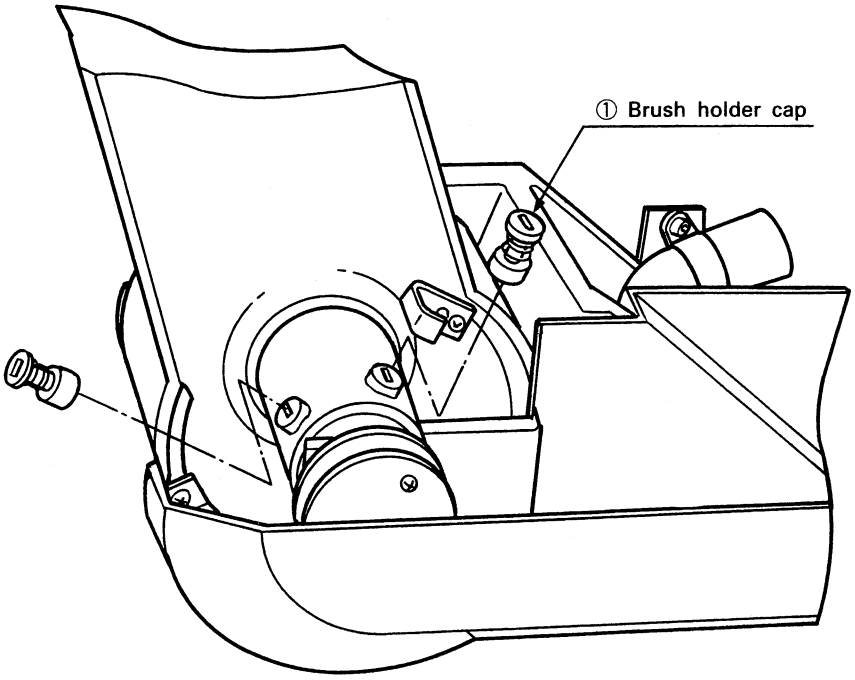


Fig. 4.3.5 Checking and Changing the Shoulder Drive Motor Brushes

A. Inspection

(1) Fig. 4.3.5 outlines the inspection and replacement procedures of the shoulder drive motor brushes.

(2) Remove the upper shoulder cover and upper arm cover in accordance with Section 3.2, Removal of Covers.

(3) Check the brushes in accordance with steps (3) to (6), A. Inspection of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

B. Replacement

(1) Change the brushes in accordance with steps (1) to (5), B. Replacement of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

3.3.3 Inspection and replacement of the elbow drive motor brushes

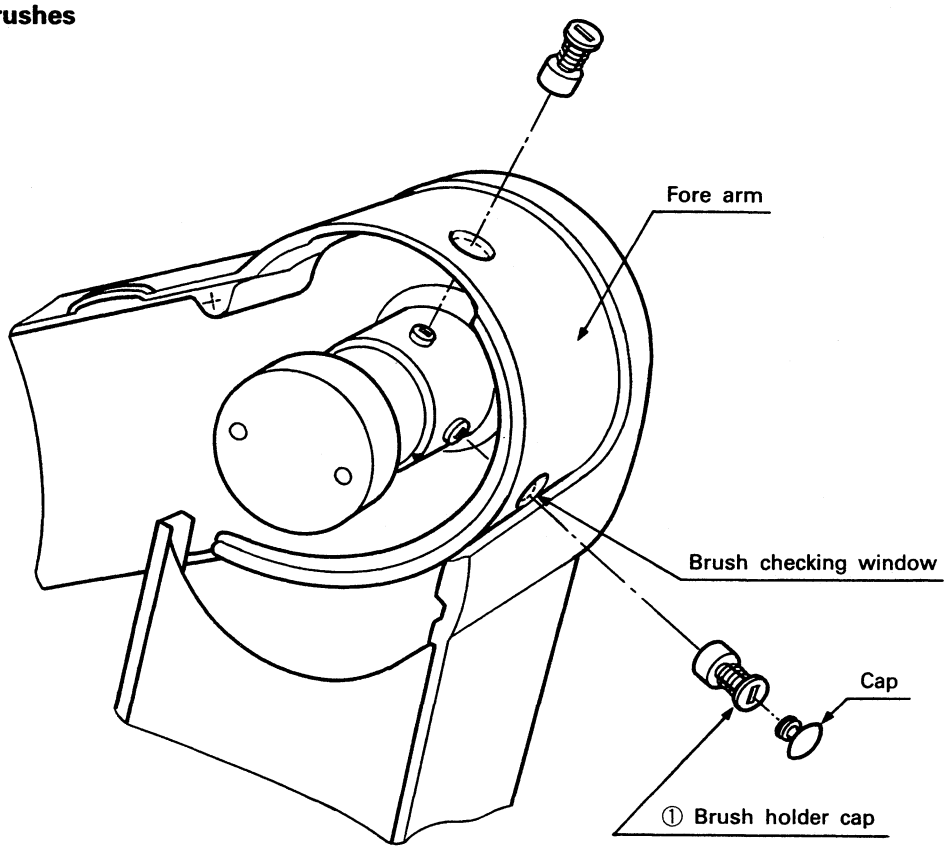


Fig. 4.3.6 Checking and Changing the Elbow Drive Motor Brushes

A. Inspection

(1) Fig. 4.3.6 outlines the inspection and replacement procedures of the elbow drive motor brushes.

(2) Remove the elbow cover (R) in accordance with Section 3.2, Removal of Covers.

(3) After removing the cap from the brush checking window in the fore arm, insert a flat-blade screwdriver into the window and remove the brush holder cap.

(4) Check the brushes in accordance with steps (3) to (6), A. Inspection of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

B. Replacement

(1) Change the brushes in accordance with steps (1) to (5), B. Replacement of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

3.3.4 Inspection and replacement of the wrist pitch and wrist roll drive motor brushes

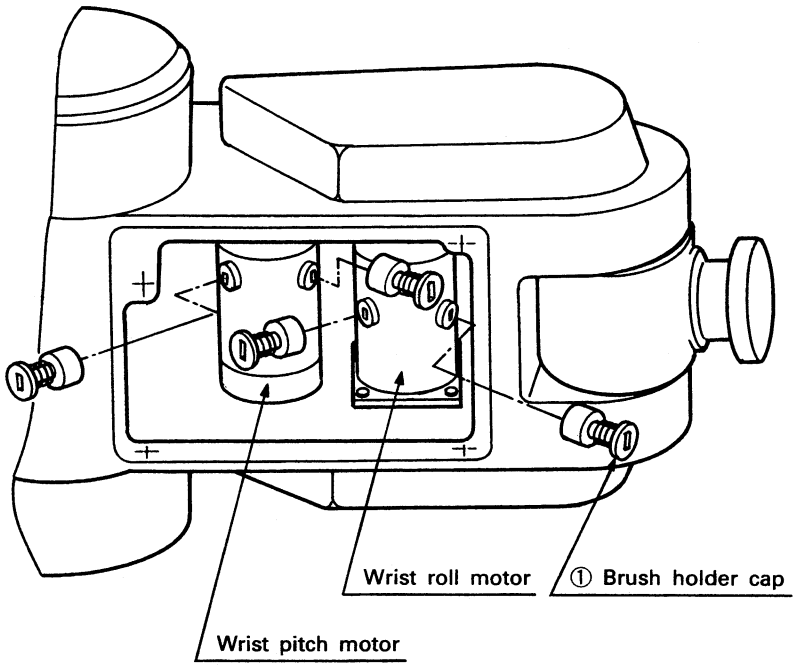


Fig. 4.3.7 Checking and Changing the Wrist Pitch and Wrist Roll Drive Motor Brushes

A. Inspection

(1) Fig. 4.3.7 outlines the inspection and replacement procedures of the wrist pitch and wrist roll drive motor brushes.

(2) Remove the pulley cover (R) and card cover in accordance with Section 3.2, Removal of Covers. Remove the card cover with special care so that the cables connected to the printed circuit board on the card cover may not be disconnected or broken.

(3) Check the brushes in accordance with steps (3) to (6), A. Inspection of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

B. Replacement

(1) Change the brushes in accordance with steps (1) to (5), B. Replacement of Section 3.3.1, Inspection and replacement of the waist drive motor brushes.

3.4 Inspection, Adjustment and Replacement of the Timing Belts

The RV-M2 uses timing belts as a drive-transmitting element as shown in the above indicated construction schematic. Unlike chains and gears, the timing belts require no lubrication and develop low noise. Belt tension, however, is very important in the timing belts. Adjust the belt tension and check and change the belts periodically to ensure long belt life and prevent a noise problem.

3.4.1 Changing intervals

The timing belts used in the RV-M2 must be changed every 2,000 hours of operation under normal condition. More frequent replacement is necessary depending on the operating condition. Use the following guidelines for the replacement of the belts. The timing belts must be changed if:

(1) The root or back of a tooth cracks.

(2) The belt swells due to oil and grease on it.

(3) Teeth wear (up to half of the face width)

(4) A belt tooth misengages with the mating pulley tooth due to excessive wear.

(5) The belt is cut off.

REMARKS

The timing belt wears in the initial run-in period as its rough edges are reduced. Worn rubber particles may be present inside the covers after the robot has been operated for about 300 hours, which does not, however, indicate a faulty condition. In this case, wipe these particles off the inner surfaces of the covers. If rubber particles are soon produced again, change the belt.

3.4.2 Inspection, adjustment and replacement of the wrist pitch drive timing belt

A. Inspection

(1) Fig. 4.3.8 outlines the inspection and adjustment procedures of the wrist pitch drive timing belt.

(2) Remove the pulley cover (L) in accordance with Section 3.2, Removal of Covers.

(3) Visually check the belt to make sure that it does not develop any of the symptoms given in (1) to (5) in Section 3.4.1.

(4) Push the belt at its center with a force of approximately 50 to 90gf and ensure that it deflects about 2mm. See Fig. 4.3.10.

B. Adjustment

(1) Fig. 4.3.8 outlines the inspection and adjustment procedures of the wrist pitch drive timing belt.

(2) Loosen the three motor installation screws ①.

(3) While feeling the tension in the timing belt ②, move the motor ③ in the direction of arrows "a" and/or "b". Note that the motor ③ is secured using the three screws in the slots so that they can be moved in the slots.

(4) Moving the motor in direction "a" tightens the belt; moving it in direction "b" slacks the belt.

(5) Do not move the motor too far in direction "b" or the belt ② comes off the timing pulleys ④ and ⑤.

(6) After the belt tension has been adjusted, tighten the three motor installation screws ① securely. Any loose screw results in the belt slacking off.

C. Replacement

(1) Fig. 4.3.8 outlines the replacement procedure of the timing belt.

(2) Secure the pulleys with care during belt replacement. If the pulleys ④ and ⑤ are misaligned, position misalignment will occur.

(3) Using a marker, mark alignment pointers on the timing belt ② and timing pulleys ④, ⑤, while ensuring that the belt teeth are in correct mesh with the pulley teeth. See Fig. 4.3.8.

(4) Loosen the three motor installation screws ① and remove the belt.

(5) Mark the alignment pointers on the new timing belt. During the procedure, keep the belts tightened.

(6) Mount the new belt, getting it around the timing pulleys ④, ⑤. Make sure that the alignment pointers are lined up.

(7) Adjust the belt tension in accordance with steps (3) to (6) in the above adjustment procedure and Section 3.4.5.

(8) After belt replacement, check that no position misalignment exists.

(9) If position misalignment is found, match the origin match marks in accordance with Section 4.3 Origin Attitude Setting, Vol. 2 to correct the position.

CAUTION

If the mechanical origin has been misaligned by belt replacement, the position data must be reconsidered.

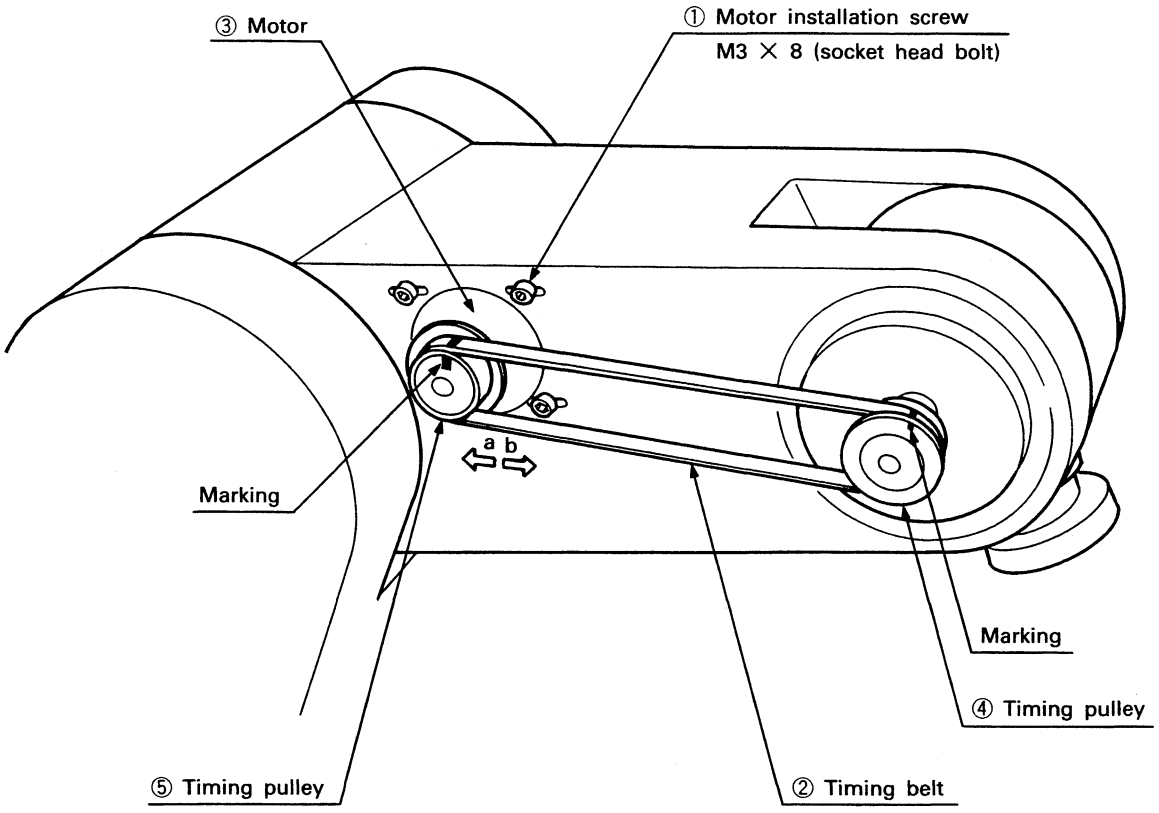


Fig. 4.3.8 Checking, Adjusting and Changing the Wrist Pitch Drive Timing Belt

3.4.3 Inspection, adjustment and replacement of the wrist roll drive timing belt

A. Inspection

(1) Fig. 4.3.9 outlines the inspection and adjustment procedures of the wrist roll drive timing belt.

(2) Remove the pulley cover (R) in accordance with Section 3.2, Removal of Covers.

(3) Visually check the belt to make sure that it does not develop any of the symptoms given in (1) to (5) in Section 3.4.1.

(4) Push the belt at its center with a force of approximately 50 to 90gf and ensure that it deflects about 1.2mm. See Fig. 4.3.10.

B. Adjustment

(1) Fig. 4.3.9 outlines the inspection and adjustment procedures of the wrist roll drive timing belt.

(2) Adjust in accordance with steps (2) to (6) in B. Adjustment of Section 3.4.2 "Inspection, adjustment and replacement of the wrist pitch drive timing belt."

C. Replacement

(1) Fig. 4.3.9 outlines the replacement procedure of the timing belt.

(2) Change in accordance with steps (2) to (9) in C. Replacement of Section 3.4.2 "Inspection, adjustment and replacement of the wrist pitch drive timing belt."

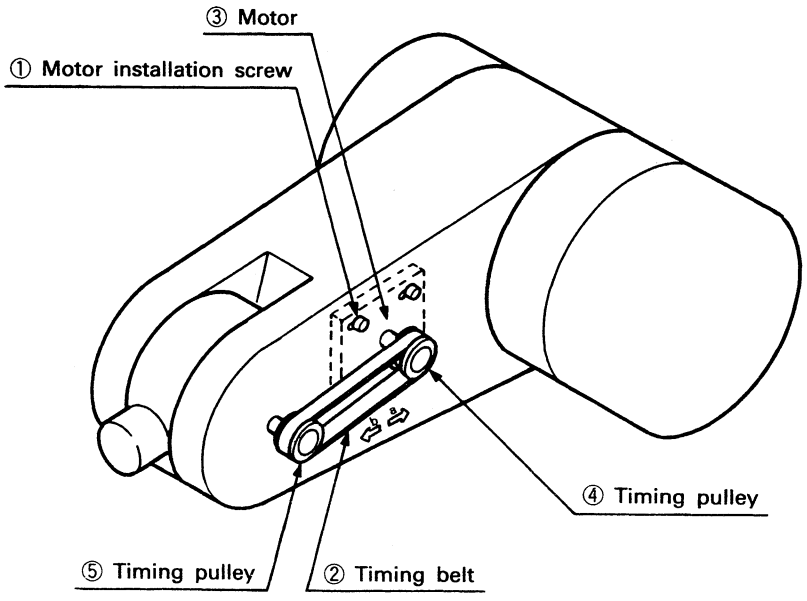


Fig. 4.3.9 Checking, Adjusting and Changing the Wrist Roll Drive Timing Belt

CAUTION

If the mechanical origin has been misaligned by belt replacement, the position data must be reconsidered.

3.4.4 Timing belt tension

Giving an adequate tension to the timing belt is prerequisite for proper drive transmission and sufficient durability. Do not keep the belt too tight or slack: give it appropriate tension so that you feed reaction when pressing it with your thumb. A belt left too slack causes the slack end of the belt to vibrate; a tight belt develops keen noise and vibrates on the tightened end.
Adjust the belt tension in accordance with Fig. 4.3.10 and Table 4.3.3 so that deflection ℓ is achieved when load Pk is applied to the belt.

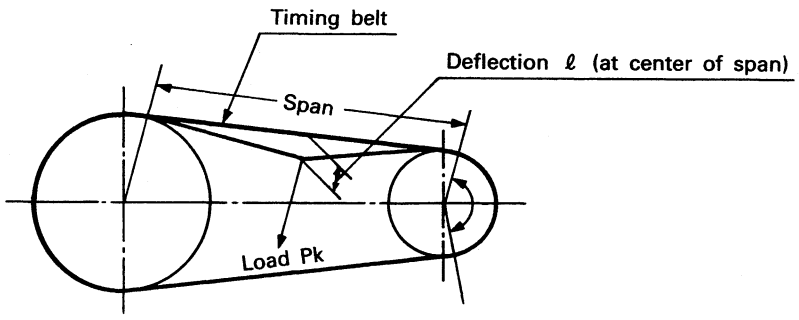


Fig. 4.3.10 Belt Deflection and Load

	Deflection ℓ	Load Pk
Wrist pitch	2.0mm	53 to 87gf
Wrist roll	1.2mm	53 to 87gf

Table 4.3.3 Belt Deflection and Load

3.5 Supplying Grease

Since the wrist roll (J5 axis) of the RV-M2 is driven via the bevel gears as shown in the construction schematic above shown, the bevel gears must be supplied with grease periodically.

3.5.1 Supplying grease to the wrist roll drive gears

The guideline of greasing intervals is 500 hours of operation. However, this greatly depends on the robot operating conditions and environments. Check the remainder of grease on the gear teeth periodically and supply grease as appropriate in the following procedure:

(1) Fig. 4.3.11 gives the way of lubricating the wrist roll drive gears.

(2) Loosen the four cover installation screws ① and remove the cover ②.

(3) Put an appropriate amount of grease (Molyton grease No 2) on your finger or brush and directly apply to the bevel gears ③, ④ through the grease supply window.

(4) Reinstall the cover ② and securely tighten the installation screws to the cover.

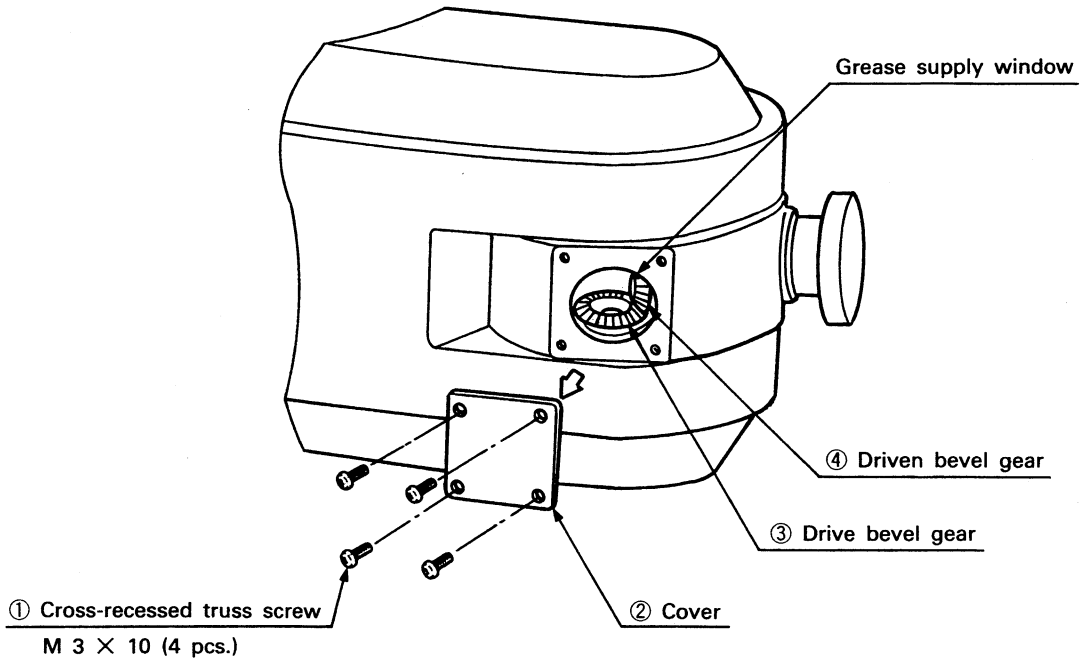


Fig. 4.3.11 Supplying Grease to the Wrist Roll (J5 Axis) Drive Gears

3.6 Changing the Backup Battery

The memory backup battery in the drive unit must be changed periodically (about every two years) in the following procedure. If the battery is not changed over a long period, the battery fault error will occur at power on (LED5 in the drive unit side door is lit).

1) More than one minute after the drive unit is powered down, pull out the first card (indicated #1a) from the left in the rear panel by loosening the upper and lower fixing screws in the card. (See Fig. 4.3.12.)

2) Remove the old battery from the bottom right of the removed card. (After cutting the Insulock Tie (Teflon band) with nippers, etc., pick up both ends of the socket inserted in the card connector and remove the cables.)

3) Secure a new battery in the empty battery area using the furnished Insulock Tie.

4) Pick up both ends of the battery cable socket and insert the socket into the connector on the card removed. At this time, make sure that the electrode side of the socket faces down (the card surface) to ensure correct polarity. (See Fig. 4.3.13.)

5) Then securely insert the card to the first slot from the left and secure it with the upper and lower fixing screws.

CAUTION

- When the power is first switched on after the battery replacement, set bit 2 of the drive unit side door switch SW1 to the lower (OFF) position. After the power is on, set bit 2 to the upper (ON) position.
- The contents of memory may be cleared if a battery fault error occurs. To protect important program and position data, it is suggested to store data into EPROM in advance.

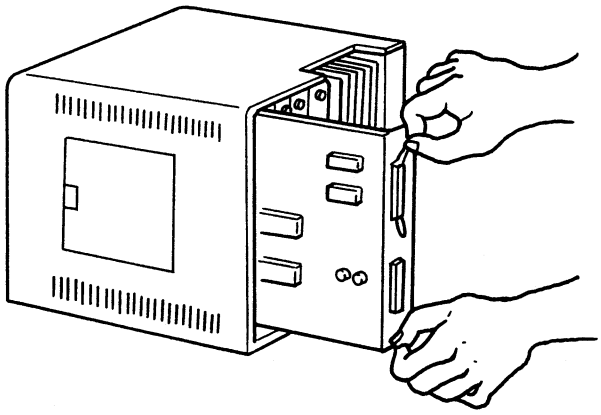


Fig. 4.3.12 Changing the Backup Battery (Pulling out the card)

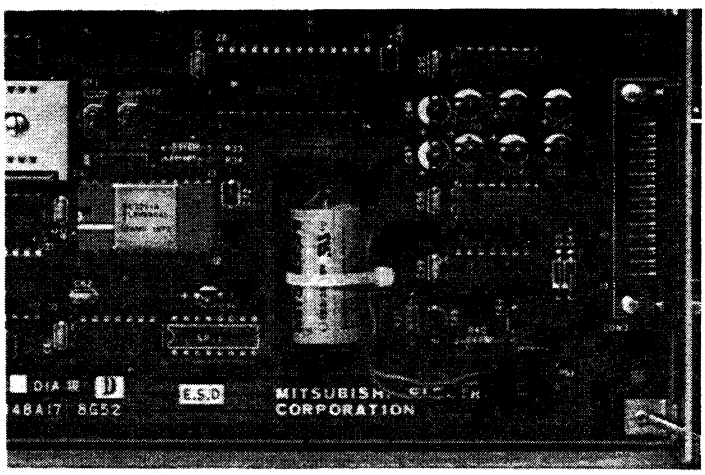


Fig. 4.3.13 Installing the Backup Battery

3.7 Servo Error Checking Operation

The cause of an error (error mode I) attributable to the robot servo system can be checked from the teaching box connected to the drive unit. If the servo error has occurred without the teaching box being connected to the drive unit, the check can be made by connecting the teaching box later. (When the servo error has occurred, LED1 in the drive unit side door is lit and the corresponding axis indicator LED of LEDs 8 to 13 is also lit. (See Section 2.1.2, Vol. 2.)

Procedure

When the servo error has occurred, set the teaching box ON/OFF switch to ON. The servo error may be attributed to any of the following six causes. The error cause is cleared by reset operation, which therefore should be performed after checking the error cause.

(1) Excessive difference error
Indicates that a difference between the command pulse and feedback pulse for any axis has exceeded the allowed value, e.g. load over the rated weight has been moved at high speed.

J1 to J4 axis:
Hold down the X−/W− key of the teaching box. The 4-digit indicator LED shows the axis where the excessive difference error has occurred. (1: present, 0: absent) The LED digits from right to left correspond to the J1 to J4 axes. This applies also to the other errors.
Example: 0010 (excessive difference error at the J2 axis)

J5 axis:
Hold down the X+/W+ key of the teaching box. The 4-digit indicator LED shows the axis where the excessive difference error has occurred. (1: present, 0: absent) The LED digits from right to left correspond to the J1 to J4 axes. This applies also to the other errors.
Example: 01 (excessive difference error at the J5 axis)

(2) Overspeed error
Indicates that the speed of any axis has exceeded the allowed value, e.g. during fast motion, the hand has been accelerated by external force in the moving direction (i.e. the hand with a load of more than the rated weight has been swung straight downward.)

J1 to J4 axes:
Hold down the Y−/S− key of the teaching box.

J5 axis:
Hold down the Y+/S+ key of the teaching box.

(3) Overcurrent error
Indicates that the instantaneous current of any axis motor has exceeded the allowed value, e.g. the robot has interfered with a surrounding obstacle during operation or attempted to abruptly accelerate or decelerate a load over the rated weight.

J1 to J4 axis:
Hold down the [Z−/E−] key of the teaching box.

J5 axis:
Hold down the [Z+/E+] key of the teaching box.

(4) Overload error
Indicates that the average current time integrating value of any axis motor has exceeded the allowed value, e.g. the robot has repeated operating a given axis only, with a load of greater than the rated weight attached to the hand.

J1 to J4 axis:
Hold down the [P−] key of the teaching box.

J5 axis:
Hold down the [P+] key of the teaching box.

(5) Overvoltage/open cable error
Indicates that the applied voltage of any axis motor has exceeded the allowed value when there is no feedback signal of the encoder of the corresponding axis, e.g. the hand has been pressed against the workpiece or the encoder signal line of the corresponding axis has been broken.

J1 to J4 axis:
Hold down the [R−] key of the teaching box.

J5 axis:
Hold down the [R+] key of the teaching box.

(6) Hardware limit exceeded
Indicates that after origin setting, any axis has exceeded its normal operating range and reached its hardware limit (positive or negative side), e.g. position misalignment due to encoder fault, external noise, etc.

J1 to J4 axis:
Hold down the [OPTION−] key of the teaching box.

J5 axis:
Hold down the [OPTION+] key of the teaching box.

3.8 Checking the Limit Pulse

To achieve correct origin setting, the limit switch position and encoder Z phase pulse of each axis should have been adjusted to an appropriate relationship. This relationship is represented by a limit pulse.
The limit pulse is factory-adjusted to the correct value. After long period of use, however, the limit pulse may change because the limit switch will be loose and deteriorate with age. The limit pulse should be within the specified range to maintain the accuracy. If the limit pulse exceeds its allowed range, operation fault may occur during origin setting or the required position may not be reached after origin setting.
The limit pulse can be checked in the following procedure using the teaching box.

1) Connect the teaching box to the drive unit and switch power on. Then perform origin setting in accordance with Section 4.4 "Origin Setting," Vol. 2 OPERATION.

2) Set the teaching box ON/OFF switch to ON. To monitor the state of the J1 axis, for example, press the keys as follows:

   [NST] [9] [9] [1] [ENT]

   The last of the three digits indicates the axis number. To monitor the state of the J2 axis, press [9] [9] [2] keys.

3) At this time, the teaching box LED indicates the limit pulse value. Check that the limit pulse is within the range indicated in Table 4.3.4.

Axis	Minimum Value	Maximum Value
J1	50	450
J2	50	450
J3	50	450
J4	50	450
J5	50	450

Table 4.3.4 Permissible Limit Pulse Range

Consult our Service Division if the limit pulse is outside the above range.

4. SERVICE PARTS

Table 4.4.1 lists consumable parts of the RV-M2 which must be changed periodically. Spare parts are indicated in Tables 4.4.2 and 4.4.3.
When any part is required, contact our Service Division.

4.1 Robot Consumables

No.	Description	Type	Manufacturer	Place of Use	Qty
1	Motor brush	J1-BRSH	Mitsubishi Electric	Waist	2
2	Motor brush	J2-BRSH		Shoulder	2
3	Motor brush	J3-BRSH		Elbow	2
4	Motor brush	J4-BRSH		Wrist pitch	2
5	Motor brush	J5-BRSH		Wrist roll	2
6	Timing belt	155MXL-6.4	Mitsuboshi Belting	Wrist pitch	1
7	Timing belt	103MXL-6.4		Wrist roll	1
8	Grease	Molyton grease No 2	SUMICO	Wrist roll drive gear	About g

Table 4.4.1 Robot Consumables List

4.2 Robot Spare Parts

No.	Description	Type	Manufacturer	Place of Use	Qty
1	DC servo motor	60W-J1M	Mitsubishi Electric	Base	1
2		B60W-J2M		Shoulder	1
3		B40W-J3M		Elbow	1
4		23W-J4, 5M		Wrist pitch Wrist roll (1 each)	2
5	Harmonic drive reduction gear	BU143C275H01	Mitsubishi Electric	Base	1
6		BU143C276H01		Shoulder	1
7		BU143C277H01		Elbow	1
8		BU144D413H01		Wrist pitch Wrist roll (1 each)	2
9	Bevel gear	BU144D604H01	Mitsubishi Electric	Wrist roll (drive)	1
10		BU144D605H01		Wrist roll (driven)	1
11	Photoelectric switch	PM-T53B	Sunx	Base, shoulder, elbow	6
12	Proximity switch assembly	PS-ASSY-4P	Mitsubishi Electric	Wrist pitch, positive side	1
13		PS-ASSY-4M		Wrist pitch, negative side	1
14		PS-ASSY-5P		Wrist roll, positive side	1
15		PS-ASSY-5M		Wrist roll, negative side	1
16	Printed circuit card (signal)	TU811A048G62	Mitsubishi Electric	Base	1
17	Printed circuit card (power)	TU811A049G61		Base	1
18	Printed circuit card (signal)	TU811A051G61		Upper arm	1
19	Printed circuit card (signal)	TU811A054G61		Fore arm	1
20	Gear head	IG-22-1/242	Sayama Precision	Wrist pitch Wrist roll (1 each)	2

Table 4.4.2 Robot Spare Parts List

4.3 Drive Unit Spare Parts

No.	Description	Type	Manufacturer	Place of Use	Qty
1	Fuse	MF60NR10A-05 (250VAC, 10A)	Toyo	In fuse holder in rear panel of drive unit	1
2	Battery	A6BAT	Mitsubishi Electric	In CPU card in drive unit	1
3	Fan	MMF-08B12-DS-R	Mitsubishi Electric	In panel of drive unit	1

Table 4.4.3 Drive Unit Spare Parts List

1. SPECIFICATIONS
2. OPERATION
3. DESCRIPTION OF THE COMMANDS
4. MAINTENANCE AND INSPECTION

5. APPENDICES

CONTENTS (APPENDICES)

1. INTERFACE WITH THE PERSONAL COMPUTER (CENTRONICS)

1.1 Centronics Connector Pin Assignments

Table 5.1.1 indicates the pin assignments for the Centronics connector.

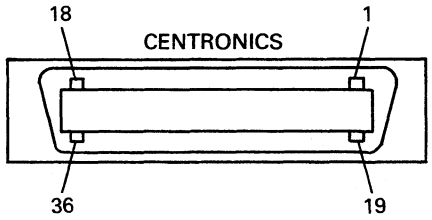


Fig. 5.1.1 Centronics Connector

Pin No.	Signal	Pin No.	Signal
1	$\overline{STB}$	19	GND
2	DB0	20	GND
3	DB1	21	GND
4	DB2	22	GND
5	DB3	23	GND
6	DB4	24	GND
7	DB5	25	GND
8	DB6	26	GND
9	DB7	27	GND
10	$\overline{ACK}$	28	GND
11	BUSY	29	GND
12	GND	30	GND
13	N.C	31	N.C
14	N.C	32	N.C
15	N.C	33	GND
16	GND	34	N.C
17	N.C	35	N.C
18	N.C	36	N.C

Table 5.1.1 Centronics Connector Pin Assignments

N.C: Not connected.

Receptacle type: 57LE-40360-77(D3) ·····DDK
Applicable plug type: 57E-30360 ·····DDK

1.2 Signal Line Functions

Signal	Direction	Function
DB0 to 7 (Data bits)	Input	Represent information of the 8-bit parallel data output from the personal computer.
$\overline{\text{STB}}$ (Strobe)	Input	Pulse signal indicating that input data from the personal computer is present to be read. Switched low to fetch data.
BUSY (Busy)	Output	Switched low to indicate that the drive unit is ready to receive data. Switched high to indicate that the drive unit is unable to receive data.
$\overline{\text{ACK}}$ (Acknowledge)	Output	A low 2 to 3 μ s pulse indicating that the drive unit has received data and is ready to accept other data.
GND (Ground)		Ground for the above signal lines.

Table 5.1.2 Functions of Signal Lines on Centronics Connector

1.3 Signal Line Timing Chart

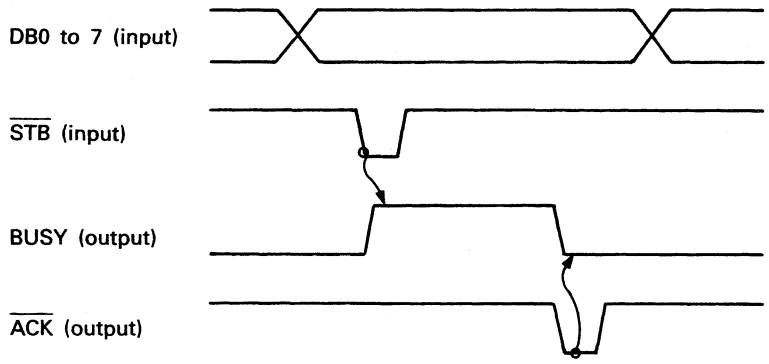


Fig. 5.1.2 Timing Chart

1) Data is transferred from the personal computer to DB0~ DB7.

2) The drive unit then outputs a high BUSY signal on the trailing edge of the $\overline{\text{STB}}$ signal from the personal computer.

3) When the drive unit has completed receiving data and is ready to accept other data, the $\overline{\text{ACK}}$ signal is sent to the personal computer; at the same time, the BUSY signal is switched low.

4) This operating sequence repeats until the terminator, hex. "0D" or hex. "0A", is input.

1.4 Centronics Cable

When connecting your personal computer to the robot through the Centronics interface, the printer cable of your personal computer may be used if it can be properly plugged into the Centronics connector of the drive unit and signal lines match correctly.
Cables for the Mitsubishi MULTI16 and NEC PC9801 are also available as option.

1.5 Centronics Interfacing Examples

Example 1: Using Mitsubishi MULTI16Ⅲ

(1) Connecting the Centronics cable

Using the optional cable for Centronics (type: C-MULTI-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off). For the connector on the MULTI16Ⅲ, use its built-in printer connector.

(2) Setting the personal computer

(Setting the switches)

Set the I/O mode switch (I-SW) on the back panel of the personal computer as follows:

Bit 1: Lower (OFF) position
Bit 2: Upper (ON) position
Bit 3: Lower (OFF) position

After the above switch settings have been made, switch on the personal computer.

(Invoking the software)

1) Activate CP/M86.

2) Activate the GENSYS command and make the following settings:

```
A > GENSYS ↓                    ·····Activates GENSYS.
      ⟆
Printer? Y ↓                    ·····Sets printer configuration.
Type? 24pin 16 ↓                ·····Sets 16-pin type.
      ⟆
END OF GENSYS (Y/N)? Y ↓        ·····Ends GENSYS.
```

3) Activate MBASIC.

```
A > MBASIC                      ·····Activates MBASIC
                                     (or MBASIC2).
```

(3) Setting the drive unit

After the drive unit has been switched on, place ST1 located inside the side door in the lower (OFF) position. This selects the personal computer mode. Make sure that the teaching box ON/OFF switch is set to OFF.

(4) Verifying proper connection

Input the following command in MBASIC:

```
LPRINT "NT" ↓
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

Example 2: Using NEC PC9801F

(1) Connecting the Centronics cable

Using the optional cable for Centronics (type: C-PC-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off).

(2) Setting the personal computer

No special settings are required on the personal computer. After activating N88DISK-BASIC, switch on the drive unit.

(3) Setting the drive unit

Perform the same procedure as in MULTI16Ⅲ.

(4) Verifying proper connection

Perform the same procedure as in MULTI16Ⅲ.

Example 3: Using Mitsubishi MAXY

(1) Connecting the Centronics cable

Using the optional cable for Centronics (type: C-MAXY-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off).
For the connector on the MAXY, use its built-in printer connector.

(2) Setting the personal computer

(Invoking the software)

If the AX Japanese printer drive is designated as an optional driver (DEVICE=PRNDRV. SYS written) in the CONFIG. SYS file (file defining the system configuration), cancel that designation.
The following example gives the operation of correcting CONFIG. SYS using EDLIN (MS-DOS line editor):

1) Activate MS-DOS.

2) Activate EDLIN and proceed as follows:

```
A: > EDLIN CONFIG. SYS ↓          ·····Activates EDLIN.
WHOLE FILE READ.
*L ↓                              ·····Displays all lines.
  1:*BUFFERS=10
  2:  FILES=10
  3:  DEVICE=ANSIJ. SYS
  4:  DEVICE=VJEB. DRV /M1 /HB /G
  5:  DEVICE=PRNDRV. SYS
  6:
*5D ↓  ·····Checks the line where DEVICE=PRNDRV. SYS is written
             and deletes that line.
             Since it exists on line 5 in this example, "5D ↓ " is entered to
             delete line 5.
*E ↓   ·····Ends EDLIN.
A: >
```

3) Reactive MS-DOS.

4) Reactivate MBASIC4.

```
A: > MBASIC4 ↓
```

(3) Setting the drive unit

Perform the same procedure as in MULTI16Ⅲ.

(4) Verifying proper connection

Perform the same procedure as in MULTI16Ⅲ.

2. INTERFACE WITH THE PERSONAL COMPUTER (RS232C)

2.1 RS232C Connector Pin Assignments

Table 5.2.1 indicates the pin assignments for the RS232C connector.

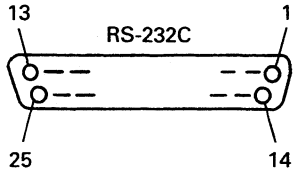


Fig. 5.2.1 RS232C Connector

Pin No.	Signal	Pin No.	Signal
1	FG	14	N.C
2	$\overline{SD}$ ($\overline{TXD}$)	15	N.C
3	$\overline{RD}$ ($\overline{RXD}$)	16	N.C
4	RS (RTS)	17	N.C
5	CS (CTS)	18	N.C
6	DR (DSR)	19	N.C
7	SG	20	ER (DTR)
8	N.C	21	N.C
9	N.C	22	N.C
10	N.C	23	N.C
11	N.C	24	N.C
12	N.C	25	N.C
13	N.C		

Table 5.2.1 RS232C Connector Pin Assignments

N. C: Not connected.

Applicable connector: JAEDB-25P (male) or equivalent (JAE)

2.2 Signal Line Functions

Signal	Direction	Function
FG	—	Frame ground connected to the FG terminal on the drive unit.
$\overline{SD}$ ($\overline{TXD}$)	Output	Provides lines on which the drive unit presents data to the personal computer.
$\overline{RD}$ ($\overline{RXD}$)	Input	Provides lines on which the personal computer presents data to the drive unit.
RS (RTS)	Output	Must be set whenever the personal computer wishes to transmit data.
CS (CTS)	Input	Used to authorize the drive unit to transmit data.
DR (DSR)	Input	Used to indicate that the personal computer is ready to transmit and receive data.
SG	—	Signal ground for data and control lines.
ER (DTR)	Output	Used to indicate that the drive unit is ready to transmit and receive data.

Table 5.2.2 Functions of Signal Lines on RS232C Connector

2.3 RS232C Settings

When the RS232C interface is used, the following communication conditions must be set to the drive unit as well as the personal computer. The settings on the drive unit must be the same as those on the personal computer. Communication cannot be accomplished properly if there is any discrepancy. The following gives drive unit settings. For personal computer settings, see the personal computer instruction manual.

(1) Baud rate setting

The baud rate can be set with DIP switch SW3 located inside the drive unit side door. Place the corresponding bit in the upper (ON) position according to the desired baud rate. (The underlined baud rate in Table 5.2.3 is the standard setting.)

CAUTION

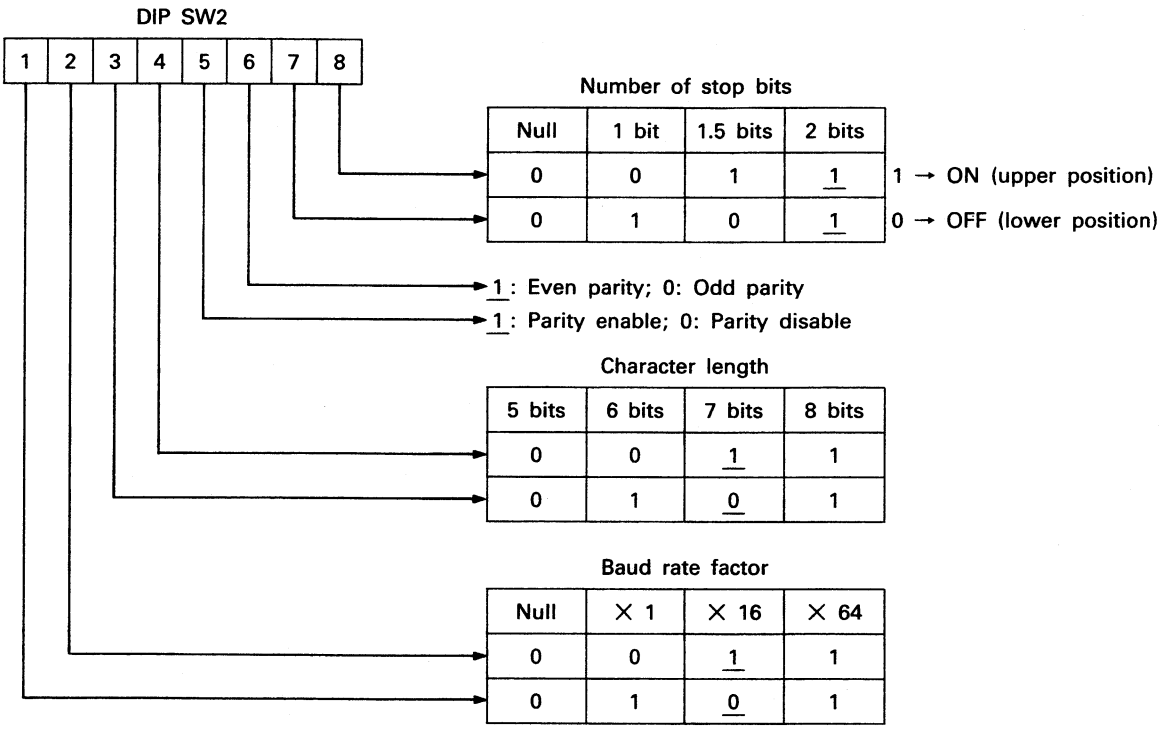
Two or more bits must not be set to ON at one time while the drive unit is on.

DIP SW3 Bit No.	Baud Rate Factor (Unit: BPS)		
	× 1	× 16	× 64
1	1200	75	—
2	2400	150	—
3	4800	300	75
4	9600	600	150
5	—	1200	300
6	—	2400	600
7	—	4800	1200
8	—	9600	2400

Table 5.2.3 Baud Rate Setting

(2) Format setting in asynchronous transmission

The transfer format can be set with DIP switch SW2 located inside the drive unit side door. Set the bits as required according to the desired format. (The underlined settings are standard.)


DIP SW2
1 2 3 4 5 6 7 8
Number of stop bits
Null | 1 bit | 1.5 bits | 2 bits
0 | 0 | 1 | 1
0 | 1 | 0 | 1
1 → ON (upper position)
0 → OFF (lower position)
1: Even parity; 0: Odd parity
1: Parity enable; 0: Parity disable
Character length
5 bits | 6 bits | 7 bits | 8 bits
0 | 0 | 1 | 1
0 | 1 | 0 | 1
Baud rate factor
Null | × 1 | × 16 | × 64
0 | 0 | 1 | 1
0 | 1 | 0 | 1

2.4 Signal Line Timing Chart

The RS232C interface was originally the standard prescribing electrical specifications, shapes of connectors, and pin numbers. As a result, different pieces of communication equipment use different signal line functions and communication procedures. When connecting your personal computer to the robot, therefore, thorough understanding is mandatory of the functions of the signal lines provided for both the personal computer and drive unit. Data transfer between the personal computer and robot is all made in ASCII.

(1) Timing of data transfer from personal computer to robot

〈Robot〉

First character:
Signals ER (DTR) and RS (RTS) are switched high for up to 7ms and 177ms, respectively. If any data is entered during this period, both ER (DTR) and RS (RTS) are switched low allowing the data to be read.

Second character:
The drive unit switches both ER (DTR) and RS (RTS) high to be ready to accept data entry. If any data is input, both ER (DTR) and RS (RTS) are switched low allowing the data to be read. This operation repeats until a hex. "0D" code (CR: carriage return) or a hex. "0A" code (LF: line feed) is input. While any command is being executed by the robot, both ER (DTR) and RS (RTS) are kept low.

〈Personal computer〉

Transfer the first character while DR (DSR) remains high. Transfer the second and subsequent characters when DR (DSR) later goes low and again goes high. The robot will result in an error if the characters are transferred continually when DR (DSR) is high.

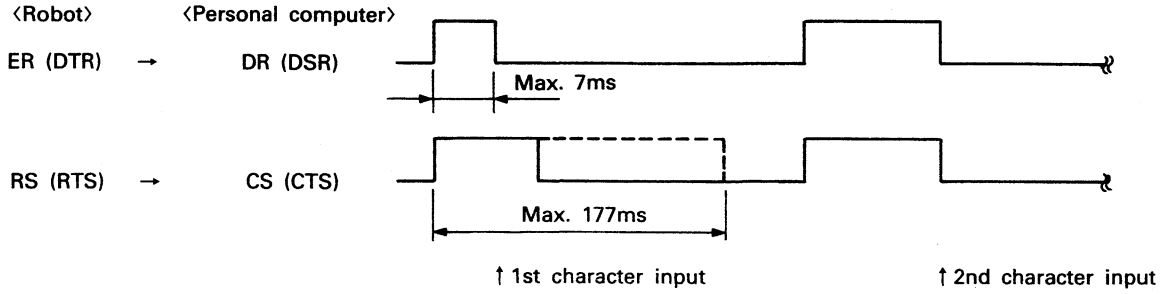


Fig. 5.2.2 Timing of Data Transfer from Personal Computer to Robot

(2) Timing of data transfer from robot to personal computer

〈Robot〉

Data transfer is initiated after ER (DTR) is switched high. ER (DTR) is switched low after the last hex. "0D" code has been transferred.

〈Personal computer〉

The personal computer switches RS (RTS) high and waits for data to be transferred from the robot. If the personal computer requires a hex. "0A" code (LF: line feed) following hex. "0D" as a terminator of received data, bit 1 of SW1 located inside the drive unit side door must be placed in the upper (ON) position.

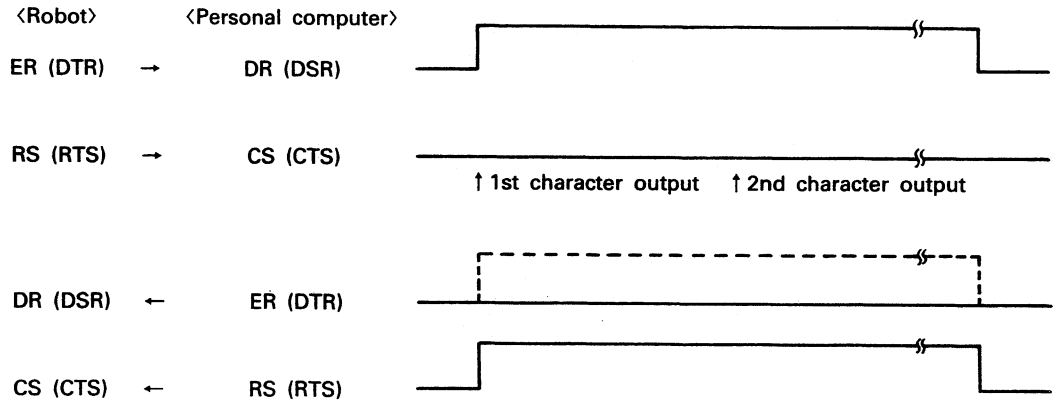


Fig. 5.2.3 Timing of Data Transfer from Robot to Personal Computer

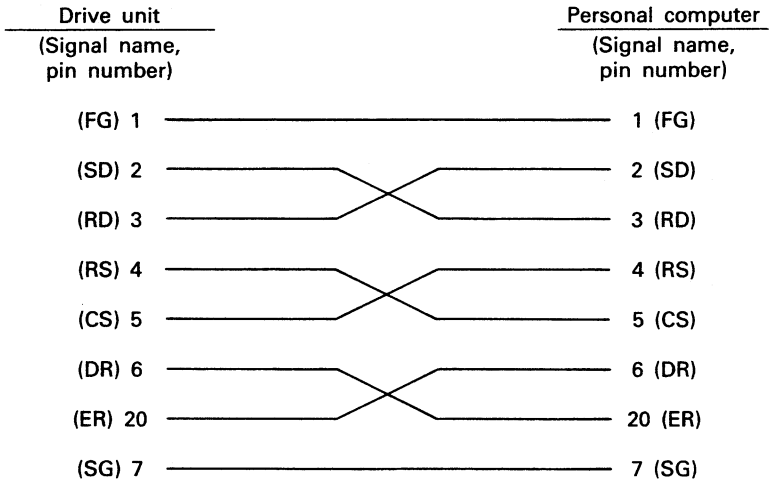
REMARKS

During data transmission from the personal computer to the robot, some personal computers may ignore ON/OFF in the DR or CS signal line and transmit data continuously, causing an error in the robot. In this case, make provisions in the personal computer, e.g. provide a timer for character transmission.

2.5 RS232C Cable

When connecting your personal computer to the robot through the RS232C interface, the RS232C cable of your personal computer may be used if it can be properly plugged into the corresponding connector of the drive unit and signal lines are connected as shown in Fig. 5.2.4. Each signal line must, however, meet the timing described earlier.
Cables for the Mitsubishi MULTI16 and MAXY and NEC PC9801 are also available as option.

(1) MULTI16, PC9801



(2) MAXY

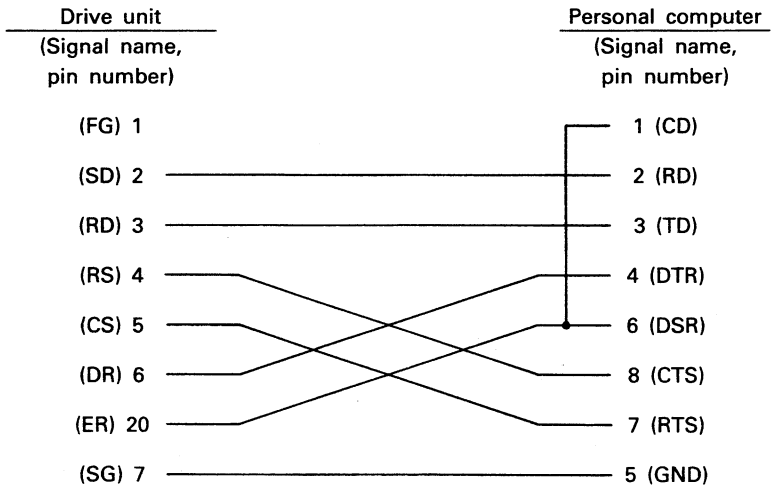


Fig. 5.2.4 RS232C Cable Connection

2.6 RS232C Interfacing Examples

Example 1: Using Mitsubishi MULTI16Ⅲ

(1) Connecting the RS232C cable

Using the optional cable for RS232C (type: RS-MULTI-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off). For the connector on the MULTI16Ⅲ, use its built-in RS232C connector.

(2) Setting the personal computer

(Setting the switches)

Set S3 inside the main cover and I/O mode switch (I-SW) on the back panel as follows:

S3: Flip down to C2 and C4 positions.
I-SW: Place bit 4 in the lower (OFF) position.

After the above switch settings have been made, switch on the personal computer. (For location of S3, see MULTI16Ⅲ Owner's Manual.)

(Invoking the software)

1) Activate CP/M86.

2) Activate the GENSYS command and make the following settings:

```
A > GENSYS ↓                        ·····Activates GENSYS.
      ⟆
RS-232C? Y ↓                        ·····Sets RS232C configuration.
Async/Sync? ASYNC ↓
      ⟆
END OF GENSYS (Y/N)? Y ↓            ·····Ends GENSYS.
```

3) Activate MBASIC.

```
A > MBASIC ↓                        ·····Activates MBASIC
                                         (or MBASIC2).
```

(3) Setting the drive unit

After the drive unit has been switched on, place ST1 located inside the side door in the lower (OFF) position. This selects the personal computer mode. Make sure that the teaching box ON/OFF switch is set to OFF.

Then, set SW1, SW2 and SW3 located in the side door as follows:

SW1: ····· Place bit 1 in the upper position.
SW2, SW3: ····· Make the standard settings according to Section 2.3 "RS232C Settings".

(4) Verifying proper connection

Enter and run the following program in MBASIC:

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT #1, "NT"
30 END
RUN ↓                                  ·····Runs above program.
OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

Example 2: Using NEC PC9801F

(1) Connecting the RS232C cable

Using the optional cable for RS232C (type: RS-PC-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off).

(2) Setting the personal computer

(Setting the switches)

Set DIP switches SW1 and SW2 on the back panel as follows:

SW1: ····· Place bits 7 and 9 in the lower (ON) position. (Internal clock used.)
Place bits 8 and 10 in the upper (OFF) position. (Internal clock used.)
SW2: ····· Place bit 5 in the lower (ON) position. (Not initialized at power on)

After the above switch settings have been made, switch on the personal computer.

(Invoking the software)

1) Activate N88DISK-BASIC.

2) Set the memory switch as follows:

```
OK
mon ↓                                  ·····Activates machine language monitor.
n] ssw2 ↓
08-08 ↓                                ·····Sets baud rate (9600 bauds).
```

3) Press the reset switch for reactivation.

(3) Setting the drive unit

Perform the same procedure as in MULTI16Ⅲ (except for placing bit 1 of SW1 in the lower position).

(4) Verifying proper connection

Enter and run the following program in N-BASIC:

```
10 OPEN "COM1: E73" AS#1
20 PRINT #1, "NT"
30 END

RUN ↓                              ·····Runs above program.
OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

Example 3: A6GPPE (Mitsubishi MELSEC-A series general-purpose programmable controller)

(1) Connecting the RS232C cable

Using the optional cable for RS232C (type: RS-PC-CBL), connect the drive unit to the corresponding A6GPPE connector. (Connect after switching off the drive unit and A6GPPE. Do not ground the A6GPPE.)

(2) Setting the A6GPPE

1) Insert the optional 3.5-inch BASIC disk (SW0GHP-BAS) into drive A of the A6GPPE and power up the A6GPPE.

2) Activate BASIC.

```
A > BASIC ↓
```

(3) Setting the drive unit

Perform the same procedure as in MULTI16Ⅲ (except for placing bit 1 of SW1 in the lower position).

(4) Verifying proper connection

Enter and run the following program in BASIC:

```
10 INIT %1, "6E72"
20 PRINT %1, "NT"
30 END
RUN ↓                              ·····Runs above program.
OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

CAUTION

When using the teaching box, do not connect the drive unit and A6GPPE by the RS232C cable.

Example 4: Using Mitsubishi MAXY

(1) Connecting the RS232C cable

Using the optional cable for RS232C (type: RS-MAXY-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer off).
For the connector on the MAXY, use its built-in RS232C connector.

(2) Setting the personal computer

(Invoking the software)

1) Activate MS-DOS.

2) Activate MBASIC4.

```
A: > MBASIC4 ↓                    ·····Activates MBASIC4.
```

(3) Setting the drive unit

Perform the same procedure as in MULTI16Ⅲ.

(4) Verifying proper connection

Enter and run the following program in MBASIC4:

```
10 OPEN "COM1 : 9600, E, 7, 2" AS #1
20 PRINT #1, "NT"
30 END
RUN ↓                             ·····Runs above program.
OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

Notes on programming with MAXY (using RS232C)

As indicated in the following program example 1, a program that continuously outputs commands from the MAXY to the drive unit may cause an error on the MAXY. (The "DEVICE I/O ERROR" message is displayed.)

Program example 1:

```
10 OPEN "COM1 : 9600, E, 7, 2" AS #1
20 PRINT #1, "NT"
30 PRINT #1, "MO1"          ·····Error occurs if the Movemas-
40 PRINT #1, "MO2"                ter is still returning to origin.
50 END
```

In such a case, insert new lines, as shown in program example 2, that check whether the drive unit is ready to receive data before the next command is output:

Program example 2:

```
10 OPEN "COM1 : 9600, E, 7, 2" AS #1
20 PRINT #1, "NT"
25 WHILE COMST (1, 6) = 0: WEND·····Causes the MAXY to wait
30 PRINT #1, "MO1"                  until the drive unit is ready to
35 WHILE COMST (1, 6) = 0: WEND     receive data.
40 PRINT #1, "MO2"
50 END
```

3. INTERFACE WITH EXTERNAL I/O EQUIPMENT

To connect the robot to external I/O equipment, plug the external I/O cable (to be described later) into the EXTERNAL I/O connector on the I/O card.
The I/O card (type A16) inserted in the drive unit uses the ground common for input and the power supply common for output. The power supply (12 to 24VDC) for the I/O circuits should be prepared by the customer.

Item / Type	General-Purpose I/O		Dedicated I/O		Remarks	ID Marking (Silk-screen printed)
	Input	Output	Input	Output		
A16	16 points $\overline{\text{STB}}$ $\overline{\text{BUSY}}$	16 points $\overline{\text{RDY}}$ ACK	3 points (start, stop, reset)	3 points (run, wait, error)	Operation can be performed using external signals.	2a
B16	16 points STB BUSY	16 points RDY $\overline{\text{ACK}}$	3 points (start, stop, reset)	3 points (run, wait, error)	Operation by means of external signals is possible.	2d

Table 5.3.1 I/O Card Specifications

NOTICE

Differences between "signal" and "$\overline{\text{signal}}$"

〈Input signal〉

"Signal": Indicates that the corresponding terminal is in a significant condition, i.e. the signal is being input to the terminal when a relatively "high voltage" is applied to the terminal.

"$\overline{\text{Signal}}$": Indicates that the corresponding terminal is in a significant condition, i.e. the signal is being input to the terminal when a relatively "low voltage" is applied to the terminal.

〈Output signal〉

"Signal": Indicates that the corresponding terminal is switched to a "high voltage" state when a significant condition is output from the internal circuit to the terminal.

"$\overline{\text{Signal}}$": Indicates that the corresponding terminal is switched to a "low voltage" state when a significant condition is output from the internal circuit to the terminal.

3.1.A External I/O Connector Pin Assignments (Type A I/O Card)

Table 5.3.2.A indicates the pin assignments for the external I/O connector on type A I/O card. "Wire colors" in the Table are keyed to those shown in Section 3.6 "External I/O Cable".

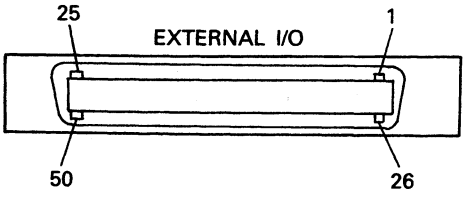


Fig. 5.3.1.A External I/O Connector

Pin No.	Signal	Wire Color	Pin No.	Signal	Wire Color
1	Output port power input	White/black A	26	Output port GND output	White/black B
2	Output port power input	Yellow/black A	27	Output port GND output	Yellow/black B
3	$\overline{\text{Output bit 0}}$	Blue/black A	28	$\overline{\text{Output bit 1}}$	Blue/black B
4	$\overline{\text{Output bit 2}}$	Green/black A	29	$\overline{\text{Output bit 3}}$	Green/black B
5	$\overline{\text{Output bit 4}}$	Orange/black A	30	$\overline{\text{Output bit 5}}$	Orange/black B
6	$\overline{\text{Output bit 6}}$	Pink/black A	31	$\overline{\text{Output bit 7}}$	Pink/black B
7	$\overline{\text{RDY}}$ output	Gray/black A	32	ACK input	Gray/black B
8	$\overline{\text{Output bit 8}}$	Red/black A	33	$\overline{\text{Output bit 9}}$	Red/black B
9	$\overline{\text{Output bit 10}}$	Purple/black A	34	$\overline{\text{Output bit 11}}$	Purple/black B
10	$\overline{\text{Output bit 12}}$	Brown/black A	35	$\overline{\text{Output bit 13}}$	Brown/black B
11	$\overline{\text{Output bit 14}}$	White/black C	36	$\overline{\text{Output bit 15}}$	White/black D
12	$\overline{\text{WAIT}}$ output	Yellow/black C	37	$\overline{\text{RUN}}$ output	Yellow/black D
13	$\overline{\text{ERROR}}$ output	Blue/black C	38	$\overline{\text{START}}$ input	Blue/black D
14	$\overline{\text{STOP}}$ input	Green/black C	39	$\overline{\text{RESET}}$ input	Green/black D
15	$\overline{\text{Input bit 15}}$	Orange/black C	40	$\overline{\text{Input bit 14}}$	Orange/black D
16	$\overline{\text{Input bit 13}}$	Pink/black C	41	$\overline{\text{Input bit 12}}$	Pink/black D
17	$\overline{\text{Input bit 11}}$	Gray/black C	42	$\overline{\text{Input bit 10}}$	Gray/black D
18	$\overline{\text{Input bit 9}}$	Red/black C	43	$\overline{\text{Input bit 8}}$	Red/black D
19	$\overline{\text{BUSY}}$ output	Purple/black C	44	$\overline{\text{STB}}$ input	Purple/black D
20	$\overline{\text{Input bit 7}}$	Brown/black C	45	$\overline{\text{Input bit 6}}$	Brown/black D
21	$\overline{\text{Input bit 5}}$	White/red A	46	$\overline{\text{Input bit 4}}$	White/red B
22	$\overline{\text{Input bit 3}}$	Yellow/red A	47	$\overline{\text{Input bit 2}}$	Yellow/red B
23	$\overline{\text{Input bit 1}}$	Blue/red A	48	$\overline{\text{Input bit 0}}$	Blue/red B
24	Input port power input	Green/red A	49	Input port GND output	Green/red B
25	Input port power input	Orange/red A	50	Input port GND output	Orange/red B

Table 5.3.2.A External I/O Connector Pin Assignments (Type A16)

Receptacle type: 57LE-40500-77 (D3) ····· (DDK)
Applicable plug type: 57E-30500 ····· (DDK)

3.1.B External I/O Connector Pin Assignments (Type B I/O Card)

Table 5.3.2.B shows the pin assignments for the external I/O connector on type B I/O card. "Wire colors" in the Table are keyed to those shown in Section 3.6 "External I/O Cable".

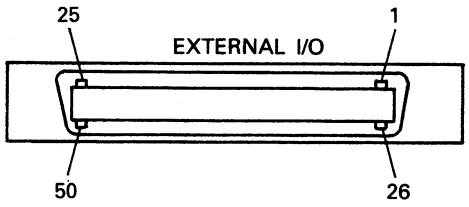


Fig. 5.3.1.B External I/O Connector

Pin No.	Signal	Wire Color	Pin No.	Signal	Wire Color
1	Output port power input	White/black A	26	Output port GND output	White/black B
2	Output port power input	Yellow/black A	27	Output port GND output	Yellow/black B
3	Output bit 0	Blue/black A	28	Output bit 1	Blue/black B
4	Output bit 2	Green/black A	29	Output bit 3	Green/black B
5	Output bit 4	Orange/black A	30	Output bit 5	Orange/black B
6	Output bit 6	Pink/black A	31	Output bit 7	Pink/black B
7	RDY output	Gray/black A	32	$\overline{\text{ACK}}$ input	Gray/black B
8	Output bit 8	Red/black A	33	Output bit 9	Red/black B
9	Output bit 10	Violet/black A	34	Output bit 11	Violet/black B
10	Output bit 12	Brown/black A	35	Output bit 13	Brown/black B
11	Output bit 14	White/black C	36	Output bit 15	White/black D
12	WAIT output	Yellow/black C	37	RUN output	Yellow/black D
13	ERROR output	Blue/black C	38	START input	Blue/black 2
14	STOP input	Green/black C	39	RESET input	Green/black D
15	Input bit 15	Orange/black C	40	Input bit 14	Orange/black D
16	Input bit 13	Pink/black C	41	Input bit 12	Pink/black D
17	Input bit 11	Gray/black C	42	Input bit 10	Gray/black D
18	Input bit 9	Red/black C	43	Input bit 8	Red/black D
19	BUSY output	Violet/black C	44	STB input	Violet/black D
20	Input bit 7	Brown/black C	45	Input bit 6	Brown/black D
21	Input bit 5	White/red A	46	Input bit 4	White/red B
22	Input bit 3	Yellow/red A	47	Input bit 2	Yellow/red B
23	Input bit 1	Blue/red A	48	Input bit 0	Blue/red B
24	Input port power input	Green/red A	49	Input port GND output	Green/red B
25	Input port power input	Orange/red A	50	Input port GND output	Orange/red B

Table 5.3.2.B External I/O Connector Pin Assignments (Type B16)

Receptacle model: 57LE-40500-77OO (D3) ····· (DDK)
Applicable plug model: 57E-30500 ····· (DDK)

3.2.A I/O Circuit Specifications (Type A I/O Card)

The I/O circuits are all isolated by photocouplers. The input port block shown below is separated from the output port block. Each block may be provided with an external power supply, or a single source be used for the two circuits.

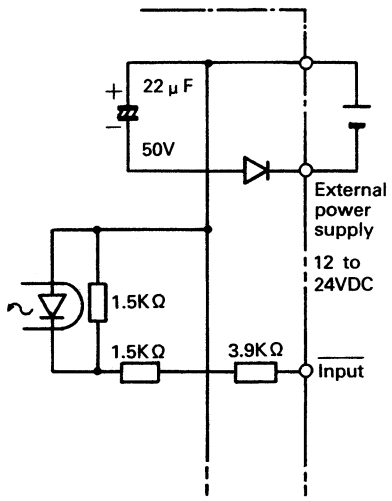
	Signal	Specifications	Internal Circuit
Input port block	Input port power input Input port GND output	Regulated power supply 12 to 24VDC	
	$\overline{\text{Input bits}}$ 0 to 15	Max. ON voltage: 2.0V	
	$\overline{\text{STB}}$ input	Max. ON current: 5.0mA (24VDC)	
	$\overline{\text{START}}$ input $\overline{\text{STOP}}$ input $\overline{\text{RESET}}$ input	Max. leakage current: 200 μA (OFF)	
	$\overline{\text{BUSY}}$ output	Same as $\overline{\text{output bits}}$.	Same as $\overline{\text{output bits}}$.

Table 5.3.3.A Input Port Block Specifications

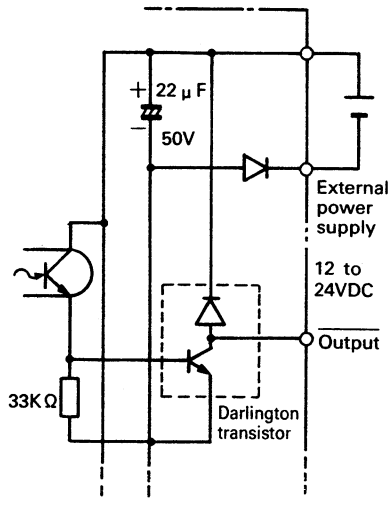
	Signal	Specifications	Internal Circuit
Output port block	Output port power input Output port GND output	Regulated power supply 12 to 24VDC	
	$\overline{\text{Output bits}}$ 0 to 15	Max. applied voltage: Less than above line voltage	
	$\overline{\text{RDY}}$ input	Max. load current: 0.1A/pin	
	$\overline{\text{RUN}}$ output $\overline{\text{WAIT}}$ output $\overline{\text{ERROR}}$ output	Max. ON voltage: 1.3V Max. leakage current: 100 μA	
	ACK input	Same as $\overline{\text{input bits}}$.	Same as $\overline{\text{input bits}}$.

Table 5.3.4.A Output Port Block Specifications

3.2.B I/O Circuit Specifications (Type B I/O Card)

The I/O circuits are all insulated by photocouplers. The input port block shown below are separated from the output port block. Each block may be provided with an external power supply, or a single source be used for the two circuits.

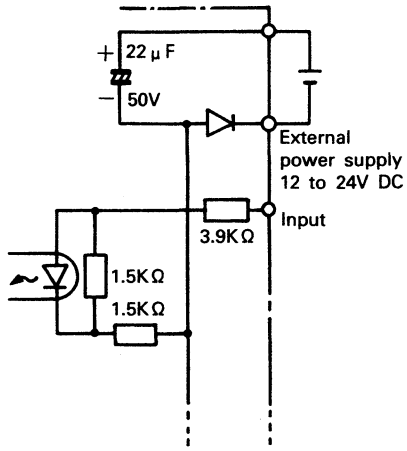
	Signal	Specifications	Internal Circuit
Input port block	Input port power input Input port GND output	Regulated power supply 12 to 24V DC	
	Input bits 0 to 15	Max. ON voltage: 2.0V	
	STB input	Max. ON current: 5.0mA (24VDC)	
	START input STOP input RESET input	Max. leakage current: 200 μA (OFF)	
	BUSY output	Same as output bits	Same as output bits

Table 5.3.3.B Input Port Block Specifications

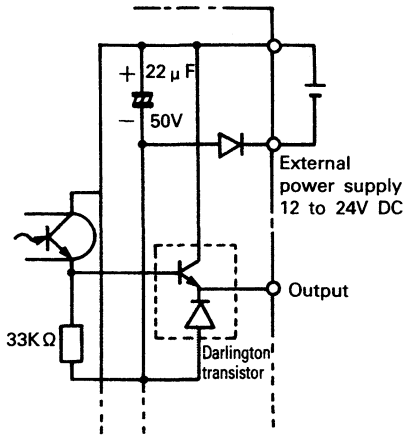
	Signal	Specifications	Internal Circuit
Output port block	Output port power input Output port GND output	Regulated power supply 12 to 24V DC	
	Output bits 0 to 15	Max. applied voltage: Less than above line voltage	
	RDY input	Max. load current: 0.1A/pin	
	RUN output WAIT output ERROR output	Max. ON voltage: 1.3V Max. Leakage current: 100 μA	
	$\overline{\text{ACK}}$ input	Same as input bits	Same as input bits

Table 5.3.4.B Output Port Block Specifications

3.3.A Functions of I/O Signal Lines (Type A I/O Card)

The following Tables show the function of each signal line.

Signal	Function
$\overline{\text{Input bits}}$ 0 to 15	Representing the parallel general-purpose input bits, these signals allow the input state to be read in parallel or on a bit-by-bit basis with a command. Used for conditional jump or interrupt by means of external signals.
$\overline{\text{STB}}$ input	Used to clock parallel data for input. Data from peripheral is read on the trailing edge of the $\overline{\text{STB}}$ signal.
$\overline{\text{BUSY}}$ output	Used to clock parallel data for input. Do not change input data from peripheral while the $\overline{\text{BUSY}}$ signal is being output.
$\overline{\text{START}}$ input	Has the same function as the start switch on the drive unit front panel when the external operation is set.
$\overline{\text{STOP}}$ input	Has the same function as the stop switch on the drive unit front panel when the external operation is set.
$\overline{\text{RESET}}$ input	Has the same function as the reset switch on the drive unit front panel when the external operation is set.

Table 5.3.5.A Input Signals and Functions

Signal	Function
$\overline{\text{Output bits}}$ 0 to 15	Representing the parallel general-purpose output bits, these signals allow the output state to be specified in parallel or on a bit-by-bit basis with a command. Used to send signals to peripheral. (Output is retained.)
$\overline{\text{RDY}}$ output	Used to clock parallel data for output. Cause peripheral to read parallel data when this signal is output.
ACK input	Used to clock parallel data for output. The state of output data is retained until this signal is input.
$\overline{\text{RUN}}$ output	Output while the program is being executed by the drive unit.
$\overline{\text{WAIT}}$ output	Output while the program run by the drive unit is being suspended.
$\overline{\text{ERROR}}$ output	Output when an error occurs in the drive unit (error mode I or II).

Table 5.3.6.A Output Signals and Functions

3.3.B Functions of I/O Signal Lines (Type B I/O Card)

The following Tables show the function of each signal line.

Signal	Function
Input bits 0 to 15	Representing the parallel general-purpose input bits, these signals allow the input state to be read in parallel or bit-by-bit with a command; Used for conditional jump or interrupt by means of external signals.
STB input	Used to clock parallel data for input; Data sent from peripheral is read in on the positive edge of the STB signal.
BUSY output	Used to clock parallel data for input; Do not change input data from peripheral while the BUSY signal is being output.
START input	Has the same function as the start switch on the drive unit front control panel while the external I/O.
STOP input	Has the same function as the stop switch on the drive unit front control panel while the external I/O.
RESET input	Has the same function as the reset switch on the drive unit front control panel while the external I/O.

Table 5.3.5.B Input Signals and Functions

Signal	Function
Output bits 0 to 15	Representing the parallel general-purpose output bits, these signals allow the output state to be specified in parallel or bit-by-bit with a command; Used to send signals to peripheral. (Output is retained.)
RDY output	Used to clock parallel data for output; Peripheral reads in parallel data when this signal is output.
$\overline{\text{ACK}}$ input	Used to clock parallel data for output; The state of output data is retained until this signal is input.
RUN output	Output while the program is being executed by the drive unit.
WAIT output	Output while the program execution by the drive unit is being suspended.
ERROR output	Output when an error occurs in the drive unit (error mode Ⅰ or Ⅱ).

Table 5.3.6.B Output Signals and Functions

3.4.A Example of Connection to I/O Circuits (Type A I/O Card)

The following diagram shows an example of connections between the I/O connector and an external peripheral device.

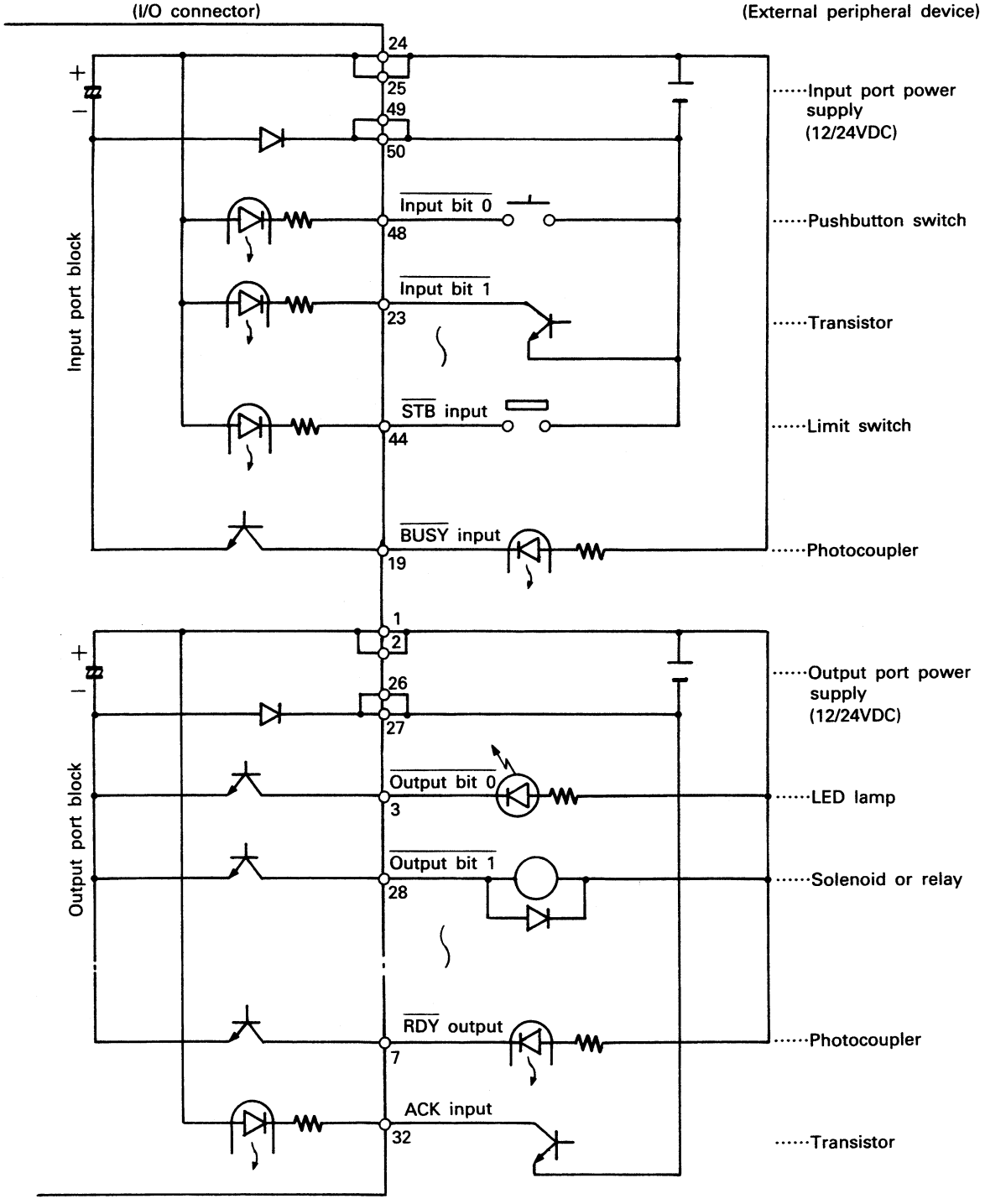


Fig. 5.3.2.A Typical I/O Circuit Connection

3.4.B Example of Connection to I/O Circuits (Type B I/O Card)

The following diagram shows an example of connections between the I/O connector and an external peripheral device.

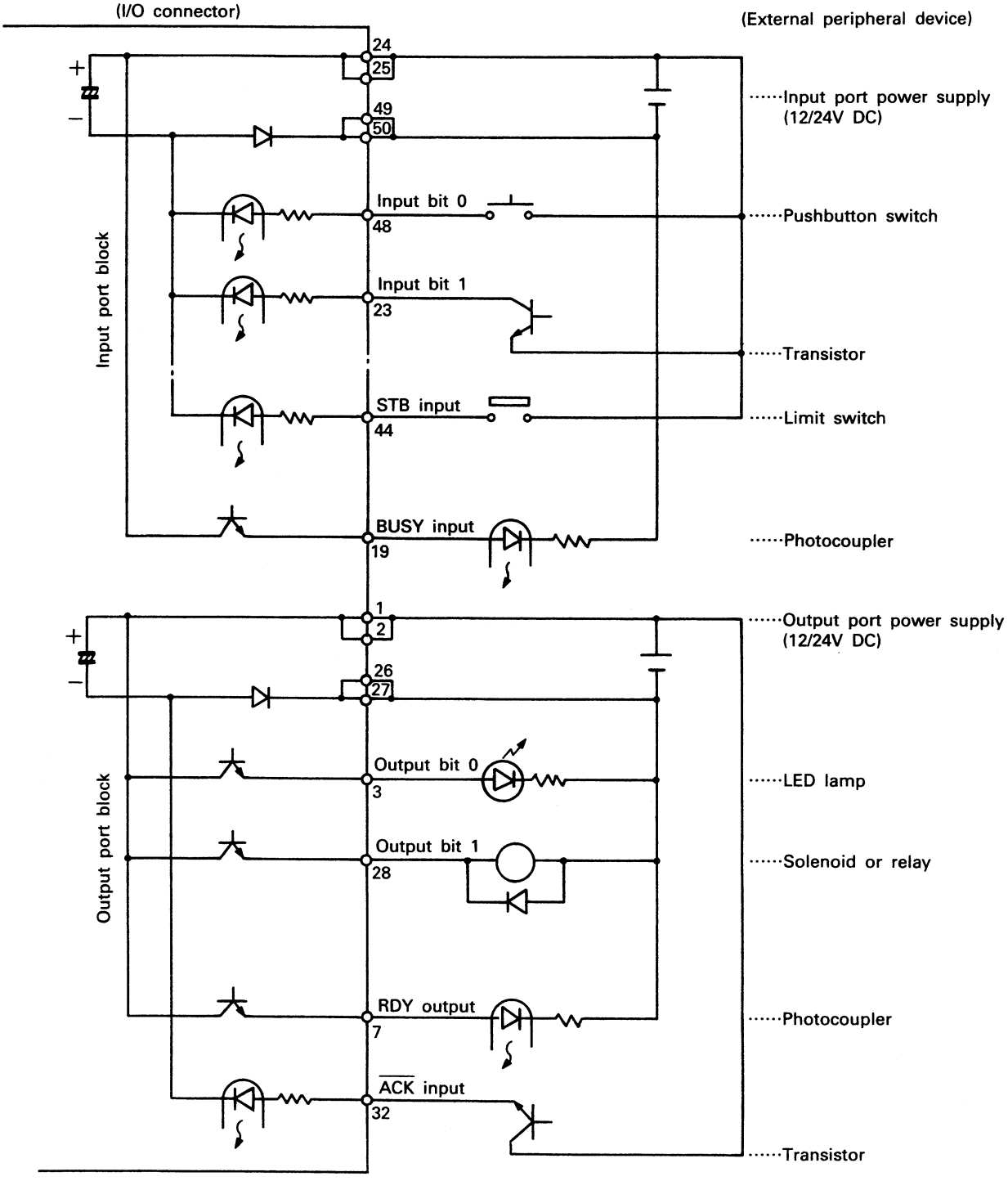


Fig. 5.3.2.B Typical I/O Circuit Connection

The following example indicates correspondence between bits and internal register values (decimal and hexadecimal) for entering data using the input instruction (ID, IN) and outputting data using the output instruction (OD, OT).

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Hexadecimal					Decimal					
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	&	7	F	F	F	+	3	2	7	6	7
0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	0	&	5	B	A	0	+	2	3	4	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	&	0	0	0	1					+	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	&	0	0	0	0						0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	&	F	F	F	F					−	1
1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1	&	C	F	C	7	−	1	2	3	4	5
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	&	8	0	0	0	−	3	2	7	6	8

Most significant bit

Sign bit

Least significant bit

Table 5.3.7 Correspondence between I/O Bits and Internal Register Values

The hand check conformation inputs (equivalent to bits 16 and 17) are fetched together with bits 0 to 15 by the input instruction (ID, IN). (Note, however, that bits 16 and 17 are not fetched as internal register values.) The TB (Test Bit) command may be used to confirm the hand check input in the program and the DR (Data Read) command to read the hand check input to the personal computer.

3.5 I/O Signal Line Timing Chart (Synchronous I/O)

3.5.1.A Synchronous input timing (Type A I/O card)

Fig. 5.3.3.A shows the timing involved in inputting data using the command "IN."

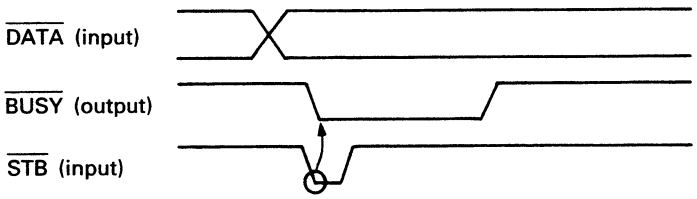


Fig. 5.3.3.A Synchronous Input Timing

(1) The drive unit is ready to receive data from the external device while the BUSY signal remains HIGH.

(2) When the external device inputs data and an STB signal, the drive unit causes the BUSY signal to go LOW and outputs the LOW signal to the external device. The external device should not change the data while the BUSY signal remains LOW.

(3) When the drive unit reads the data in, the BUSY signal goes HIGH allowing other data to be entered.

3.5.1.B Synchronous input timing (Type B I/O card)

Fig. 5.3.3.B shows the timing involved in inputting data using the command "IN."

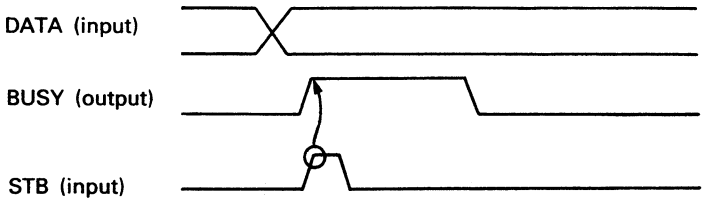


Fig. 5.3.3.B Synchronous Input Timing

(1) The drive unit is ready to receive data from the external device while the BUSY signal remains LOW.

(2) When the external device inputs data and a STB signal, the drive unit causes the BUSY signal to go HIGH and outputs the HIGH signal to the external device. The external device should not change the data while the BUSY signal remains HIGH.

(3) When the drive unit reads the data in, the BUSY signal goes LOW allowing other data to be entered.

3.5.2.A Synchronous output timing (Type A I/O card)

Fig. 5.3.4.A shows the timing involved in outputting data using the command "OT."

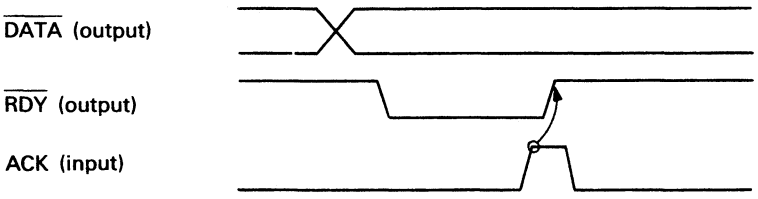


Fig. 5.3.4.A Synchronous Output Timing

(1) The drive unit outputs data as specified by parameters defined in the command "OT" while the RDY signal remains HIGH and the ACK signal remains LOW.

(2) As soon as the data is output, the drive unit causes the RDY signal to go LOW and outputs the LOW signal to the external device.

(3) On receiving an ACK signal input from the external device, the drive unit causes the RDY signal to go HIGH allowing other data to be output.

3.5.2.B Synchronous output timing (Type B I/O card)

Fig. 5.3.4.B shows the timing involved in outputting data using the command "OT."

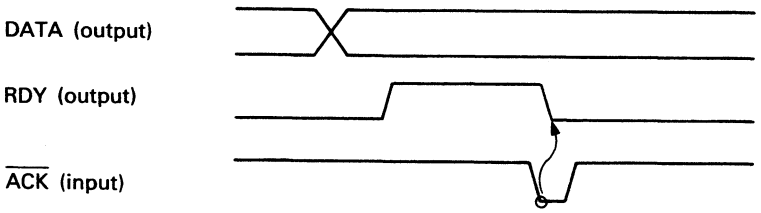


Fig. 5.3.4.B Synchronous Output Timing

(1) The drive unit outputs data as specified by parameters defined in the command "OT" while the RDY signal as well as the ACK signal remains HIGH.

(2) As soon as the data is output, the drive unit causes the RDY signal to go HIGH and outputs the HIGH signal to the external device.

(3) On receiving an ACK signal input from the external device, the drive unit causes the RDY signal to go LOW allowing other data to be output.

3.5.3.A Dedicated I/O timing

In a robotic operation environment on the actual production floor, operation can be performed through the dedicated signal lines of the I/O connector, instead of using the front control switches on the drive unit.
The following describes the necessary settings to be made and signal timing involved in this operation:

(1) Make the following switch setting in accordance with Section 2.1.2 "Functions of side setting switches and LEDs", Vol. 2. OPERATION:
Set bit 5 of SW1 to the upper (ON) position.

(2) When the above setting has been made, the drive unit front control switches are disabled except for the emergency stop and stop switches.

(3) In this setting, the program can be started, stopped, resumed and reset (including resetting of error modes I and II) by the dedicated input signals. The function of each dedicated input signal corresponds to that of the drive unit front control switch as follows:

⟨Front control switches⟩		⟨Dedicated input signals⟩
START switch	⟷	$\overline{\text{START}}$ input signal
STOP switch	⟷	$\overline{\text{STOP}}$ input signal
RESET switch	⟷	$\overline{\text{RESET}}$ input signal

(4) The dedicated outputs provide signals corresponding to the robot operating status ($\overline{\text{RUN}}$, $\overline{\text{WAIT}}$ and $\overline{\text{ERROR}}$ outputs). These signals are output in either of external operation and front control switch operation modes.
Fig. 5.3.5 shows the dedicated I/O timing:

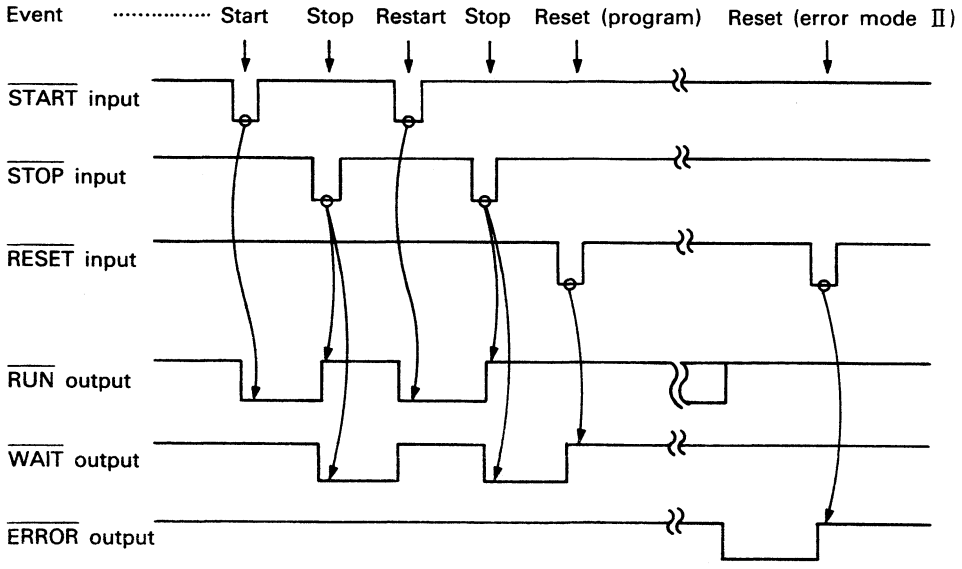


Fig. 5.3.5.A Dedicated I/O Timing

3.5.3.B Dedicated I/O timing

In a robotic operation environment on the actual production floor, operation can be performed through the dedicated signal lines of the I/O connector, instead of using the front control switches on the drive unit.
The following describes the necessary settings to be made and signal timing involved in this operation:

(1) Make the following switch setting in accordance with Section 2.1.2 "Functions of side setting switches and LEDs", Vol. 2. OPERATION:
Set bit 5 of SW1 to the upper (ON) position.

(2) When the above setting has been made, the drive unit front control switches are disabled except for the emergency stop and stop switches.

(3) In this setting, the program can be started, stopped, resumed and reset (including resetting of error modes I and II) by the dedicated input signals. The function of each dedicated input signal corresponds to that of the drive unit front control switch as follows:

⟨Front control switches⟩		⟨Dedicated input signals⟩
START switch	⟷	START input signal
STOP switch	⟷	STOP input signal
RESET switch	⟷	RESET input signal

(4) The dedicated outputs provide signals corresponding to the robot operating status (RUN, WAIT and ERROR outputs). These signals are output in either of external operation and front control switch operation modes.
Fig. 5.3.5 shows the dedicated I/O timing:

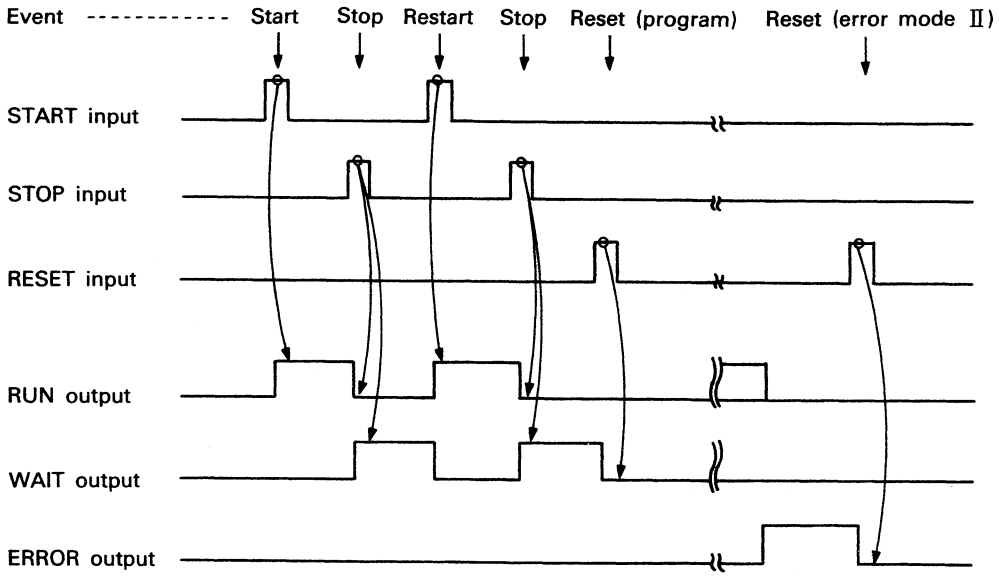


Fig. 5.3.5.B Dedicated I/O Timing

3.6 External I/O Cable

A dedicated cable is available as an option to connect the drive unit to the external peripheral device. One end of the external I/O cable can be plugged into the external I/O connector on the drive unit. The other free end is used to hook it up to the external device. The jacket of each signal line (50 in total) is marked with an identification color and dot marking. Connect the right wire to the right pin in accordance with the color coding in Section 3.1 "External I/O Connector Pin Assignments".

Example:

Pin No.	Signal	Wire Color
21	Input bit 5	White red A

Dot marking type (see below)
Dot marking color
Jacket color

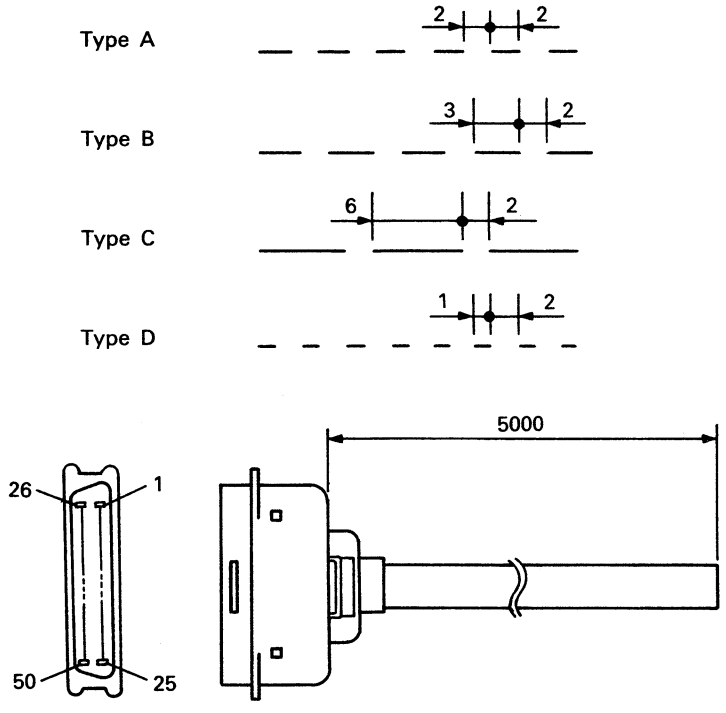


Fig. 5.3.6 External I/O Cable Marking

3.7 Precautions for Connection to External Equipment

(1) Make sure that the external power voltage is within the specified range. Also check that the polarity of the power supply is correct.

(2) The input signal used should be a no-voltage contact signal or a transistor open-collector signal.

(3) When energizing a coil in relays and solenoids, connect a surge suppressor diode in parallel with the load. (Note the correct polarity of the diode. See Fig. 5.3.2.)

(4) When lighting an LED, connect a protective resistor in series corresponding to the rated current.

(5) When lighting an incandescent lamp, a rush current flows about 10 times larger than the rating when the lamp is first energized. To ensure that a current about 20% of the rating flows at all times, a resistor (R) must be connected in series. (See Fig. 5.3.7.)

(6) When connecting the drive unit with any peripheral equipment by the external I/O cable, install the equipment away from the other noise-generating line to prevent I/O error.

(7) The load connected to the output signal must be selected carefully so that the maximum output current is not exceeded. The output signal must not be connected with the power supply (type A) or ground (type B) to protect the internal output transistor.

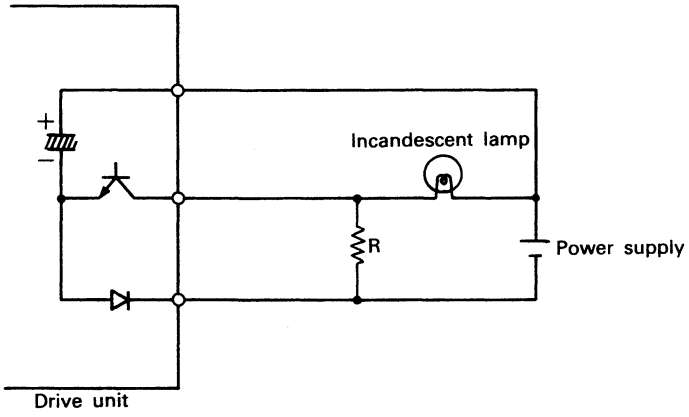


Fig. 5.3.7.A Typical Circuit Including an Incandescent Lamp (Type A I/O Card)

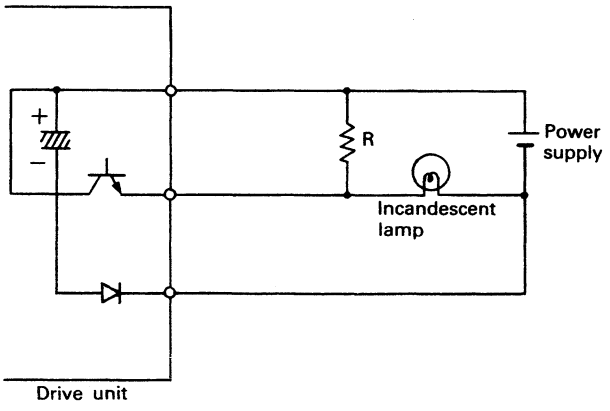


Fig. 5.3.7.B Typical Circuit Including an Incandescent Lamp (Type B I/O Card)

4. HAND CHECK SIGNALS

The Movemaster RV-M2 has two dedicated input points to confirm operation, gripping, etc. of the hand, etc.
The hand check confirmation contacts (to be prepared by the customer) should be selected in accordance with Section 4.2.

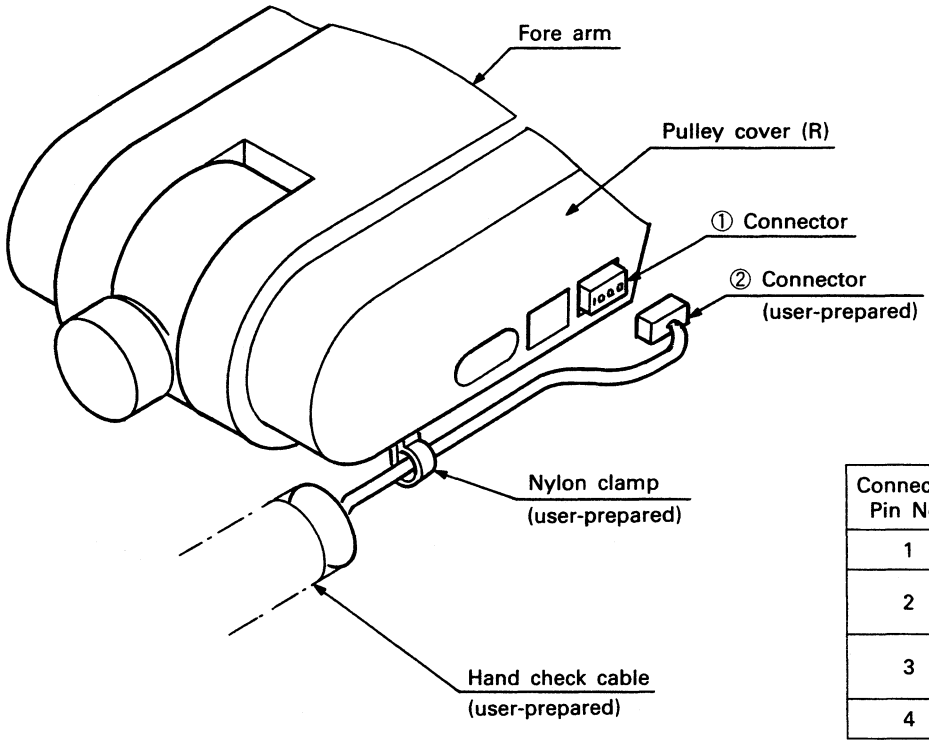
4.1 Connection of Hand Check Signals

(1) Fig. 5.4.1 shows the way of connecting the hand check signals.

(2) Connect a hand check cable (user-prepared) to the connector ① in the pulley cover (R).

(3) Secure the hand check cable to the fore arm with a nylon clamp (user-prepared) using the pulley cover (R) installation screw as shown in Fig. 5.4.1.

(4) When the input instruction (ID, IN) is executed, hand check signals 1 and 2 correspond to bits 16 and 17, respectively.



Connector Pin No.	Signal
1	+5V
2	Hand check 1
3	Hand check 2
4	0V

No.	Description	Type		Manufacturer	Remarks
		Connector	Connector pin		
①	Connector	SMP-04V-B	SHF-001T-0.8SS	Japan Solderless Terminal	
②	Connector	SMR-04V-B	SYM-001T-0.6	Japan Solderless Terminal	

Fig. 5.4.1 Connection of Hand Check Signals

4.2 Hand Check Signal Interface

The hand check confirmation contacts are generally available in mechanical, photoelectric and magnetic types, and their output circuits in sink and source types. The following gives the sink and source type input circuits available in the robot and their typical connections. For the actual interface setting, see Section 4.3.

(1) Sink type input circuit connection example

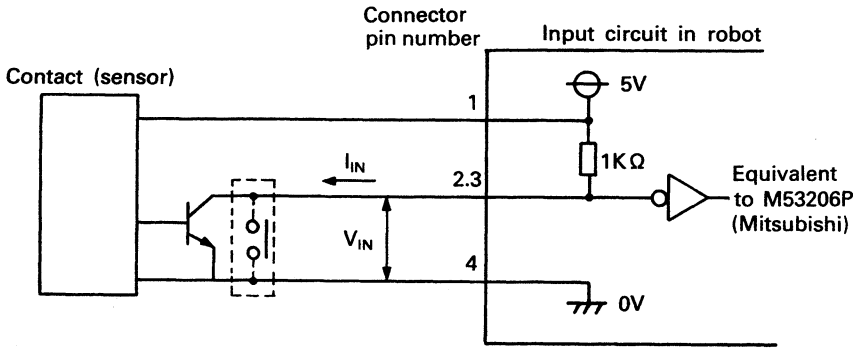


Fig. 5.4.2 Typical Sink Type Input Circuit Connection

Note: When the mechanical contact is used, connect it to the area indicated by the broken line shown in Fig. 5.4.2. At this time, the 5V line (pin No. 1) need not be connected. The contact used must be of microcurrent rating (minimum energizing current: 5V, 5mA or less).

(2) Source type input circuit connection example

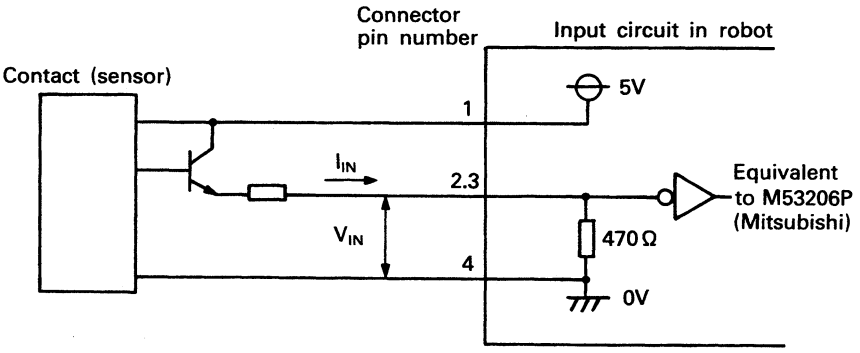


Fig. 5.4.3 Typical Source Type Input Circuit Connection

In the above (1) and (2) examples, the electrical specifications at on/off of the contact must satisfy the following:

		Sink Type	Source Type
At on	V_{IN}	Max. 0.7V	Max. 0.7V
	I_{IN}	Min. 4.3mA	Max. 0.1mA
At off	V_{IN}	Min. 2.2V	Min. 2.2V
	I_{IN}	Max. 2.8mA	Min. 4.7mA

Table 5.4.1 Electrical Specifications of Sink and Source Type Input Circuits

4.3 Interface Setting

(1) Fig. 5.4.4 shows the way of setting the interface.

(2) Remove the cover installation screws ① and remove the base side cover (R) ② .

(3) Pins S11 correspond to hand check signal 1 and S2 to hand check signal 2. (For hand check signals 1 and 2, see Fig. 5.4.1.)

(4) When the sensor is of the sink type, set the interface as indicated by "a" in Fig. 5.4.4. When the sensor is of the source type, set the interface as indicated by "b" in Fig. 5.4.4. Hand check signals 1 and 2 can be set independently. (For the optional pneumatic hand, set the interface to the source type.)

(5) Hand check signals 1 and 2 are both factory-set to the sink type (a).

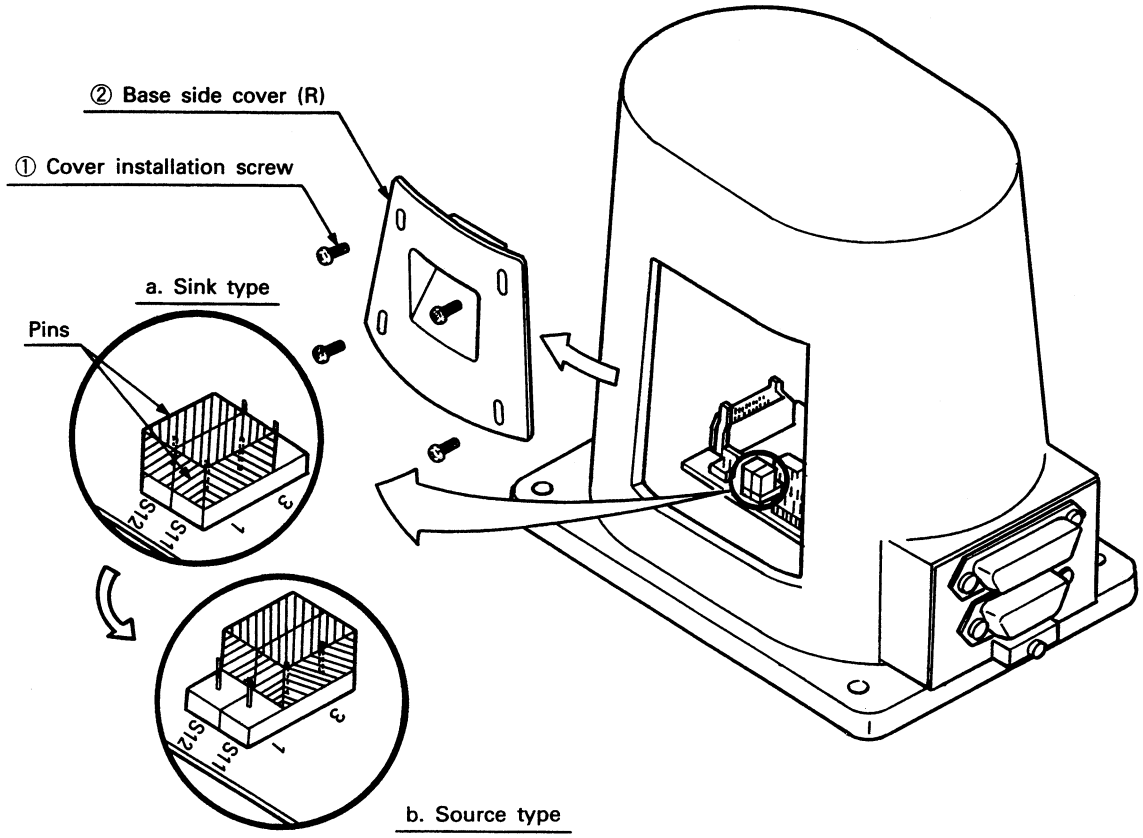


Fig. 5.4.4 Interface Setting

5. PROGRAMMING SYSTEM USING PERSONAL COMPUTER

5.1 Writing and Running the Program

It is wise to create a programming system when a personal computer is used to operate the Movemaster and generate programs. This section introduces a simple BASIC program as an example of the personal computer application. This program assumes the use of the RS232C interface and allows the use of all Movemaster commands. In addition, by making use of the screen editor facility of the personal computer, data shown on the display can be corrected by means of the cursor and keyboard.
Note that some modifications of the program are necessary depending on the model of your personal computer.

```
1000  '******** RV-M2 PROGRAMMING SYSTEM ********
1010  '******** USING MULTI 16 BY MITSUBISHI ELECTRIC
1020  '
1030  OPEN "COM 1 : 9600, E, 7, 2" AS #1·····Depends on the personal computer used.
1040  LINE INPUT ; A$
1050  C$ =LEFT$ (A$, 2)
1060  IF C$="DR" THEN PRINT : GOTO 1170
1070  IF C$="LR" THEN PRINT : GOTO 1240
1080  IF C$="PR" THEN PRINT : GOTO 1360
1090  IF C$="WH" THEN PRINT : GOTO 1600
1100  IF C$="CR" THEN PRINT : GOTO 1780
1110  IF C$="ER" THEN PRINT : GOTO 1910
1115  IF C$="WT" THEN PRINT : GOTO 1980
1117  IF C$="NR" THEN PRINT : GOTO 2050
1120  PRINT  #1, A$ : PRINT
1130  GOTO 1040
1140  '
1150  '
1160  ' ******** RS232C INPUT DATA READ ('DR') ********
1170  PRINT #1, A$
1180  LINE INPUT #1, B$
1190  PRINT "INPUT DATA=" ; B$
1200  GOTO 1040
1210  '
1220  '
1230  ' ******** RS232 C PROGRAM LINE READ ('LR') ********
1240  INPUT "START LINE="; S
1250  INPUT "END  LINE="; E
1260  FOR I=S TO E
1270      PRINT #1, "LR" +STR$ (I)
1280      LINE INPUT #1, A$
1290      IF A$=" "THEN 1310
1300      PRINT I ; : PRINT A$ ····· (Note)
1310  NEXT I
1320  GOTO 1040
1330  '
1340  '
```

When using NEC's 9801, modify the program as follows:

```
1030 OPEN "COM1 : E73" AS #1
  ⟆
1060 IF C$="DR" THEN 1170
1070 IF C$="LR" THEN 1240
1080 IF C$="PR" THEN 1360
1090 IF C$="WH" THEN 1600
1100 IF C$="CR" THEN 1780
1110 IF C$="ER" THEN 1910
1115 IF C$="WT" THEN 1980
1117 IF C$="NR" THEN 2050
1120 PRINT #1, A$
```

For Mitsubishi's A6GPPE, modify the program as follows:

```
1025 INIT %1, "6E72"
1040 LINE INPUT A$
```

In addition, modify the following commands:

```
PRINT #1        → PRINT %1
LINE INPUT #1 → LINE INPUT %1
```

(Note)
If the LPRINT statement is used instead of the PRINT statement, the program data can be output to the printer.

```
1350  ' ******** RS232C POSITION READ ('PR') ********
1360  INPUT "START POSITION=" ; S
1370  INPUT "END  POSITION=" ; E
1380  PRINT "POS.NO." ; SPC (3) ; " X(mm) Y(mm) Z(mm) P(deg) R(deg) H(O/C)" : PRINT
1390  FOS I=S TO E
1400      PRINT #1 "PR" +STR$ (I)
1410      LINE INPUT #1, A$
1420      IF A$=" 0, 0, 0, 0, 0 " THEN 1550
1430      PRINT "PD" ; : PRINT USING "###" ; I ; : PRINT ", " ;
1440      K$=", "
1450      K=1
1460      FOR J=1 TO 5
1470          IF J=5 THEN 1490
1480          A(J)=INSTR (K, A$, ", ") : GOTO 1500
1490          A(J)=LEN (A$) +1 : K$=" "
1500          V(J)=VAL (MID$ (A$, K, A(J) - 1))
1510          K=A(J) + 1
1520          PRINT USING "#####.#" ; V(J) ; : PRINT K$ ;
1530      NEXT J
1540      PRINT ", " ; RIGHT$ (A$, 1)
1550  NEXT I
1560  GOTO 1040
1570  '
1580  '
1590  ' ******** RS232C CURRENT POSITION READ ('WH')********
1600  PRINT #1, A$
1610  LINE INPUT #1, B$
1620  PRINT SPC (14) ; " X(mm) Y(mm) Z(mm) P(deg) R(deg) H(O/C)" : PRINT
1630  K=1
1640  PRINT "CUR. POS. ";
1650  FOR I=1 TO 5
1660      IF I=5 THEN 1680
1670      A(I) =INSTR (K, B$, ", ") : GOTO 1690
1680      A(I) =LEN (B$) +1
1690      V(I) =VAL (MID$ (B$, K, A(I) -1))
1700      K=A(I) +1
1710      PRINT USING "#####.#" ; V(I) ; : PRINT " " ;
1720  NEXT I
1730  PRINT RIGHT$ (B$, 1)
1740  GOTO 1040
1750  '
1760  '
1770  ' ******** RS232C COUNTER READ ('CR')********
1780  INPUT "START COUNTER=" ; S
1790  INPUT "END  COUNTER=" ; E
1800  PRINT : PRINT "CNT.NO. " : PRINT
1810  FOR I=S TO E
1820      PRINT #1, "CR" +STR$ (I)
1830      LINE INPUT #1, A$
```

```
1840      IF A$="0" THEN 1860
1850      PRINT "SC" ; :PRINT USING " ##" ; I ; : PRINT "," ; : PRINT USING " ######" ; VAL (A$)
1860  NEXT I
1870  GOTO 1040
1880  '
1890  '
1900  ' ******** RS232C ERROR READ ('ER') ********
1910  PRINT #1, A$
1920  LINE INPUT #1, B$
1930  PRINT "ERROR MODE=" ; B$
1940  GOTO 1040
1950  '
1960  '
1970  ' ******** RS232C TOOL LENGTH READ ('WT') ********
1980  PRINT #1, A$
1990  LINE INPUT #1, B$
2000  PRINT "TOOL=" ; B$
2010  GOTO 1040
2020  '
2030  '
2040  ' ******** RS232C NEST READ ('NR') ********
2050  INPUT "START JOINT=" ; S
2060  INPUT "END   JOINT=" ; E
2070  PRINT : PRINT "JT. LIMIT PULSE " : PRINT
2080  FOR I=S TO E
2090      PRINT #1, " NR" + STR$ (I)
2100      LINE INPUT #1, A$
2110      PRINT " " ; : PRINT USING "##" ; I ; : PRINT USING "########"; VAL (A$)
2120  NEXT I
2130  GOTO 1040
```

〈Explanation〉

Start:

After the program has been written, run the program. The cursor lights up, indicating that the system is ready to accept key entry.

ex. RUN ↓

□

Direct execution:

Directly key in the intelligent commands. The robot executes the commands entered.

ex. NT ↓ ····· Executes return-to-origin operation.

For any of the RS232C read instructions containing a parameter, enter the first two letters only and define the parameter using the BASIC input statements (commands LR, PR, CR and NR).

Program generation:

Directly key in the intelligent commands following line numbers (1 to 3584). This will generate the program for the specified line number.

ex. 10 MO 1, C ····· Moves to position 1 with hand closed.

Note that part of the commands cannot be programmed (those marked ※ in DESCRIPTION OF THE COMMANDS).

Program execution:

The generated program can be executed starting with the specified line number by keying in the start command "RN".

ex. RN 5↓ ····· Executes the program starting with line 5.

Error mode II results if a wrong command has been entered, a wrong parameter has been defined, or an unexecutable command has been entered. In this case, key in the reset command (RS) or press the reset switch on the drive unit front panel. (When the teaching box is set to ON, the error may be reset by its reset key.)

〈Applications〉

```
RUN↓
NT↓                              ····· Direct execution example
MO 1↓
GO↓
MO 2↓
DW 10, 20, 30↓

10 MO 1,C↓
20 MO 2,O↓                       ····· Program generation example
30 GC↓
40 DW 10, --20, 30↓
50 ED↓

LR↓                              ····· Program read example *1
START LINE=? 10↓                       (Reading the program on specified line
END LINE=? 50↓                         numbers through RS232C)
10 MO 1,C
20 MO 2,O
30 GC
40 DW +10.0, −20.0, +30.0,
50 ED

RN 10↓                           ····· Program execution example

WH↓                              ····· Current position read example

                X (mm)   Y (mm)   Z (mm)   P (deg)   R (deg)   H (O/C)
CUR. POS.       10.0     380.0    50.0     −70.0     −40.0     C

PR↓                              ····· Position read example *2
START POSITION= 1↓                     (Reading the position on specified line
END POSITION= 2↓                       numbers through RS232C)
POS. No.        X (mm)   Y (mm)   Z (mm)   P (deg)   R (deg)   H (O/C)
 PD 1,          0.0,     380.0,   300.0,   −70.0,    −40,      C
 PD 2,          −10.0,   350.0,   280.0,   −70.0,    −30,      O
```

*1, *2 :To correct any program and position read by using the cursor and required keys, the return key (↓) must be pushed at the end (on the right) of the line and position corrected.

*3 : Hexadecimal data in the program read is displayed in decimal (e.g. OD&FF → OD255).

5.2 Saving and Loading the Program and Position Data

The following BASIC program allows the required program and position data to be read from the drive unit and stored into floppy disk in the personal computer via RS-232C, or the other way. Note that some modifications of the program are necessary depending on the personal computer used.

```
1000  ' ********** RV-M2 PROGRAM & POSITION SAVE & LOAD **********
1010  ' ********** USING MULTI 16 BY MITSUBISHI ELECTRIC **********
1020  '
1030  PRINT "1. PROGRAM SAVE ?"
1040  PRINT "2. PROGRAM LOAD ?"
1050  PRINT "3. POSITION SAVE ?"
1060  PRINT "4. POSITION LOAD ?"
1070  INPUT ; N
1080  IF N=1 THEN 1150
1090  IF N=2 THEN 1300
1100  IF N=3 THEN 1400
1110  IF N=4 THEN 1550
1120  GOTO 1070
1130  '
1140  ' ********** PROGRAM SAVE **********
1150  OPEN "COM1 : 9600, E, 7, 2" AS #1
1160  INPUT "FILE NAME (*.PRG)=" ; F$
1170  OPEN "A :" +F$ FOR OUTPUT AS #2
1180  INPUT "START LINE=" ; S
1190  INPUT "END  LINE=" ; E
1200  FOR I=S TO E
1210      PRINT#1, "LR"+STR$ (I)
1220      LINE INPUT #1, A$
1230      IF A$=" " THEN 1250
1240  PRINT #2, I ; :PRINT #2, A$ :PRINT I ; : PRINT A$
1250  NEXT I
1260  CLOSE
1270  GOTO 1030
1280  '
1290  ' ********** PROGRAM LOAD **********
1300  OPEN "COM1 : 9600 , E, 7, 2" AS #1
1310  INPUT "FILE NAME (*.PRG)=" ; F$
1320  OPEN "A :" +F$ FOR INPUT AS #2
1330  LINE INPUT #2, A$
1340  PRINT #1, A$ : PRINT A$
1350  IF EOF (2)=-1 THEN 1360 ELSE 1330
1360  CLOSE
1370  GOTO 1030
1380  '
```

For NEC's PC9801, modify the program as follows:

```
1150 OPEN "COM1 : E73" AS #1
1300 OPEN "COM1 : E73" AS #1
1400 OPEN "COM1 : E73" AS #1
1550 OPEN "COM1 : E73" AS #1
```

Some file open commands may be different.

Example: OPEN "A:" → OPEN "1:"

For Mitsubishi's A6GPPE, modify the program as follows:

1025 INIT %1, "6E72"

Delete lines 1150, 1300, 1400 and 1550.

In addition, modify the following commands:

PRINT #1	→ PRINT %1
LINE INPUT #1	→ LINE INPUT %1
OPEN "A:"	→ OPEN "2:"
CLOSE	→ CLOSE #2

```
1390  ' ******* POSITION SAVE ********
1400  OPEN "COM1 : 9600, E, 7, 2" AS #1
1410  INPUT "FILE NAME (*. POS)=" ; F$
1420  OPEN "A :" +F$ FOR OUT PUT AS #2
1430  INPUT "START POSITION=" ; S
1440  INPUT "END POSITION=" ; E
1450  FOR I=S TO E
1460      PRINT#1, "PR"+STR$ (I)
1470      LINE INPUT #1, A$
1480      IF A$="0, 0, 0, 0, 0  THEN 1500
1490      PRINT #2, "PD" ; I ; ", " ; A$ : PRINT "PD" ; I ; ", " ; A$
1500  NEXT I
1510  CLOSE
1520  GOTO 1030
1530  '
1540  ' ******* POSITION LOAD ********
1550  OPEN "COM1 : 9600, E, 7, 2" AS #1
1560  INPUT "FILE NAME (*.POS)=" ; F$
1570  LINE INPUT #2, A$
1580  OPEN "A :" +F$ FOR INPUT AS #2
1590  PRINT #1, A$ : PRINT A$
1600  IF EOF (2)=-1 THEN 1610 ELSE 1580
1610  CLOSE
1620  GOTO 1030
```

Note: When the program and position data has been loaded into the drive unit as indicated above, the line and position numbers loaded from the personal computer remain unchanged. When newly loading the data, therefore, the following lines must be added to the above program:

```
1305 PRINT #1, "DL1, 3584" ····· Deletes all program.
1555 PRINT #1, "PC1, 999" ····· Deletes all positions.
```

〈Explanation〉

Start: After the program has been written, run the program. The following menu is then displayed:

```
1. PROGRAM SAVE? (Program data save)
2. PROGRAM LOAD? (Program data load)
3. POSITION SAVE? (Position data save)
4. POSITION LOAD? (Position data load)
? □
```

"SAVE" indicates that data is transferred from the drive unit to the personal computer floppy disk. "LOAD" indicates that data is transferred from the floppy disk to the drive unit.

Selection: Type the required number and push the return key. Operation examples are indicated on the next page.

Example 1: Program save

```
FILE NAME (*.PRG) = ? TEST.PRG ↓ ····· Enters any file name.
START LINE=? 10 ↓ ····· Enters the head line number of program data
                        to be saved.
END LINE=? 100 ↓ ····· Enters the final line number of program data
                       to be saved.
```

The above operation stores the specified area of the program (lines 10 to 100) onto the floppy disk as file "TEST.PRG".

Example 2: Program load

```
FILE NAME (*.PRG) = ? TEST.PRG ↓ ····· Enters the existing file
                                       name.
```

The above operation transfers the program from the specified file (TEST.PRG) to the drive unit. The area need not be specified.

Example 3: Position save

```
FILE NAME (*.POS) = ? TEST.POS ↓ ····· Enters any file name.
START POSITION=? 100 ····· Enters the head number of position data
                           to be saved.
END POSITION=? 200 ····· Enters the final number of position data
                         to be saved.
```

The above operation stores the specified area of the position data (positions 100 to 200) onto the floppy disk as file "TEST.POS".

Example 4: Position load

```
FILE NAME (*.POS) = ? TEST.POS ↓ ····· Enters the existing file
                                       name.
```

The above operation transfers the position data from the specified file (TEST.POS) to the drive unit. The area need not be specified.

REMARKS

Note that after saved and re-loaded, the position data taught does not strictly match the original position data. (Because the internal data is rounded down to one decimal place when saved.)

The position data saved is treated as a position where the coordinates of the end of the hand are ahead of the robot's Z axis (to the front of the robot). Hence, note that if a position where the coordinates of the end of the hand are at the back of the Z axis has been saved and then re-loaded, that position may become different from the original position and result in an error. (The J1 axis may be different 180°.)

6. SAMPLE PROGRAMS

The following samples give programs wherein intelligent commands are used. Note that the first line numbers of the sample programs are that of the Movemaster program, and not that of the BASIC.

Example 1: Pick-and-place work

The following program causes the robot to shift a workpiece from one place to another. The robot is only taught through positions 1 and 2 and the aerial distances of travel from the respective positions are to be predefined by the "PD" command.

〈Positions used〉

Position 1: At which the workpiece is grasped.
Position 2: Onto which the workpiece is placed.
} ··· Teaching

Position 10: Aerial distance of travel from position 1.
Position 20: Aerial distance of travel from position 2.
} ··· Numeric values are predefined.

〈Sample program〉

```
PD 10, 0, 0, 20, 0, 0      ; Defines the aerial distance of travel from
   (X, Y, Z, P, R)           position 1 (Z = 20mm) in direct mode and
                             identifies the aerial position as position 10.
PD 20, 0, 0, 30, 0, 0      ; Defines the aerial distance of travel from
   (X, Y, Z, P, R)           position 2 (Z = 30mm) in direct mode and
                             identifies the aerial position as position 20.
30 SP 17                   ; Sets the initial speed.
40 MA 1, 10, O             ; Moves the robot to a location above the
                             workpiece (20mm above position 1) with the
                             hand open.
50 MO 1, O                 ; Moves the robot to the workpiece (to position
                             1).
60 GC                      ; Closes the hand to grasp the workpiece.
70 MA 1, 10, C             ; Moves the robot above position 1 (distance
                             20mm) with the workpiece grasped.
80 MA 2, 20, C             ; Moves the robot to a location 30mm above
                             position 2.
90 MO 2, C                 ; Moves the robot to position 2.
100 GO                     ; Opens the hand to release the workpiece.
110 MA 2, 20, O            ; Moves the robot above position 2 (distance
                             30mm) with the workpiece grasped.
120 GT 40                  ; Returns the robot to position 10 allowing it to
                             repeat the sequence (jump to line number 40).
```

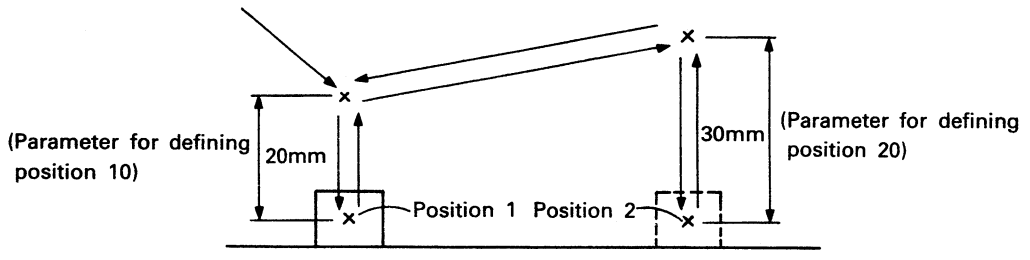


Fig. 5.6.1 Pick-and-Place Work

Example 2: Interrupt

The following program causes the robot to grasp workpieces of varying heights by means of the hand equipped with a limit switch. It assumes that the limit switch signal is coupled to the hand check confirmation input terminal.

〈Position used〉

Position 1 : A position above the workpiece (teaching)

〈Input signal used〉

Bit 16 : Workpiece detecting signal.

〈Sample program〉

```
    {
 90 SP 10          ; Selects speed 10.
100 EA +16, 140    ; Enables interrupt to be executed by bit 16.
110 MO 1, O        ; Moves the robot to a position above the
                     workpiece.
120 DS 0, 0, -50   ; Moves the robot 50mm in -Z direction (linear
                     interpolation).
130 GT 110         ; Jumps to line number 110 causing the robot to
                     return to position 1 as no workpiece has been
                     detected.
140 DA 16          ; Disables interrupt from being executed by bit
                     16.
150 GC             ; Closes the hand to grasp the workpiece.
160 MO 1, C        ; Moves the robot to position 1 with the work-
    }                piece grasped.
```

*In the above sample program, line number 120 causes the robot to move 50mm in the −Z tool direction and, if there is a workpiece, the limit switch signal is input and the robot stopped. Then, the program jumps to line number 140 to disable interrupt and to allow the robot to grasp the workpiece and return to position 1. If there is no workpiece in the robot's moving range, no limit switch signal is input and line number 130 causes the program to jump to line number 110 which, in turn, causes the robot to return to position 1 repeating the same sequence.

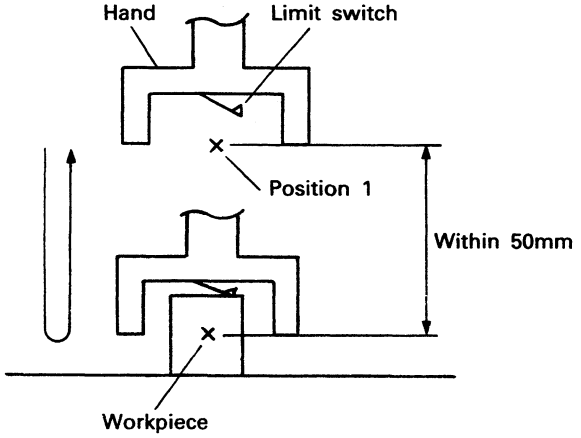


Fig. 5.6.2 Interrupt

Example 3: Palletizing

The following program causes the robot to pick up a workpiece from a pallet containing workpieces subject to test, set it up on the test equipment, and place it in position in another pallet to contain the workpieces that have undergone the test. The program assumes that the shapes of the two pallets are different.

〈Positions used〉

Position 1 : Pallet 1 set position
Position 2 : Pallet 2 set position
} ··· Defined by PT command

Position 10: Pallet 1 reference position
Position 11: Pallet 1 column terminating position
Position 12: Pallet 1 row terminating position
Position 13: Pallet 1 corner position opposite to reference
Position 20: Pallet 2 reference position
Position 21: Pallet 2 column terminating position
Position 22: Pallet 2 row terminating position
Position 23: Pallet 2 corner position opposite to reference
} ··· Teaching

Position 30: Test equipment set position
Position 50: Aerial distance of travel from pallets ····· Numeric values are predefined.

〈Counters used〉

Counter 11: Pallet 1 column counter
Counter 12: Pallet 1 row counter
Counter 21: Pallet 2 column counter
Counter 22: Pallet 2 row counter

〈Input signal used〉

Bit 7 : Test complete signal

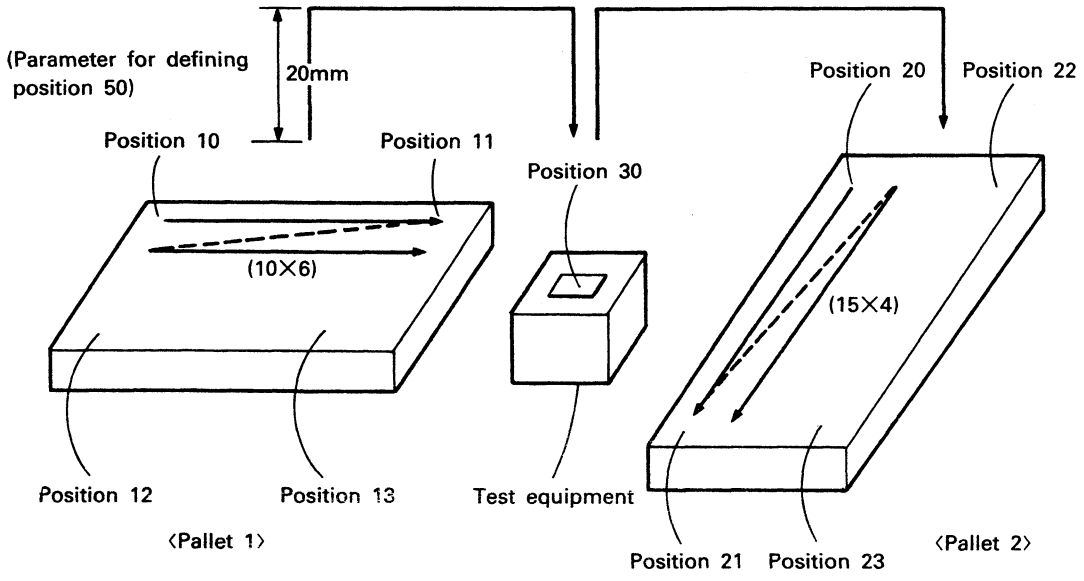


Fig. 5.6.3 Palletizing

〈Sample program〉

```
PD 50, 0, 0, 20, 0, 0      ; Defines the aerial distance of travel (Z =
   (X, Y, Z, P, R)           20mm) in direct mode and identifies the aerial
                             position as position 50.
```

(Initial setting)

```
10 NT                      ; Nesting
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                ; Loads initial value in pallet 1 column counter
                             11. (Sets value 1).
40 SC 12, 1                ; Loads initial value in pallet 1 row counter 12.
                             (Sets value 1).
45 SC 21, 1                ; Loads initial value in pallet 2 column counter
                             21. (Sets value 1).
50 SC 22, 1                ; Loads initial value in pallet 2 row counter 22.
                             (Sets value 1).
```

(Main program)

```
100 RC 60                  ; Sets the number of repeat cycles of a loop up to
                             line number 140.
110 GS 200                 ; Causes the robot to pick up a workpiece from
                             pallet 1.
120 GS 300                 ; Causes the robot to set up the workpiece on
                             test equipment.
130 GS 400                 ; Causes the robot to place the workpiece in
                             pallet 2.
140 NX                     ; Returns to line number 100.
150 ED                     ; End.
```

*The following program indicates subroutines used in the main program:

(Subroutine: Picking up the workpiece to be tested)

```
200 SP 17                  ; Sets speed.
202 PT 1                   ; Identifies the coordinates of calculated grid
                             point on pallet 1 as position 1.
204 MA 1, 50, O            ; Moves the robot to a location above position 1
                             (distance = 20mm in Z direction).
206 SP 8                   ; Sets speed.
208 MO 1, O                ; Moves the robot to position 1.
210 GC                     ; Closes the hand to grasp the workpiece.
212 MA 1, 50, C            ; Moves the robot to a location above position 1
                             with the workpiece grasped (distance = 20mm
                             in Z direction).
214 IC 11                  ; Increments pallet 1 column counter 11 by 1.
216 CP 11                  ; Loads the value in counter 11 into the internal
                             comparison register.
218 EQ 11, 230             ; Jumps to line number 230 on completing the
                             column line. (Compares with value 11.)
```

```
220 RT          ; Ends the subroutine otherwise.
230 SC 11, 1    ; Initializes counter 11. (Sets value 1.)
232 IC 12       ; Increments pallet 1 row counter 12 by 1.
234 RT          ; Ends the subroutine.
```

(Subroutine: Setting up the workpiece on the test equipment)

```
300 SP 17           ; Sets speed.
302 MT 30, -50, C   ; Moves the robot to a location 50mm ahead of
                      the test equipment.
304 SP 8            ; Sets speed.
306 MO, 30, C       ; Causes the robot to set up the workpiece on the
                      test equipment.
308 ID              ; Fetches input data.
310 TB -7, 308      ; Causes the robot to wait for the test to com-
                      plete.
312 MT 30, -50, C   ; Moves the robot to a location 50mm ahead of
                      the test equipment.
314 RT              ; Ends the subroutine.
```

(Subroutine: Placing the tested workpiece in pallet 2)

```
400 SP 17           ; Sets speed.
402 PT 2            ; Identifies the coordinates of calculated grid
                      point on pallet 2 as position 2.
404 MA 2, 50, C     ; Moves the robot to a location above position 2
                      (distance = 20mm in Z direction).
406 SP 8            ; Sets speed.
408 MO 2, C         ; Moves the robot to position 2.
410 GO              ; Opens the hand to grasp the workpiece.
412 MA 2, 50, O     ; Moves the robot to a location above position 2
                      (distance = 20mm in Z direction).
414 IC 21           ; Increments pallet 2 column counter 21 by 1.
416 CP 21           ; Loads the value in counter 21 into the internal
                      comparison register.
418 EQ 16, 430      ; Jumps to line number 430 on completing the
                      column line. (Compares with value 16.)
420 RT              ; Ends the subroutine otherwise.
430 SC 21, 1        ; Initializes counter 21. (Sets value 1.)
432 IC 22           ; Increments pallet 2 row counter 22 by 1.
434 RT              ; Ends the subroutine.
```

*In the above sample program, the column counter of each pallet is incremented by 1 and is initialized when the robot reaches the terminating end of the column. The row counter is then incremented to allow the robot to move to the next column line. (See line numbers 214 to 232, 414 to 432.)

*The robot waits until the test complete signal is input. (See line number 310.)

*The completion of the entire sequence is determined by the number of main program cycles. (See line number 100.)

Example 4: Connection with external I/O equipment

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.

〈Connection〉

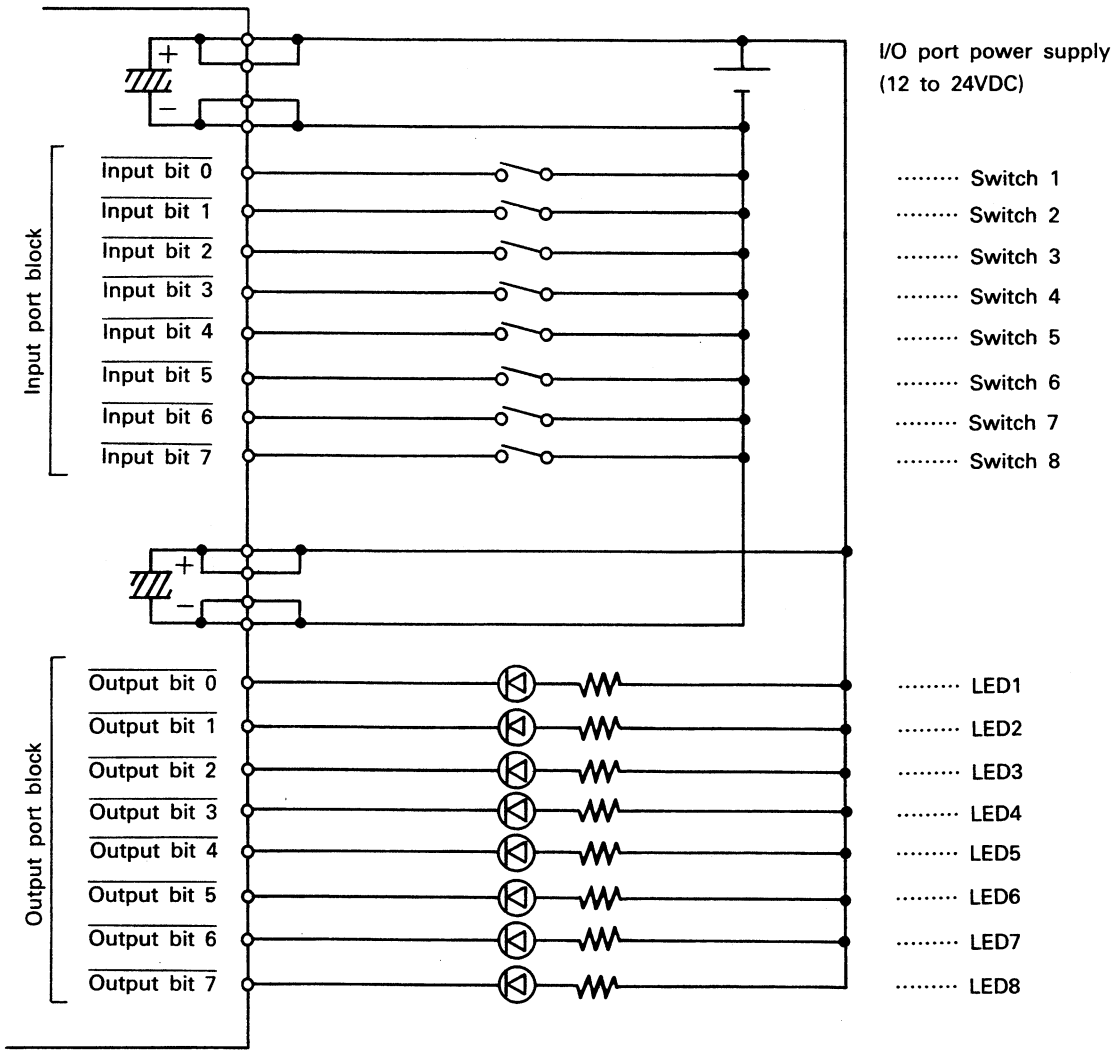
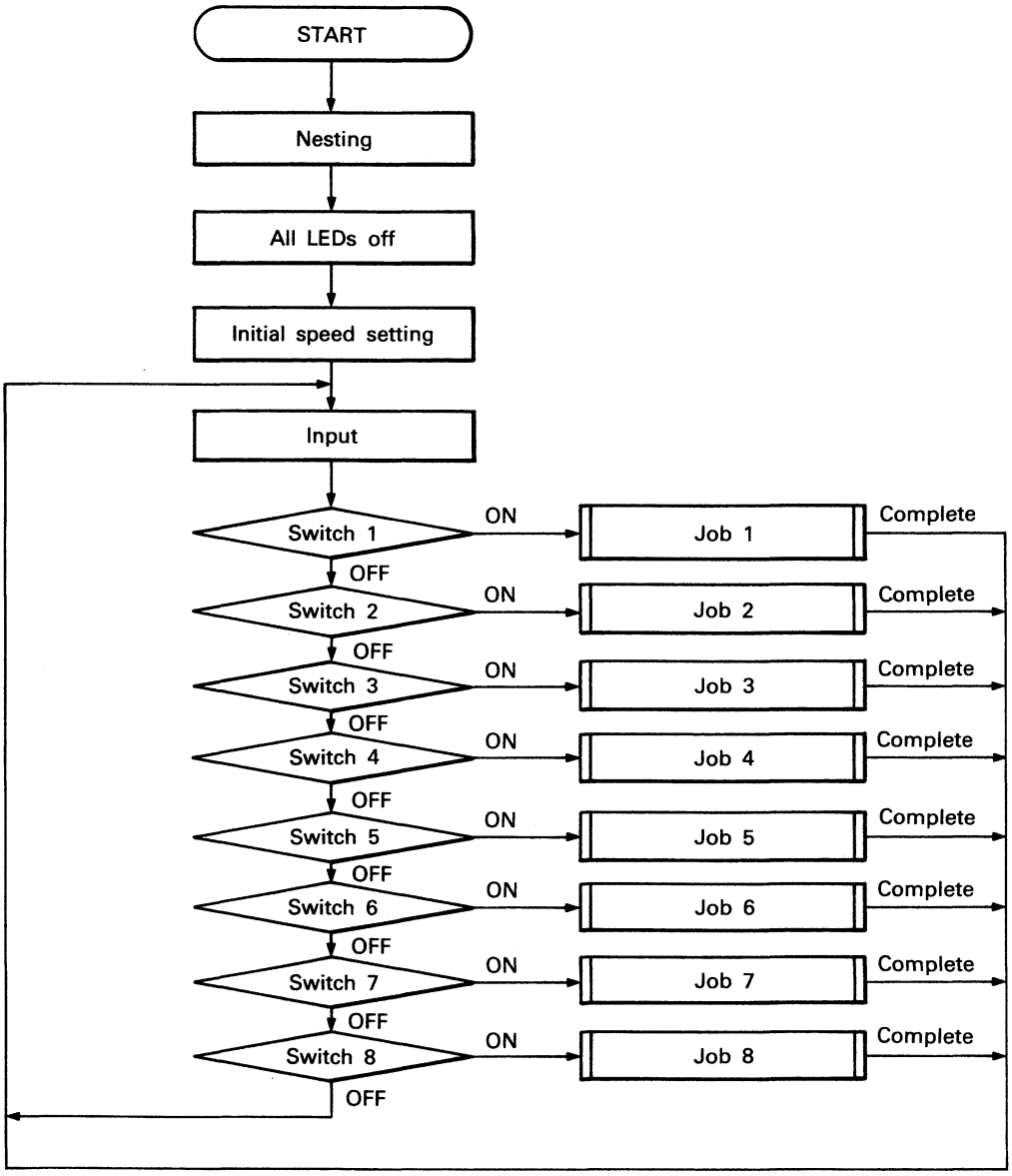


Fig. 5.6.4 Typical Connection with External I/O Equipment

〈Flowchart〉



Example

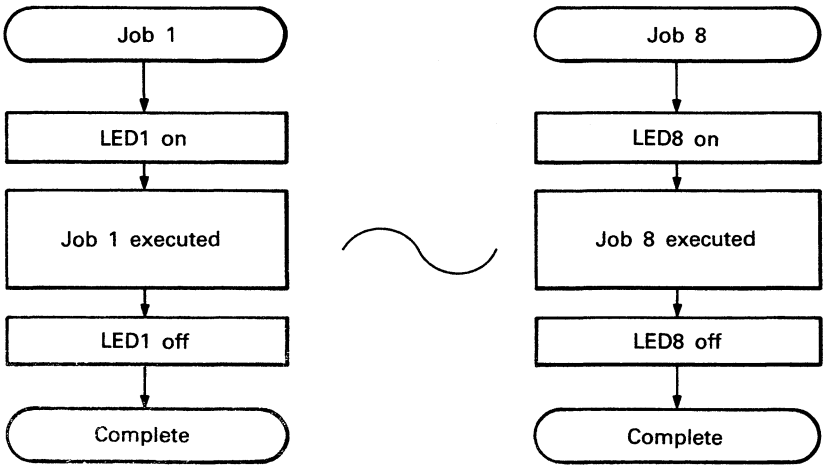


Fig. 5.6.5 Flowchart

〈Sample program〉

(Main program)

```
10 NT              ; Nesting
15 OD 0            ; Switches off all LEDs.
20 SP 5            ; Initial speed setting.
25 ID              ; Input.
30 TB +0, 100      ; Jumps to line number 100 when switch 1 is
                     turned on. (Job 1)
31 TB +1, 200      ; Jumps to line number 200 when switch 2 is
                     turned on. (Job 2)
32 TB +2, 300      ; Jumps to line number 300 when switch 3 is
                     turned on. (Job 3)
33 TB +3, 400      ; Jumps to line number 400 when switch 4 is
                     turned on. (Job 4)
34 TB +4, 500      ; Jumps to line number 500 when switch 5 is
                     turned on. (Job 5)
35 TB +5, 600      ; Jumps to line number 600 when switch 6 is
                     turned on. (Job 6)
36 TB +6, 700      ; Jumps to line number 700 when switch 7 is
                     turned on. (Job 7)
37 TB +7, 800      ; Jumps to line number 800 when switch 8 is
                     turned on. (Job 8)
38 GT 25           ; Returns to line number 25 (when all switches
                     are off).
```

(Subroutines)

```
100 OB +0          ; Switches on LED1 (job started).
105 MO 10          ; Executes job 1.
    ⋮                  ⋮                          } Job 1
198 OB -0          ; Switches off LED1 (job complete).
199 GT 25          ; Returns to line number 25.

800 OB +7          ; Switches on LED8 (job started).
805 MO 80          ; Executes job 8.
    ⋮                  ⋮                          } Job 8
898 OB -7          ; Switches off LED8 (job complete).
899 GT 25          ; Returns to line number 25.
```

7. OPERATIONAL SPACE DIAGRAM

Use the following diagram for examining the layout of peripheral equipment and pallet sizes:

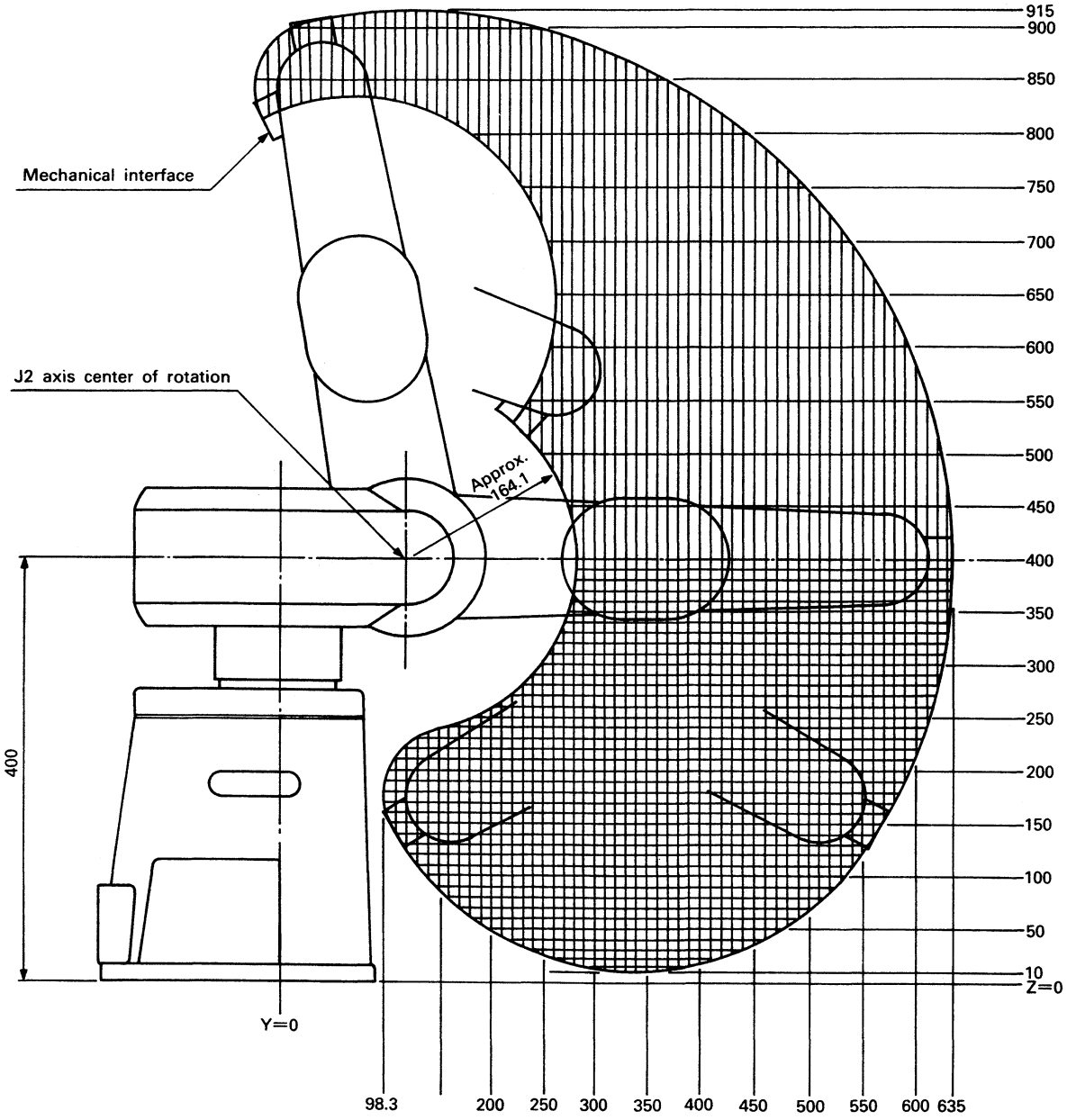


Fig. 5.7.1 Operational Space Diagram

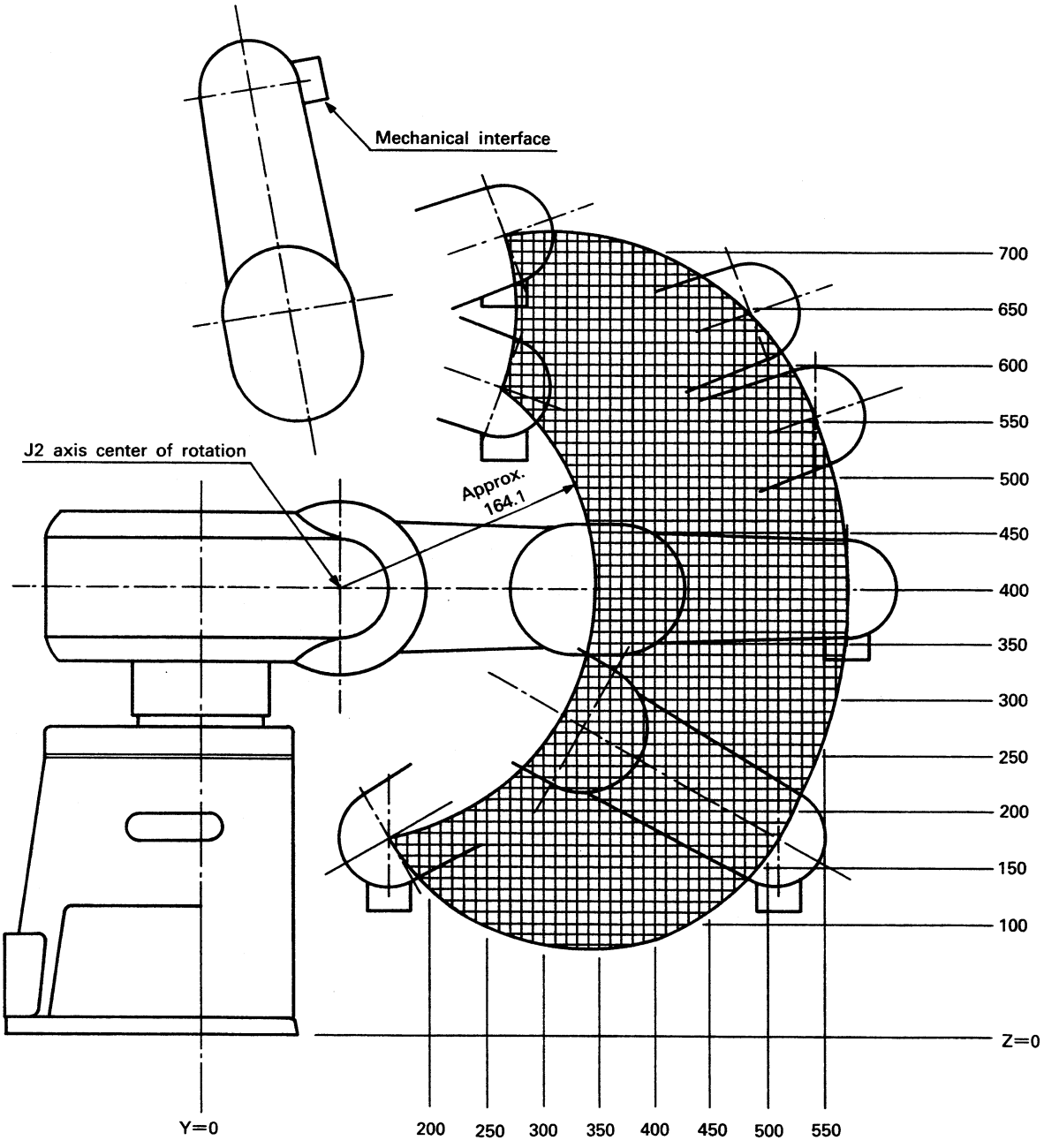


Fig. 5.7.2 Operational Space Diagram

8. TECHNICAL INFORMATION

8.1 Reduction Ratios and Encoder Resolutions

Item	Unit	J_1	J_2	J_3	J_4	J_5
Reduction ratio	—	128	161	129	110	80.88
Encoder resolution	Pulses/rotation	512	512	512	512	512

Table 5.8.2 Reduction Ratios and Encoder Resolutions

Note: The command and feedback pulses for control are quadrupled to 2048 pulses per rotation as to all axes.

8.2 Service Life of DC Servo Motors

The service life of each DC servo motor depends on those of the brush and bearing grease, which are indicated in Table 5.8.3. These service lives assume that the motor is used with the rated load at the rated speed, and vary greatly depending on the operating conditions. Use the values as guidelines for changing the brush and motor.

Item	J_1	J_2	J_3	J_4	J_5
Brush life	2000 hours	2000 hours	2000 hours	2000 hours	2000 hours
Bearing grease life	20000 hours	20000 hours	20000 hours	20000 hours	20000 hours

Table 5.8.3 Motor Brush and Bearing Grease Lives

8.3 Selection of Wrist Roll (J5 Axis) Coordinate System

Either of the two wrist roll (J5 axis) coordinate representation systems (articulated angle and general angle systems) in the Movemaster coordinate system may be selected by setting the switch (bit 7 of SW1) in the drive unit in accordance with the actual work to be performed. The J5 axis is factory-set to the "articulated angle system."

(1) Articulated angle system (Set bit 7 of SW1 to lower position)

Definition: Defined by the mechanical operation space (−180° to +180°) of the wrist roll (J5 axis). Indicates a relative angle between the wrist roll and fore arm.

Features: If the wrist roll value is within 180 degrees in the coordinate data set by the PD, MP or other command, the wrist roll always satisfies the operation space. When the coordinate data is read by the WH or PR command, the position of the robot wrist can be estimated relatively easily by the value of the wrist roll. If the parallel movement of only the X and Y coordinate components is effected by the above indicated PD command, etc. with the wrist pointed straight down, the attitude of the wrist roll in the X-Y plane differs before and after the movement.

(2) General angle system (Set bit 7 of SW1 to upper position)

Definition: Defined by the angle created by the wrist roll in the X-Y plane and the +Y axis (robot front direction) with the wrist pointed straight down. When viewed from top, a rotation in the +X axis (CW) direction is defined as negative and a rotation in the −X axis (CCW) direction is defined as positive.

Features: Actually, the angle of difference between the wrist roll (J5 axis) and the then waist (J1 axis) is defined as a new wrist roll angle. Hence, the angle ranges between 330 degrees (from −180, −150 degrees to +180 +150 degrees) as coordinate data. If within the above range in the coordinate data set by the PD, MP or other command, the wrist roll value may not satisfy the operation space depending on the robot attitude. If the parallel movement of only the X and Y coordinate components is effected by the PD command, etc. with the wrist pointed straight down, the attitude of the wrist roll in the X-Y plane remains unchanged before and after the movement.

The above coordinate systems are both determined by the set position of bit 7 of SW1 when the drive unit is switched on. When changing the coordinate system, therefore, change the setting of bit 7 before switching on the power. (Any change of bit 7 setting is invalid after the power is switched on.)

CAUTION

Position data taught in one coordinate system is different in attitude of the wrist roll from that in the other coordinate system and they are not compatible.
If the coordinate system is changed , resume the operation from that described in Section 4.3 "Origin Attitude Setting" in OPERATION.

9. OPTIONAL FEATURES

9.1 Optional Line Voltage

The optional voltages in Table 5.9.1 are available for the drive unit. A power supply (24VDC, 0.7A) for the DC solenoid-driven air hand may also be incorporated in the drive unit.

Power Line Voltage	AC Solenoid Line Voltage	MAIN Destinations
100V AC	100V, 0.2A	Japan
120V AC	120V, 0.2A	U.S.A.
200V AC	24V, 0.8A	Japan, Hong Kong
220V AC		West Germany, France
230V AC		Singapore
240V AC		Great Britain, Australia

Table 5.9.1 Optional Line Voltages

9.2 Optional Origin Setting Position

Among the origin setting positions shown in Section 3.1.6 ("Origin setting", Vol. 1 SPECIFICATIONS), that of each axis may be changed to the negative operation end. In this case, there is restriction on the origin setting position. For more information, consult Mitsubishi.

10. COMMAND LIST [A] Position/Motion Control Instructions

Program Yes ····· Possible
No ····· Not possible

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
1	Decrement Position	DP	Moves robot to a predefined position with a position number preceding the current one.	Yes	Articulated interpolation	3-10
2	Draw Straight	DS x, y, z	Moves hand end to a position away from the current one covering the distance specified in X-, Y- and Z-axis directions.	Yes	Linear interpolation	3-12
3	Draw	DW x, y, z	Moves hand end to a position away from the current one covering the distance specified in X-, Y- and Z-axis directions.	Yes	Articulated interpolation	3-13
4	Here	HE a	Defines the coordinates of the current position by assigning position number (a) to it.	Yes	$1 \leqq a \leqq 999$	3-25
5	Home	HO	Establishes the reference position for Origin setting.	Yes		3-26
6	Increment Position	IP	Moves robot to a predefined position with a position number subsequent to the current one.	Yes	Articulated interpolation	3-30
7	Move Approach	MA a_1, a_2 [, O/C]	Moves hand end from the current position to a position away from position (a_1) in increments as specified for position (a_2).	Yes	$1 \leqq a_1, a_2 \leqq 999$ O: Hand opened; C: Hand closed. Articulated interpolation	3-34
8	Move Continuous	MC a_1, a_2	Moves robot continuously through predefined intermediate points between positions (a_1) and (a_2) by linear interpolation.	Yes	$1 \leqq a_1, a_2 \leqq 999$ $\lvert a_1 - a_2 \rvert \leqq 99$ O: Hand opened; C: Hand closed.	3-35
9	Move Joint	MJ w, s, e, p, r	Turns each joint the specified angle from the current position.	No	Articulated interpolation	3-37
10	Move	MO a [, O/C]	Moves hand end to position (a) by articulated interpolation.	Yes	$1 \leqq a \leqq 999$ O: Hand opened; C: Hand closed.	3-38
11	Move Position	MP x, y, z, p, r	Moves hand end to a position whose coordinates (position and angle) are specified as x, y, z, p and r.	No	Articulated interpolation	3-39
12	Move Straight	MS a [, O/C]	Moves hand end to position (a) by linear interpolation.	Yes	$1 \leqq a \leqq 999$ O: Hand opened; C: Hand closed.	3-40
13	Move Tool	MT a, b [, O/C]	Moves hand end from the current position to a position away from the specified position (a) in incremental distance b in the tool direction. (Unit: mm)	Yes	$1 \leqq a \leqq 999$ O: Hand opened; C: Hand closed. Articulated interpolation	3-41
14	Nest	NT	Returns robot to mechanical origin.	Yes		3-44
15	Origin	OG	Moves robot to the reference position for origin setting.	Yes	Articulated interpolation	3-49
16	Pallet Assign	PA i, j, k	Defines the number of grid points (j, k) in the column and row directions for pallet (i).	Yes	$1 \leqq i \leqq 9$ $1 \leqq j, k \leqq 32767$	3-52

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
17	Position Clear	PC a_1, [, a_2]	Clears all position data from positions a_1 to a_2.	No	$a_1 \leqq a_2$ $1 \leqq a_1, a_2 \leqq 999$ (or $a_1 = 0$)	3-53
18	Position Define	PD a, x, y, z, p, r [,O/C]	Defines the coordinates (x, y, z, p, r) of position (a).	No	$1 \leqq a \leqq 999$	3-54
19	Position Load	PL a_1, a_2	Assigns the coordinates of position (a_2) to position (a_1).	Yes	$1 \leqq a_1, a_2 \leqq 999$	3-55
20	Pallet	PT a	Calculates the coordinates of a grid point on pallet (a) and identifies the coordinates as position (a).	Yes	$1 \leqq a \leqq 9$	3-57
21	Position Exchange	PX a_1, a_2	Exchanges the coordinates of position (a_1) for those of position (a_2).	Yes	$1 \leqq a_1, a_2 \leqq 999$	3-61
22	Speed Define	SD a_1, [, a_2]	Defines operating speed (a1) and first order time constant (a2) for linear interpolation.	Yes	$1 \leqq$ a1 $\leqq 500.0$ $80 \leqq$ a2 $\leqq 300$	3-67
23	Shift	SF a_1, a_2	Shifts the coordinates of position (a_1) in increments representing the coordinates of position (a_2) and redefines the new coordinates.	Yes	$1 \leqq a_1, a_2 \leqq 999$	3-68
24	Speed	SP a [, H/L]	Sets the operating velocity and acceleration/deceleration time for robot. 0: Minimum speed; 20: Maximum speed	Yes	$0 \leqq a \leqq 20$ H: High accel/decel time; L: Low accel/decel time	3-70
25	Timer	TI a	Halts motion for time (a). (In increments of 0.1 seconds)	Yes	$0 \leqq a \leqq 32767$	3-73
26	Tool	TL a	Establishes distance (a) between hand installation surface and hand end.	Yes	$0 \leqq a \leqq +300.0$ Unit: mm	3-74

[B] Program Control Instructions

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
27	Counter Load	CL a	Loads internal register value into specified counter.	Yes	$1 \leqq a \leqq 99$	3-4
28	Compare Counter	CP a	Loads value in counter (a) into the internal register.	Yes	$1 \leqq a \leqq 99$	3-5
29	Disable Act	DA a	Disables interrupt by a signal through bit (a) of external input terminal.	Yes	$0 \leqq a \leqq 17$	3-7
30	Decrement Counter	DC a	Decrements counter (a) by 1.	Yes	$1 \leqq a \leqq 99$	3-8
31	Delete Line	DL a_1 [, a_2]	Deletes lines a_1 to a_2 from program.	No	$a_1 \leqq a_2$ $1 \leqq a_1, a_2 \leqq 3584$	3-9
32	Enable Act	EA a_1, a_2	Enables interrupt by a signal through bit (a_1) of external input terminal and specifies line number (a_2) to which the program jumps when interrupt occurs.	Yes	$-17 \leqq a_1 \leqq +17$ +: ON; −: OFF $1 \leqq a_2 \leqq 3584$	3-14
33	End	ED	Ends the program.	Yes		3-16
34	If Equal	EQ a_1 (or &b), a_2	Causes a jump to line number (a_2) if external input data or counter data equals a_1 (or &b).	Yes	$-32768 \leqq a_1 \leqq +32767$ $\&8000 \leqq b \leqq \&7FFF$ $1 \leqq a_2 \leqq 3584$	3-17

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
35	Go Sub	GS a	Permits the instruction sequence to jump to subroutine which starts with line number (a).	Yes	$1 \leqq a \leqq 3584$	3-23
36	Go To	GT a	Permits the instruction sequence to jump to line number (a) unconditionally.	Yes	$1 \leqq a \leqq 3584$	3-24
37	Increment Counter	IC a	Increments counter (a) by 1.	Yes	$1 \leqq a \leqq 99$	3-27
38	If Larger	LG a_1 (or &b), a_2	Causes a jump to line number (a_2) if external input data or counter data is greater than a_1 (or &b).	Yes	$-32768 \leqq a_1 \leqq +32767$ &8000 $\leqq$ b $\leqq$ &7FFF $1 \leqq a_2 \leqq 3584$	3-31
39	If Not Equal	NE a_1 (or &b), a_2	Causes a jump to line number (a_2) if external input data or counter data does not equal a_1 (or &b).	Yes	$-32768 \leqq a_1 \leqq +32767$ &8001 $\leqq$ b $\leqq$ &7FFF $1 \leqq a_2 \leqq 3584$	3-42
40	New	NW	Deletes all program and position data in RAM.	No		3-45
41	Next	NX	Specifies the range of a loop in a program executed by command RC.	Yes		3-46
42	Repeat Cycle	RC a	Repeats the loop specified by command NX (a) times.	Yes	$1 \leqq a \leqq 32767$	3-62
43	Run	RN a_1 [, a_2]	Executes lines (a_1) to (a_2) in program, (a_2) not included.	No	$1 \leqq a_1, a_2 \leqq 3584$	3-63
44	Return	RT	Completes subroutine activated by command GS and returns to main program.	Yes		3-65
45	Set Counter	SC a_1, [a_2]	Loads value (a_2) into counter (a_1).	Yes	$1 \leqq a_1 \leqq 99$ $-32768 \leqq a_2 \leqq +32767$	3-66
46	If Smaller	SM a_1 (or &b), a_2	Causes a jump to line number (a_2) if external input data or counter data is smaller than a_1 (or &b).	Yes	$-32768 \leqq a_1$ +32767 &8000 $\leqq$ b $\leqq$ &7FFF $1 \leqq a_2 \leqq 3584$	3-69

[C] Hand Control Instructions

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
47	Grip Close	GC	Closes hand grip.	Yes		3-19
48	Grip Flag	GF a	Defines the open/close state of hand grip, used in conjunction with command PD.	Yes	a = 0 (open), 1 (closed)	3-20
49	Grip Open	GO	Opens hand grip.	Yes		3-21
50	Grip Pressure	GP a_1, a_2, a_3	Defines motor-operated hand gripping force and gripping force retention time.	Yes	$0 \leqq a_1, a_2 \leqq 63$ $0 \leqq a_3 \leqq 99$ (in 0.1 second increments)	3-22

[D] I/O Control Instructions

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
51	AND	AN a (or &b)	ANDs internal register value and specified value a (or &b).	Yes		3-3
52	Input Direct	ID	Fetches external signal unconditionally from input port.	Yes		3-28

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
53	Input	IN	Fetches external signal synchronously from input port.	Yes		3-29
54	Output Bit	OB a	Sets the output state of bit (a) of external output terminal.	Yes	$-15 \leqq a \leqq +15$ +: ON; −: OFF	3-47
55	Output Direct	OD a (or &b)	Outputs data a (or &b) unconditionally through output port.	Yes	$-32768 \leqq a \leqq +32767$ $\&8000 \leqq b \leqq \&7FFF$	3-48
56	Or	OR a (or &b)	ORs internal register value and specified value a (or &b).	Yes		3-50
57	Output	OT a (or &b)	Outputs data a (or &b) synchronously through output port.	Yes	$-32768 \leqq a \leqq +32767$ $\&8000 \leqq b \leqq \&7FFF$	3-51
58	Test Bit	TB a_1, a_2	Causes a jump to line number a_2 in accordance with the state of bit (a_1) in external input terminal.	Yes	$-17 \leqq a_1 \leqq +17$ +: ON; −: OFF $1 \leqq a_2 \leqq 3584$	3-72
59	Exclusive Or	XO a (or &b)	EXCLUSIVE ORs internal register value and specified value a (or &b).	Yes		3-79

[E] RS232C Read Instructions

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
60	Counter Read	CR a	Reads contents of counter (a).	Yes	$1 \leqq a \leqq 99$	3-6
61	Data Read	DR	Reads data in external input terminal, used in conjunction with commands ID and IN.	Yes		3-11
62	Error Read	ER	Reads status of error (no error: 0; error mode I : 1; error mode II: 2).	No		3-18
63	Line Read	LR a	Reads program on line number (a).	No	$1 \leqq a \leqq 3584$	3-32
64	Nest Read	NR a	Reads the number of limit pulses of axis (a) after origin setting.	Yes	$1 \leqq a \leqq 5$	3-43
65	Position Read	PR a	Reads coordinates of position (a).	Yes	$1 \leqq a \leqq 999$	3-56
66	Where	WH	Reads coordinates of current position.	Yes		3-76
67	What Tool	WT	Read current tool length.	Yes		3-77

[F] Other Commands

No.	Description	Input Format	Function	Program	Remarks	Ref. Page
68	Reset	RS	Resets error mode II.	No		3-64
69	Transfer	TR	Transfers contents of EPROM to RAM.	No		3-75
70	Write	WR	Writes contents of RAM into EPROM.	No		3-78
71	Comment	'	Allows programmer to write a comment following '.	Yes		3-80

(1) MOVEMASTER CODING FORM

No. _____

Program Name :

Programmer :

Date • •

	LINE NO.		CODE	PARAMETER	COMMENT

(2) MOVEMASTER POSITION DATA SHEET No. ______

Program Name : Programmer : Date • •

	POS. NO.		X [mm]	Y [mm]	Z [mm]	P [deg]	R [deg]	L [mm]	H[O/C]	COMMENT

(3) MOVEMASTER I/O ASSIGN TABLE

No.

Program Name :

Programmer :

Date • •

INPUT			OUTPUT		
	bit	COMMENT		bit	COMMENT
	0			0	
	1			1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
	7			7	
	8			8	
	9			9	
	10			10	
	11			11	
	12			12	
	13			13	
	14			14	
	15			15	
	16				
	17				
	$\overline{\text{STB}}$			$\overline{\text{RDY}}$	
	ACK			$\overline{\text{BUSY}}$	
	$\overline{\text{START}}$			$\overline{\text{RUN}}$	
	$\overline{\text{STOP}}$			$\overline{\text{WAIT}}$	
	$\overline{\text{RESET}}$			$\overline{\text{ERROR}}$	

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

(9101)I3OD Printed in Japan