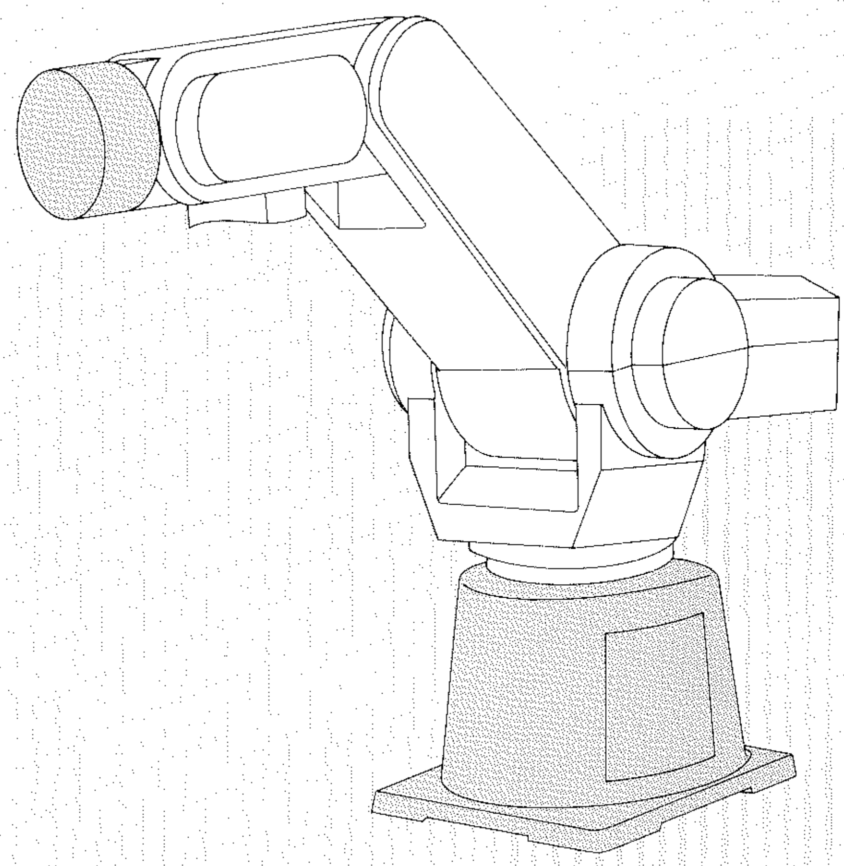
MITSUBISHI

INDUSTRIAL MICRO-ROBOT SYSTEM

Model

RV-M1



MOVEMASTER EX

BFP-A5191E-B

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 in accordance with functions. The commands appear 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, cartesian coordinate system reference position setting, command list, application programs, etc.

CONTENTS

1 SPECIFICATIONS

2 OPERATION

3 DESCRIPTION OF THE COMMANDS

4 MAINTENANCE AND INSPECTION

5 APPENDICES

1. SPECIFICATIONS

CONTENTS (SPECIFICATIONS)

1. SPECIFICATIONS

1. UNPACKING AND ACCEPTANCE INSPECTION

1.1 Unpacking Instructions

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

(2) Do not hold the cover (area A in Fig. 1.1.1.) when removing the robot.

(3) The origin limit switches and dogs (areas B, C) have been factory-adjusted. Do not touch them to ensure high repeatability.

(4) Brake is being applied to the arms (areas D, E) Do not force these arms to extend.

(5) Do not remove the arm fixing plate (area F) until the robot installation is complete. This plate protects the arm during transportation.
After removal, this arm fixing plate must be kept for future use for protection of the arm during transportation.

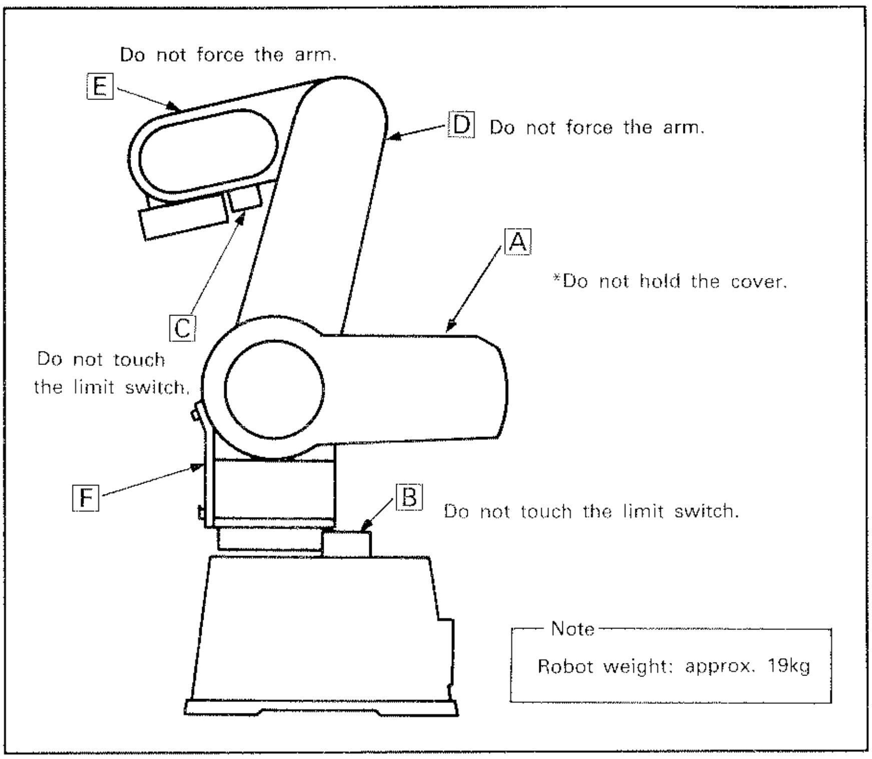


Fig. 1.1.1 Robot Attitude in the Package

The arm fixing plate must be reinstalled before transporting the robot.

1.2 Acceptance Inspection

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

No.	Description	Type	Quantity
1	Robot	RV-M1	1
2	Drive unit	D/U-M1	1
3	Motor signal cable (5m)	MS-M1	1
4	Motor power cable (5m)	MP-M1	1
5	Power cord (2.5m)	POW-M1	1
6	Spare fuse (10A)		1
7	Instruction manual	BFP-A5191	1
8	Warranty card		1
9	Installation bolt	M8 x 30	4
10	Spring washer for installation bolt	For M8	4
11	Flat washer for installation bolt	For M8	4
12	I/O card	A8 or A16 (B8 or B16)	1

Table 1.1.1 Standard Components

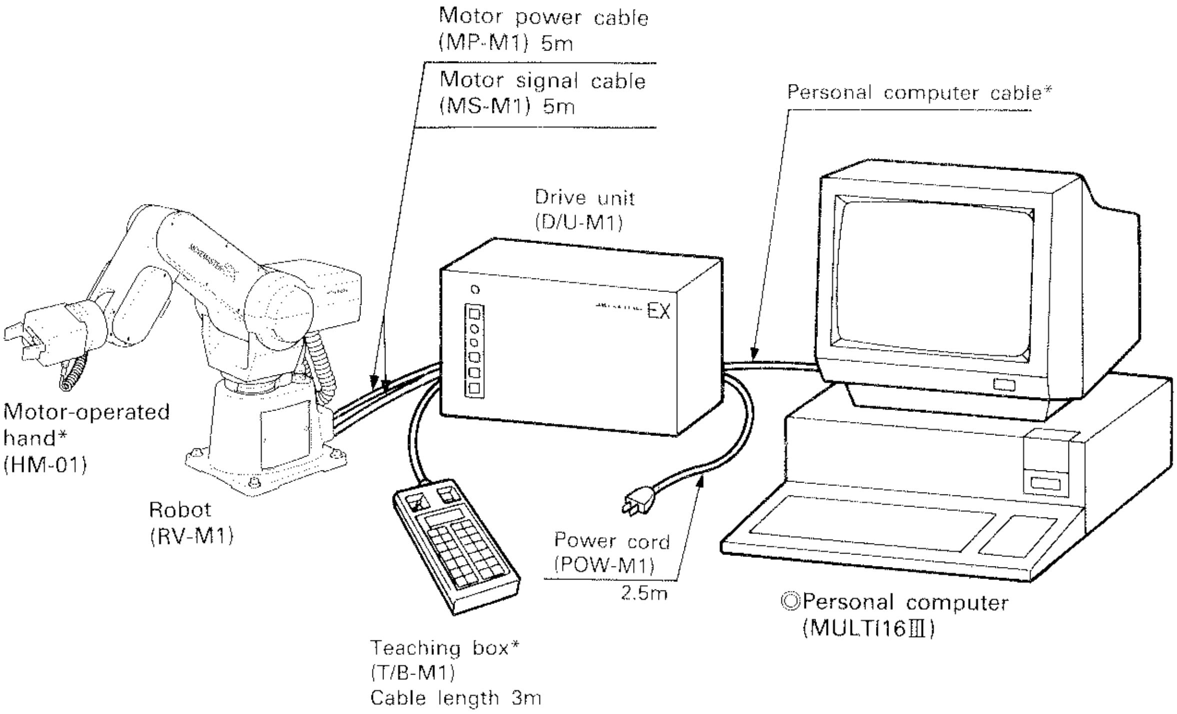
The following options are available.

No.	Description	Type
1	Teaching box	T/B-M1
2	Motor-operated hand	HM-01
3	EP-ROM	256K-ROM
4	External I/O cable	I/O-CBL (5m)
5	Personal computer cable	MULTI16, PC9801
6	Backup battery	BAT-M1

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-M1	Vertical articulated robot with 5 degrees of freedom.
	Drive unit			D/U-M1	Robot controller.
	Motor signal cable (5m)			MS-M1	Gives control signals from the drive unit to the robot.
	Motor power cable (5m)			MP-M1	Supplies power from the drive unit to the robot.
	Power cord (2.5m)			POW-M1	Supplies power to the drive unit.
	I/O card A8 (or B8)			I/O-A8 (I/O-B8)	8 inputs/8 outputs.
Optional equipment	I/O card A16 (or B16)			I/O-A16 (I/O-B16)	16 inputs/16 outputs.
	Teaching box (Cable length 3m)			T/B-M1	Handy control switch box with a cable for teaching, checking, correcting positions.
	Motor-operated hand			HM-01	Hand which is only used with the RV-M1 and allows 16-step holding power control.
	EP-ROM			256K-ROM	Stores written programs and set positions.
	Backup battery			BAT-M1	Backs up the memory during power off.
	External I/O cable (5m)			I/O-CBL	Connects with an external peripheral, e.g. programmable controller.
	Personal computer cable	MULTI16	RS-232-C	RS-MULTI-CBL (3m)	RS-232C cable used to connect the MULTI16.
			Centronics	C-MULTI-CBL (2m)	Centronics cable used to connect the MULTI16.
		PC9801	RS-232C	RS-PC-CBL (3m)	RS-232C cable used to connect the PC9801.
			Centronics	C-PC-CBL (1.5m)	Centronics cable used to connect the PC9801.
		Free cable	RS-232C	RS-FREE-CBL (3m)	RS-232C 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

Note: I/O card A8 or A16 is for sink load.
B8 or B16 is for source load.

3. MAIN SPECIFICATIONS

3.1 Robot

3.1.1 Nomenclature

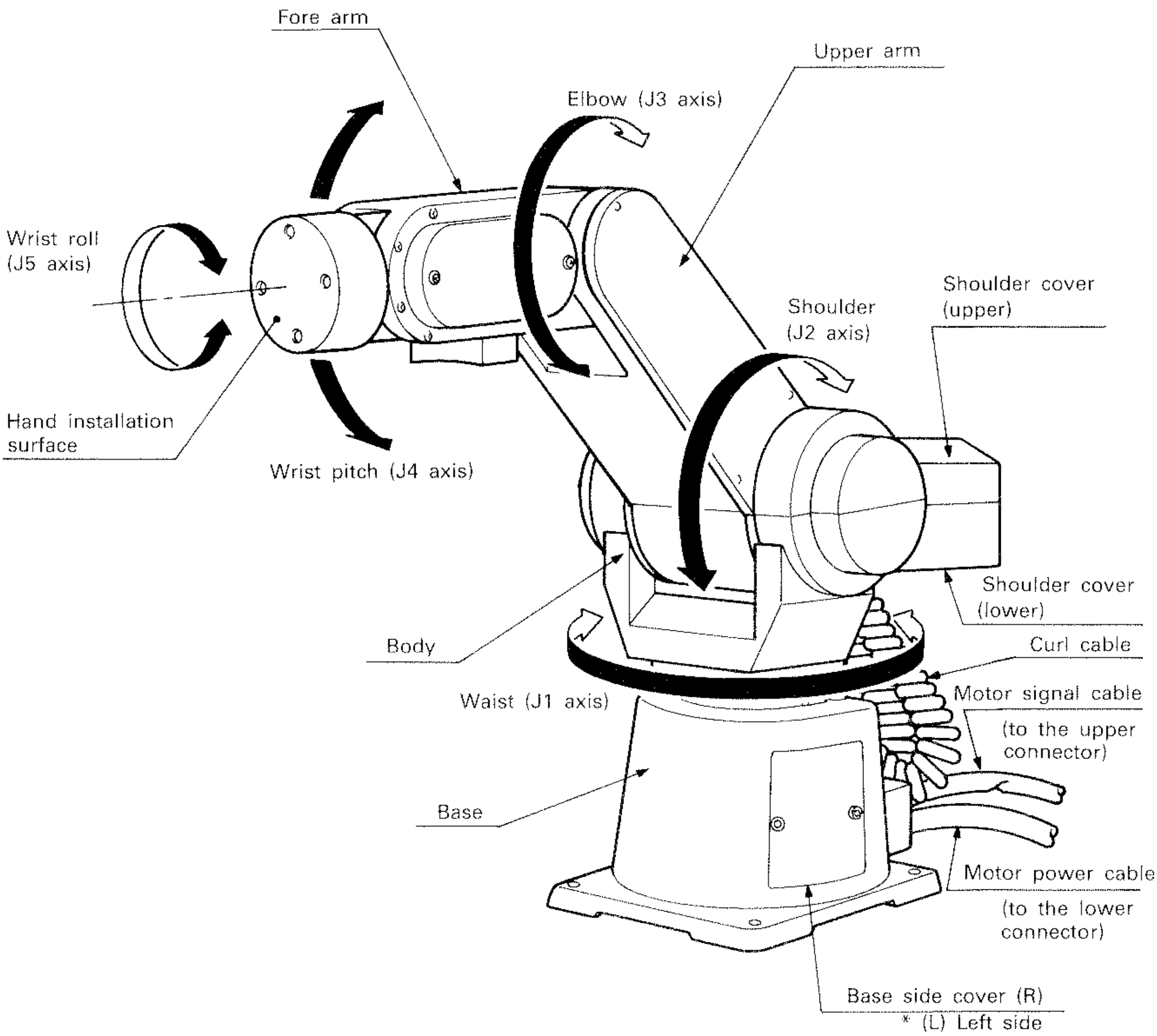


Fig. 1.3.1 Nomenclature (External View)

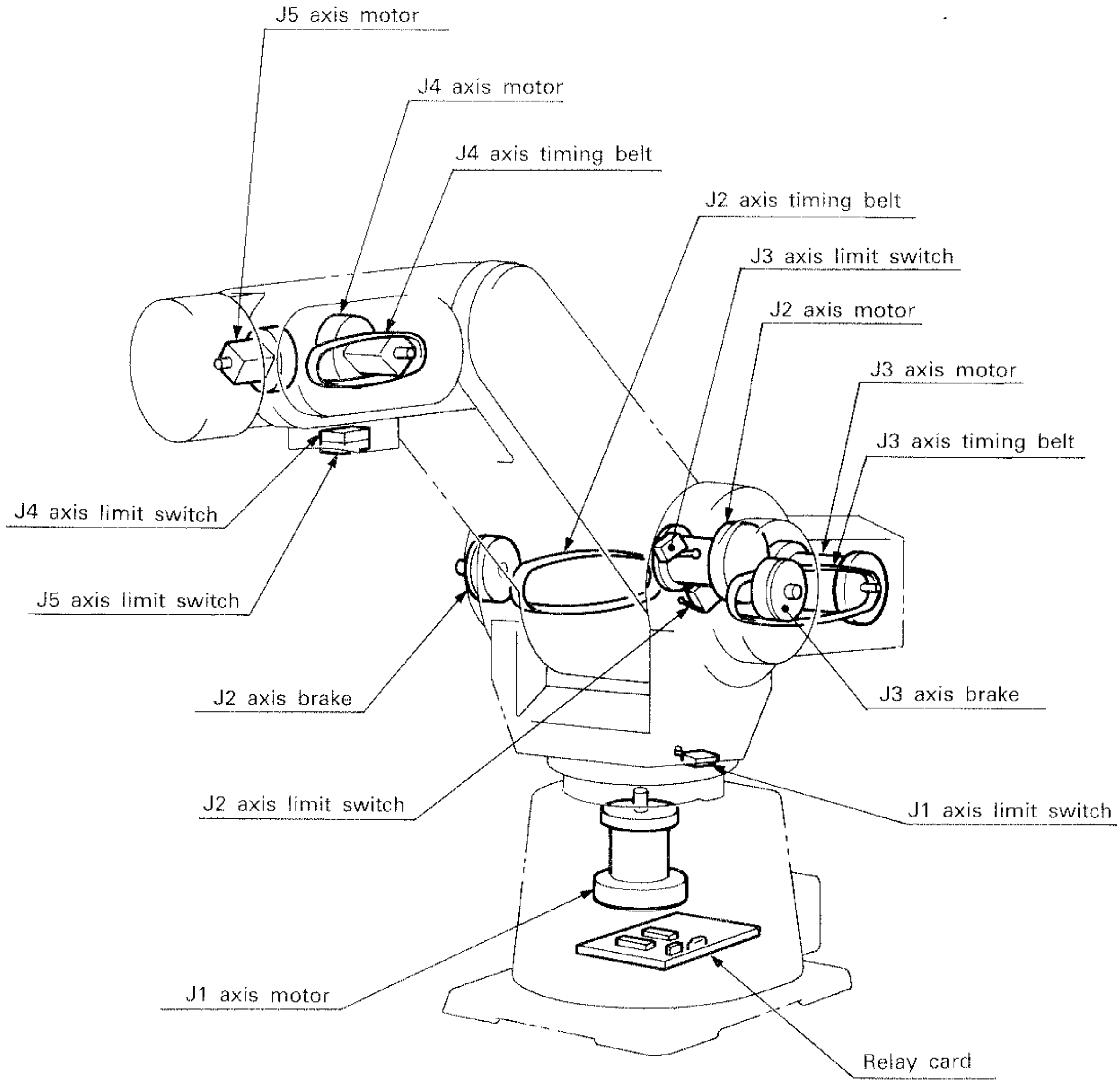


Fig. 1.3.2 Nomenclature (Internal View)

3.1.2 Standard specifications

Item		Specifications	Remarks
Mechanical Structure		5 degrees of freedom, vertical articulated robot	
Operation range	Waist rotation	300° (max. 120°/sec)	J1 axis
	Shoulder rotation	130° (max. 72°/sec)	J2 axis
	Elbow rotation	110° (max. 109°/sec)	J3 axis
	Wrist pitch	±90° (max. 100°/sec)	J4 axis
	Wrist roll	±180° (max. 163°/sec)	J5 axis
Arm length	Upper arm	250mm	
	Fore arm	160mm	
Weight capacity		Max. 1.2kgf (including the hand weight)	75mm from the mechanical interface (center of gravity)
Maximum path velocity		1000mm/sec (wrist tool surface)	Speed at point P in Fig. 1.3.4
Position repeatability		0.3mm (roll center of the wrist tool surface)	Accuracy at point P in Fig. 1.3.4
Drive system		Electrical servo drive using DC servo motors	
Robot weight		Approx. 19kgf	
Motor capacity		J1 to J3 axes: 30W; J4, J5 axes: 11W	

Table 1.3.1 Standard Specifications

3.1.3 External dimensions

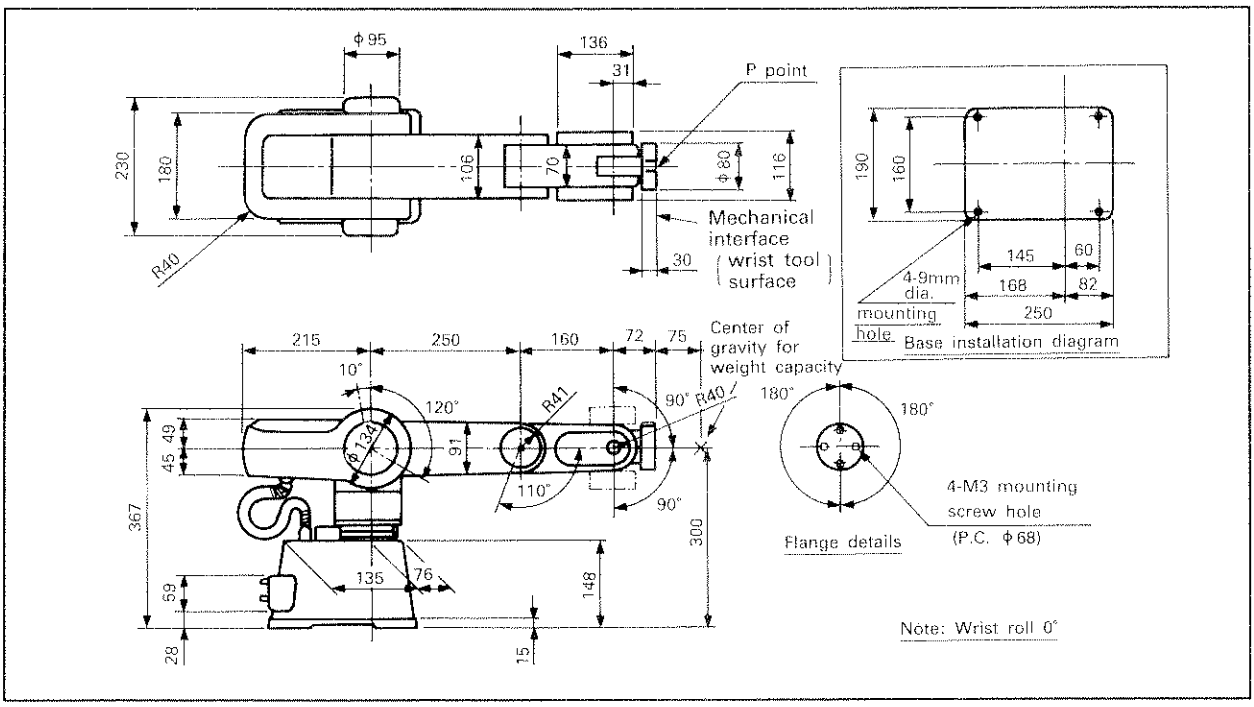


Fig. 1.3.3 External Dimensions

3.1.4 Operation space

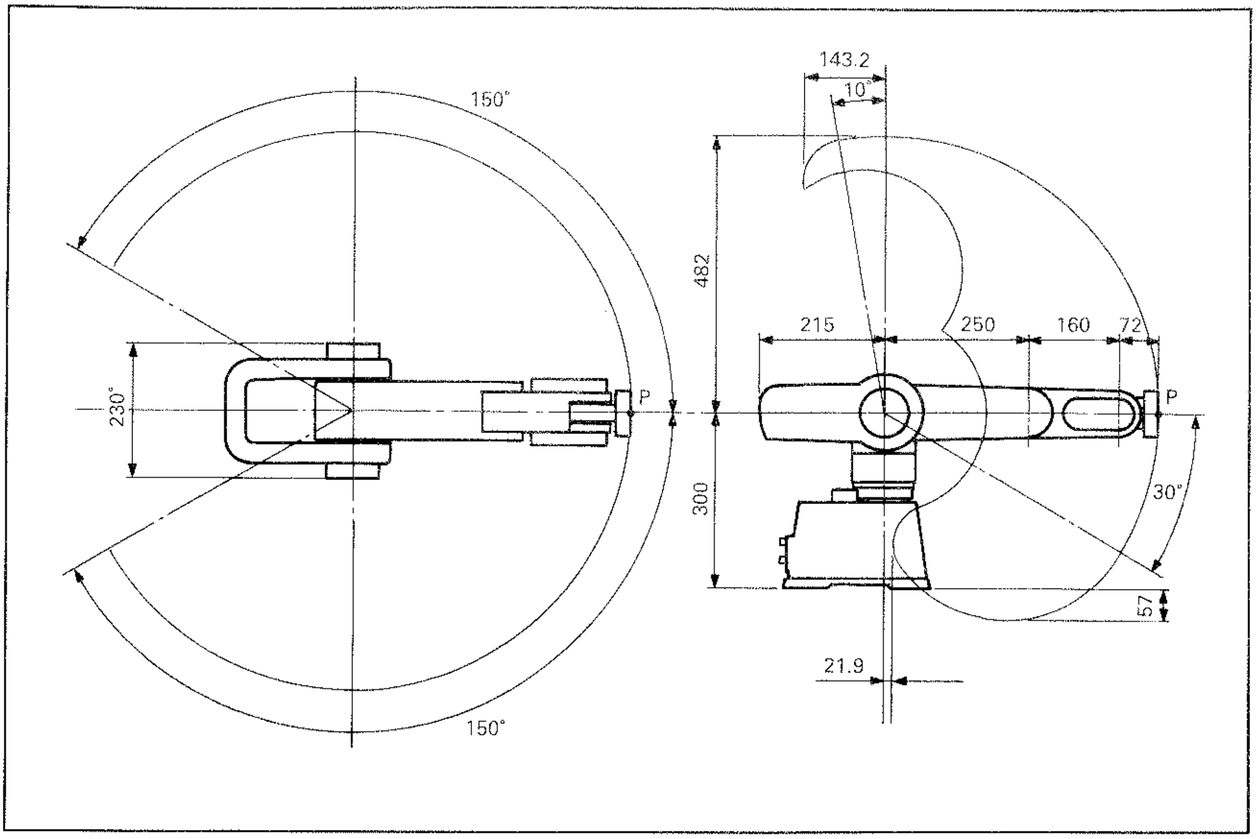


Fig. 1.3.4 Operation Space

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

Note 2: The wrist pitch operation may be restricted in some area depending the upper arm and fore arm positions. For details, see Fig. 5.11.1 in the appendix.

Note 3: Jog operation must be performed with special care as the wrist may interfere with the robot base and floor surface.

REMARKS
Jog operation indicates a manual operation using the teaching box.

3.1.5 Basic operations

Fig. 1.3.5 shows axis operations in the articulated system.

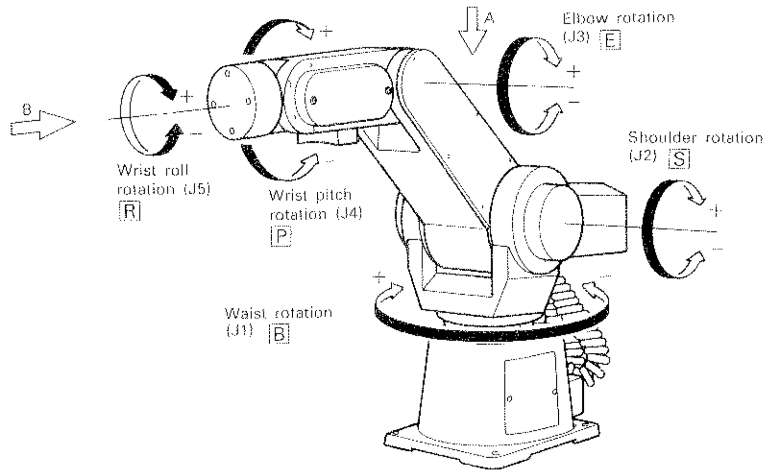


Fig. 1.3.5 Operations in Articulated System

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

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

Fig. 1.3.6 shows operations in the cartesian coordinate system.

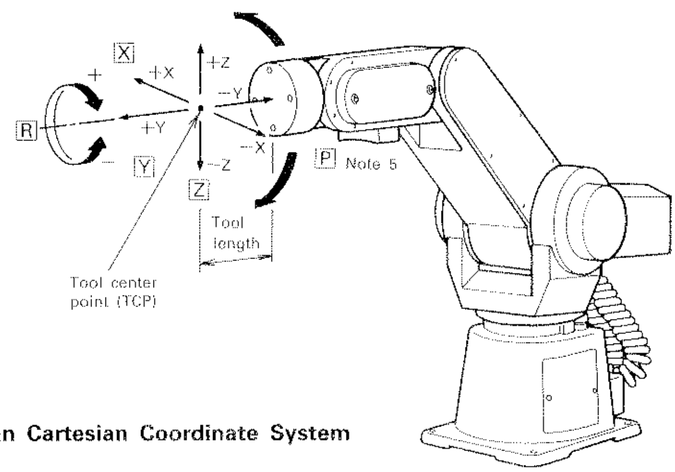


Fig. 1.3.6 Operations in Cartesian Coordinate System

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

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

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

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

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

3.1.6 Origin set (Return to origin)

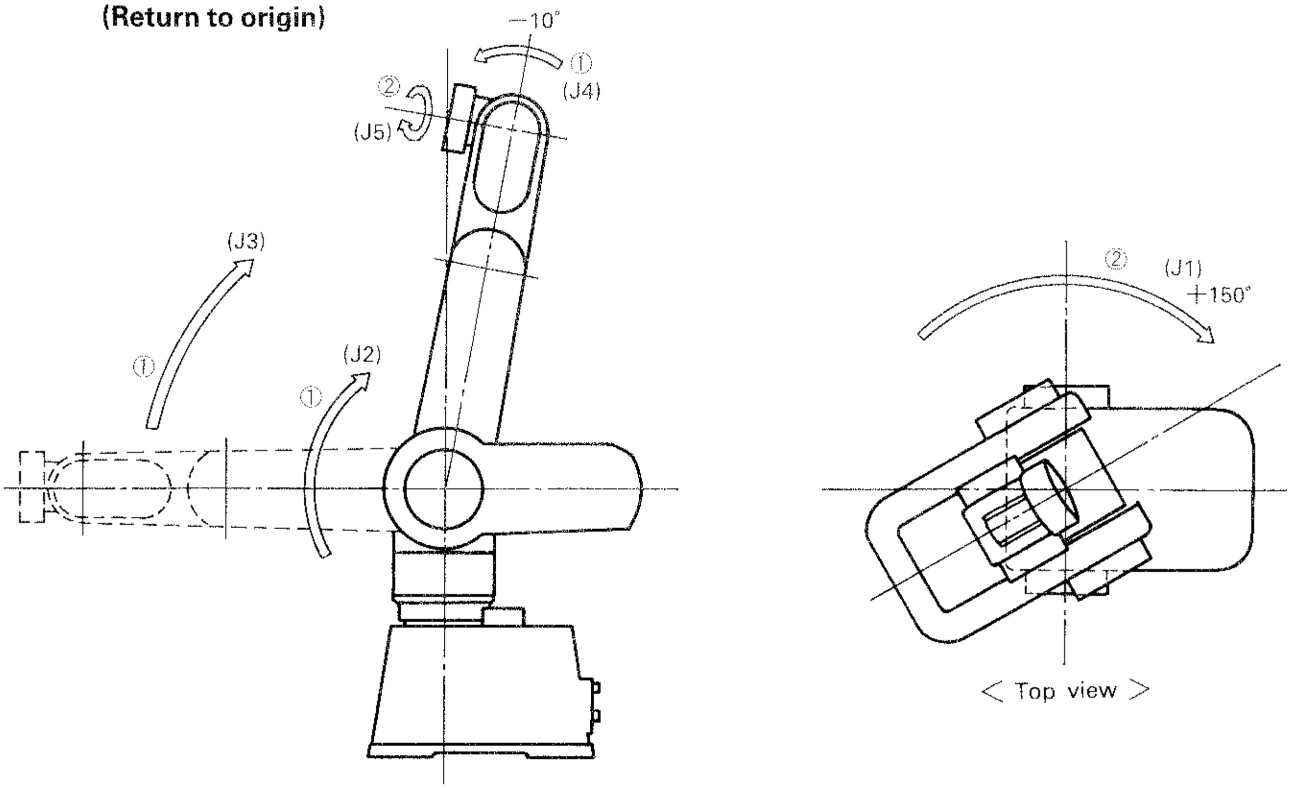


Fig. 1.3.7 Robot Zeroing Operation

(1) The robot of the Movemaster EX RV-M1 must be returned to origin after power on. (For details, see Section 4.3 Origin Setting, Vol. 2.)

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

(3) Return to origin 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.	Move the waist (J1) in the positive direction. Move the wrist roll (J5) in the positive direction.

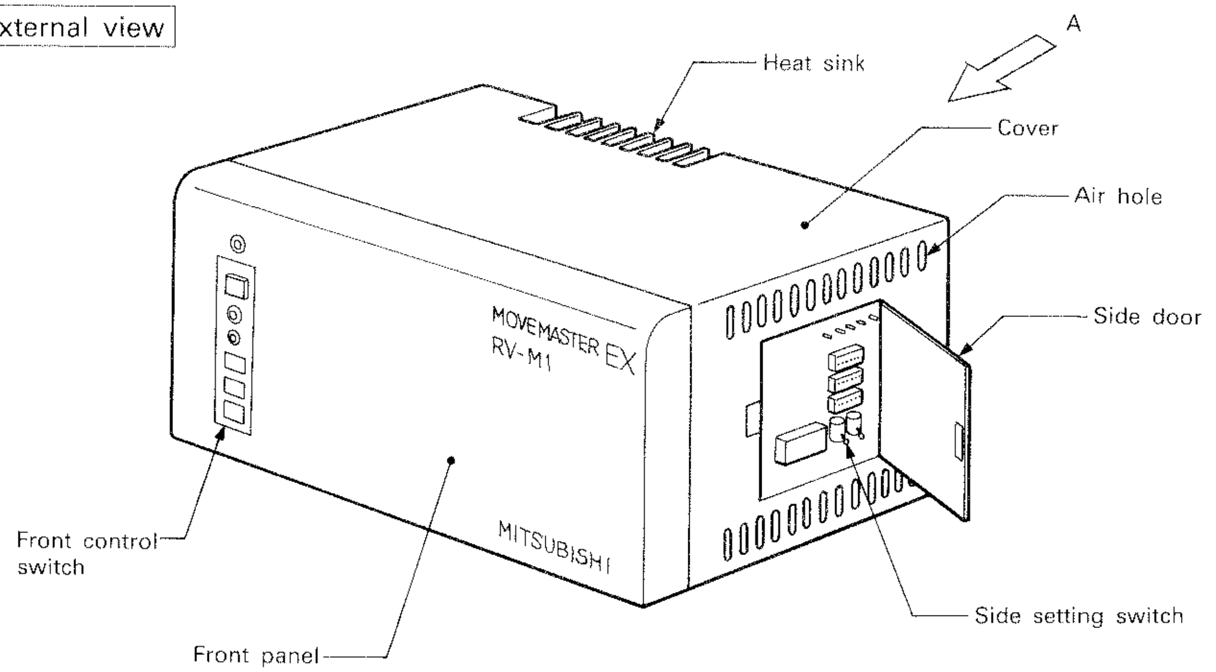
Note 1: All axes in step 1), step 2) must be returned to origin at the same time.

Note 2: The wrist roll (J5) angle is +179.9° after origin set.

3.2 Drive Unit

3.2.1 Nomenclature

External view



Rear view (View A)

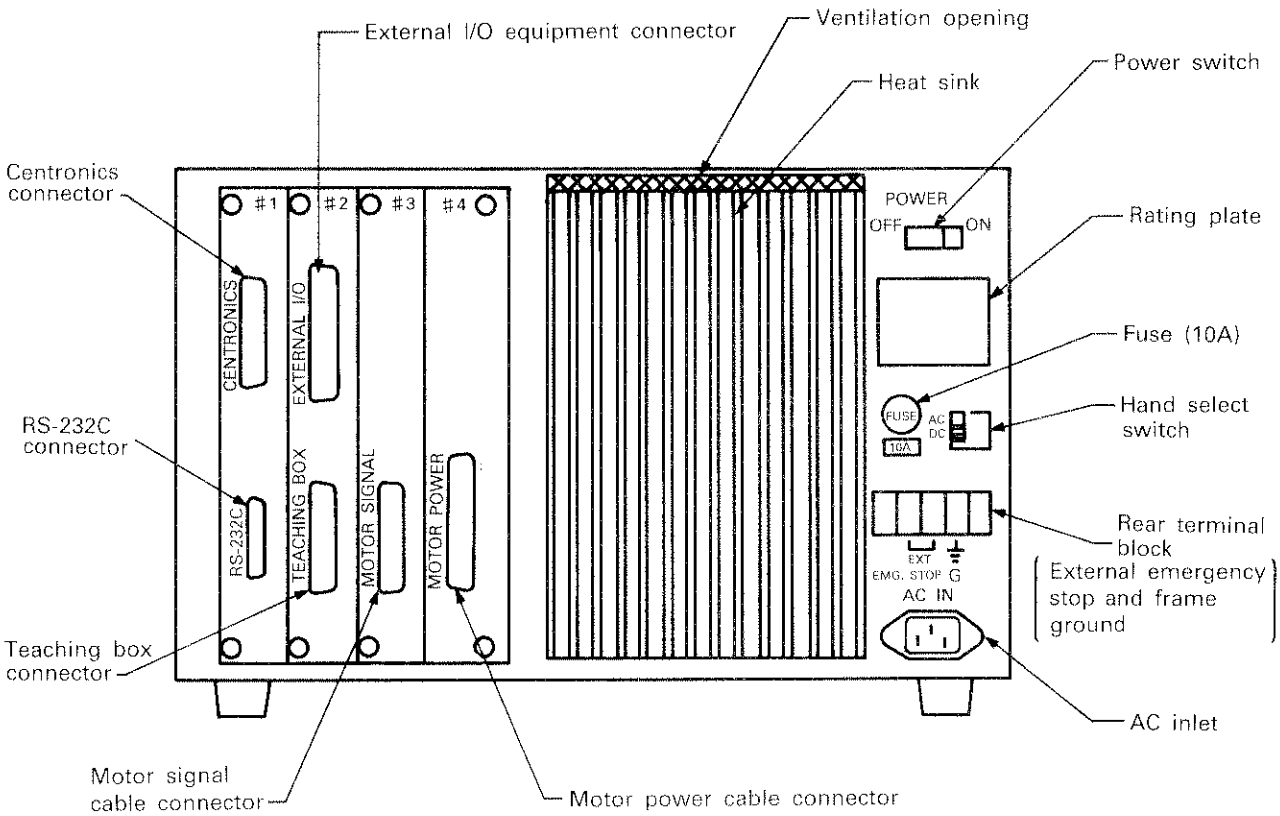


Fig. 1.3.8 Nomenclature (Drive unit)

3.2.2 Standard specifications

Item	Specifications
Teaching method	Programming language system (63 commands), MDI*1 (using a personal computer)
Control method	PTP position control system using DC servo motors
Number of control axes	5 axes (+1 optional axis)
Position detection	Pulse encoder system
Return to Origin Origin setting	Limit switches and pulse encoders (Z phase detection method)
Interpolation function	Articulation interpolation, linear interpolation
Speed setting	10 steps (max. 1000mm/sec)
Number of positions	629 (8KB)
Number of program steps	2048 (16KB)
Data storage	Write to EP-ROM using the built-in EP-ROM writer or storage in the battery-backed static RAM (the battery is optional and backs up the RAM for about 2 years).
Position teaching equipment	Teaching box (option) or personal computer*2
Programming equipment	Personal computer*2
External I/O	General-purpose I/O, 8 points each (16-point type available) General-purpose synchronous signals (STB, BUSY, ACK, RDY) No dedicated I/O (dedicated I/O of 3 points each available) Power for external I/O should be prepared by the user (12V to 24V DC)
Interface	1 parallel interface (conforming to Centronics) 1 serial interface (conforming to RS-232C)
Emergency stop	Using any of the front control switch, teaching box switch, and rear terminal block (N/C contact terminal)
Hand control	Motor-operated hand or pneumatically-operated hand (using AC solenoid)
Brake control	J2 axis (shoulder), J3 axis (elbow)
Power source	120V/220V/230V/240V AC ±10%, 0.5KVA *3
Ambient temperature	5°C to 40°C
Humidity	10 to 85%, non-condensing.
Weight	Approx. 23kgf
Size	380 (W) x 331 (D) x 246 (H) mm

Table 1.3.2 Drive Unit Standard Specifications

Note 1: MDI stands for Manual Data Input

Note 2: To be prepared by the user.

Note 3: Depending on the source power voltage in your country.

3.2.3 External dimensions

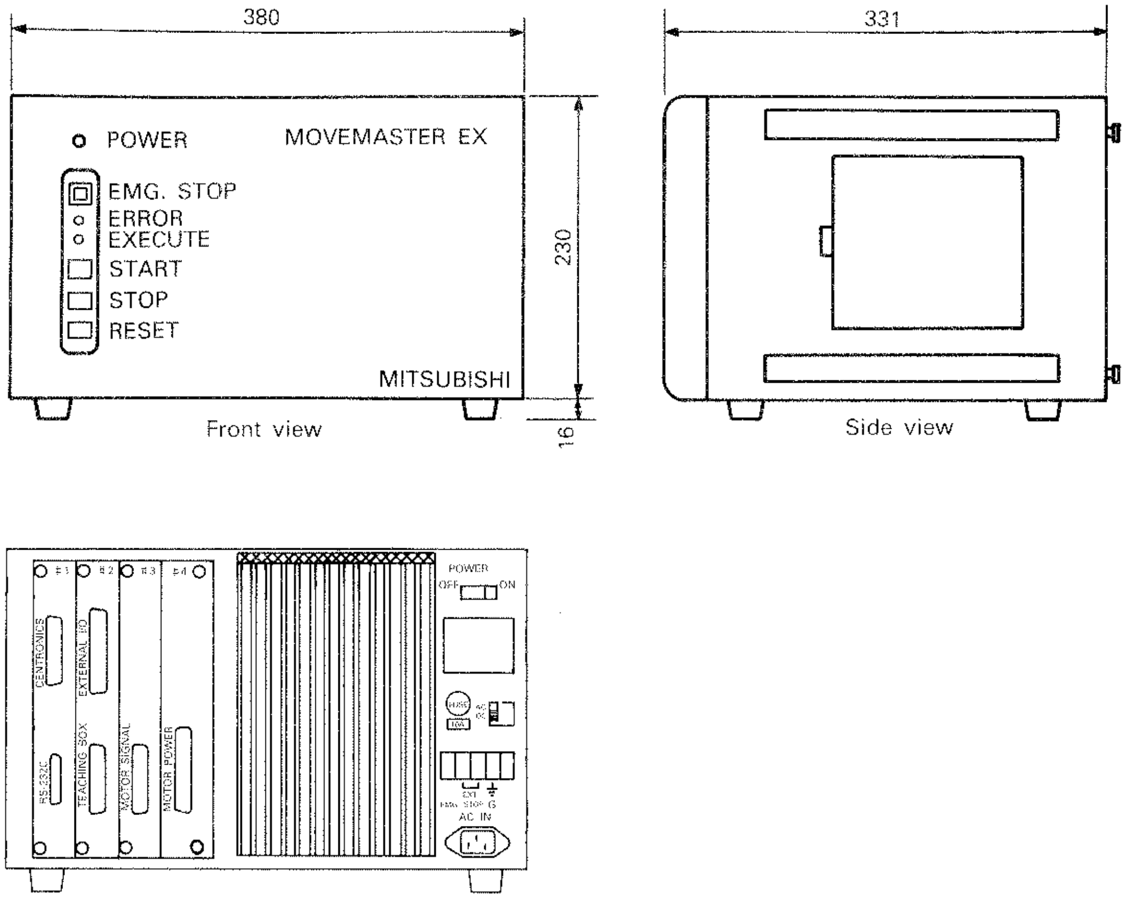


Fig. 1.3.9 External Dimensions (Drive unit)

3.3 Teaching Box (Option)

3.3.1 Nomenclature

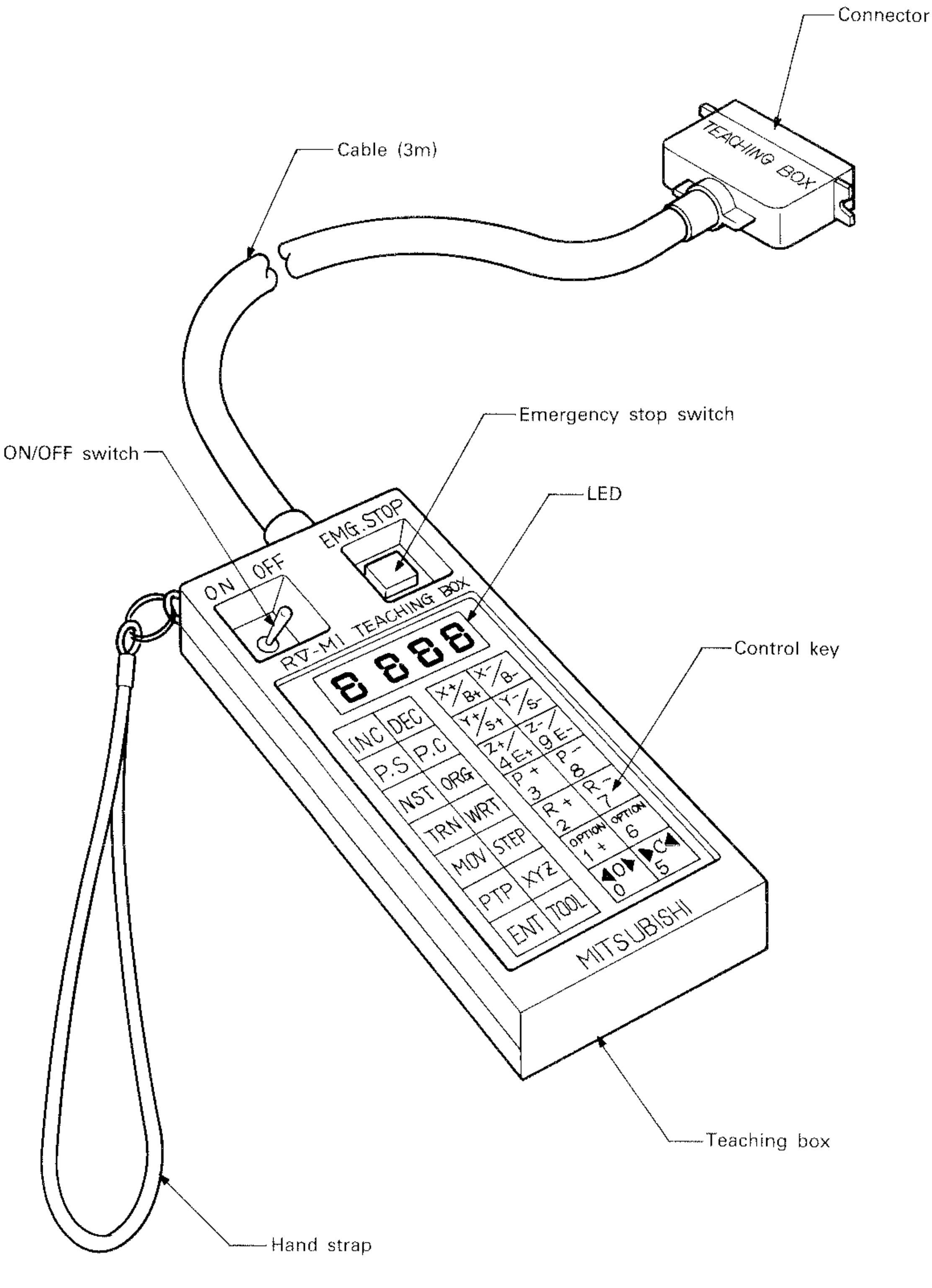


Fig. 1.3.10 Nomenclature (Teaching box)

3.3.2 External dimensions

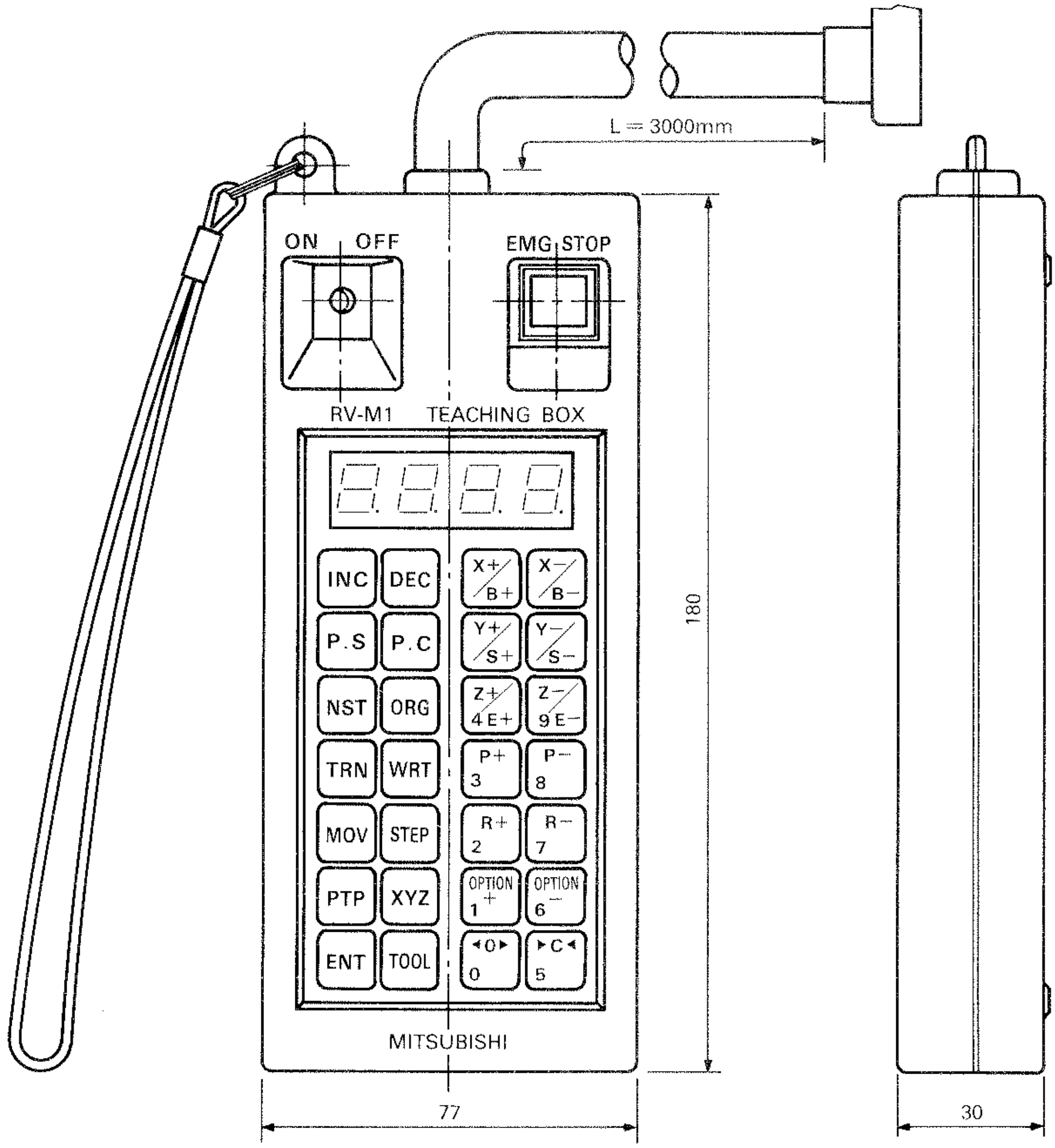


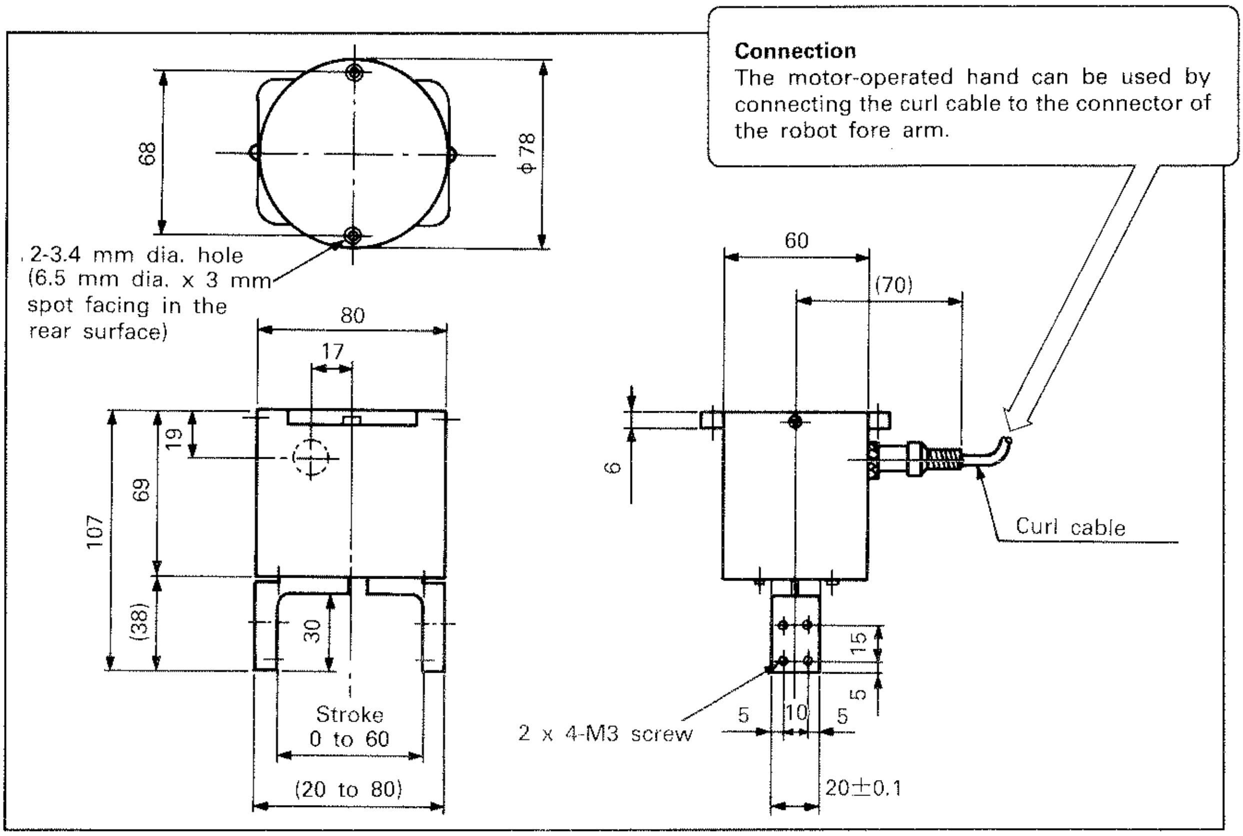
Fig. 1.3.11 External Dimensions (Teaching box)

3.4 Motor-operated Hand (Option)

The motor-operated hand for the RV-M1 can be easily installed to the robot and allows the holding power to be set by current control.

Item	Specifications	Remarks
Type	HM-01	The gripping force can be set in 12 steps. 4≦starting/retained gripping force≦15 (see command GP.)
Drive system	DC 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



4. USING INSTRUCTIONS

4.1 Safety

(1) Do not touch any moving part of the robot during operation. Any work within the operation ranges must be done after switching off the power.

(2) Several emergency stop switches should be installed as required.

(3) Do not allow the robot to make contact with any person (e.g. set up a fence around the robot).

(4) Signals corresponding to the robot operating states from the drive unit I/O connector may be used as appropriate to indicate the operation status of a line, etc. (Dedicated output signal available for the option I/O-A16 or B16)

(5) Ground the Movemaster to prevent noise and electric shock.

4.2 Operating Environments

(1) Use the robot and drive unit within the allowed temperature range (between 5°C and 40°C).

(2) Install the Movemaster away from the direct sunlight and any heat source.

(3) Do not allow water, liquid and metal shavings to enter the robot and drive unit.

(4) Use the Movemaster anywhere that there are no explosive gases, dust, dirt, oil mist, etc.
Do not use this robot in atmospheres contsining explosive gas, dust or mist.

(5) Protect the robot against impact and vibration.

(6) Keep the robot and drive unit connectors clean. Wipe the conductive areas with alcohol to protect from dirt, dust, etc.

4.3 Robot

(1) Protect the plastic covers from damage.

(2) Inspection must be done daily and periodically. If any noise is generated or unusual operation performed, immediately stop the robot and make the required check and adjustment according to Vol. 4 "MAINTENANCE AND INSPECTION PROCEDURE."

(3) The drive motor may be overloaded and burned depending on the robot using condition. The robot must not be used under the following conditions:

① The workpiece is extremely large. (The center of gravity of the workpiece is away from the hand mounting flange surface.)

② Load over the drive motor rating (e.g. forcing, pressure application) is generated continuously.

③ The robot arm is extended for a long time.

④ Acceleration/deceleration is repeated without stopping the given axis.

⑤ The jog operation key is kept pushed with the robot pressed against a mechanical stopper.

(4) Do not operate the robot with the brake applied (shoulder, elbow axes only). (Check that the brake "clicks" in the shoulder cover at power on.)

(5) Fix the robot to a surface plate, etc. using the accessory installation bolts. (Minimum size of the iron plate: 270 x 350 x 12 mm)

(6) Before transporting the robot, the supplied arm fixing plate must be installed in accordance with "TRANSPORTATION, INSTALLATION AND SETTING-UP" in Vol. 2.

(7) When the robot is used on a moving table, e.g. automated guided vehicle, the starting or stopping shock should be 0.5G max.

4.4 Hand

(1) The motor-operated hand loses the grip power during power off.

(2) A less holding power is provided by extremely long fingers.

4.5 Line Voltage

(1) The line voltage must be within ±10% of the rating.

(2) The robot is reset or placed in an error state and is then brought to a stop when instantaneous power failure continues more than 20ms. In this case, restart the operation beginning with the step after power on.

(3) It is recommended to use the memory backup battery (option) to retain data during any power failure.

4.6 Noise

(1) A robot fault or dislocation may occur if surge voltage of more than 1000V, 1 μs is applied to the power line.

(2) Use the Movemaster away from any equipment that generates large noise (e.g. large-sized inverter, high-frequency oscillator, contactor).

(3) Use a radio or television away from the Movemaster. (Concerning electromagnetic interference, the Movemaster conforms to class A in FCC, U.S.A.)

(4) In hostile environments, disconnect the teaching box when not in use so that external noise may not cause a robot fault.

4.7 Instructions for Unpacking and Secondary Transportation

First, carefully read Section 1.1 "Unpacking Instructions" on page 1-1 in the SPECIFICATIONS Part and Section 2.2.4 "Releasing the brakes" on page 2-23 in the OPERATION Part.
Especially for the secondary transportation of the robot, it is recommanded to reuse the packaging material used for shipment from the manufacturer. If the robot cannot be repackaged (e.g. as it has been incorporated in a unit), the robot must be set to the packaging attitude indicated in the instruction manual and the arm fixing plate must be installed to the first arm and body.

> The Movemaster EX RV-M1 has electromagnetic brakes on the J2 (shoulder) axis and J3 (elbow) axis to prevent the arm from dropping when the power is switched off.
> For this reason, the robot comprises a torsional vibration system of 1 degree of freedom which uses the harmonic drive reduction gear output shafts of the J2 and J3 axes as a center. Hence, a large external force will be applied to the harmonic drive reduction gear if the robot is transported without the arm fixing plate for the J2 axis. Especially the harmonic drive reduction gear of the J2 axis may be damaged because the external force generated by acceleration during transportation is greater than the permissible torque of the harmonic drive reduction gear.
> To prevent such an accident, the arm fixing plate for the J2 axis must be installed and the specified packaging material used.

4.8 Operating Instructions

4.8.1 Operation before origin setting

Before origin setting, jog operation must be performed while checking the motion of the robot.
Especially, each arm of the robot must not be left pressed against the mechanical stopper for more than three seconds. The DC servo motor and transistor may burn out if the robot is left in that state longer.

> Each axis of the RV-M1 has only one limit switch which tells the absolute positions of the robot itself. Hence, the movable range cannot be detected before origin setting. After origin setting, the movable range may be monitored by the software which is referenced from the limit switches and phase Z of the encoder.

4.8.2 Instructions for usability

Any incorrect use of the robot will lead to an accident. For instance, the harmonic drive reduction gear of each joint axis will be damaged or the DC servo motor and transistor will burn out if the robot hand does not reach the position taught and pulses remain accumulated due to incorrect operation during teaching or due to insufficiency of interlock or pressing work with more than the permissible load during automatic operation.

4.9 Position Repeatability

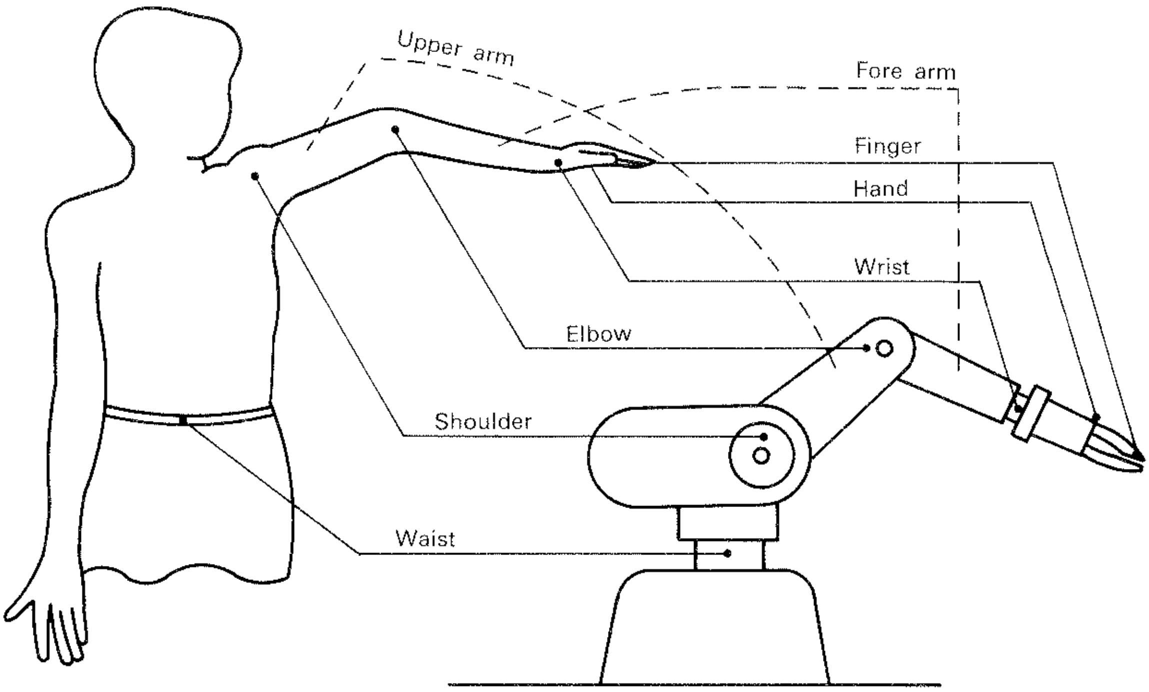
The Position repeatability of the Movemaster robot is based on the following industrial robot definition in the Japanese Industrial Standards. (JIS)

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, and to the numerically set coordinates and to the positioning precision, etc.

Movemaster Information (1)

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



4.10 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. 1.3.13 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. 1.3.14 defines the weight capacity of the robot use with the motor-operated hand (option).

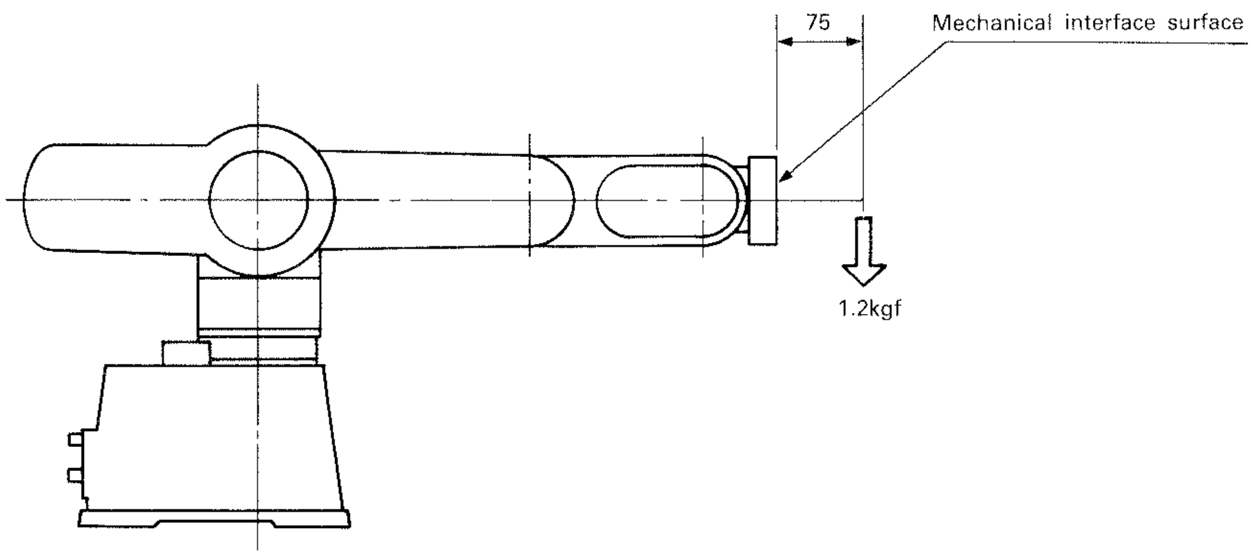


Fig. 1.3.13 Definition of Weight Capacity

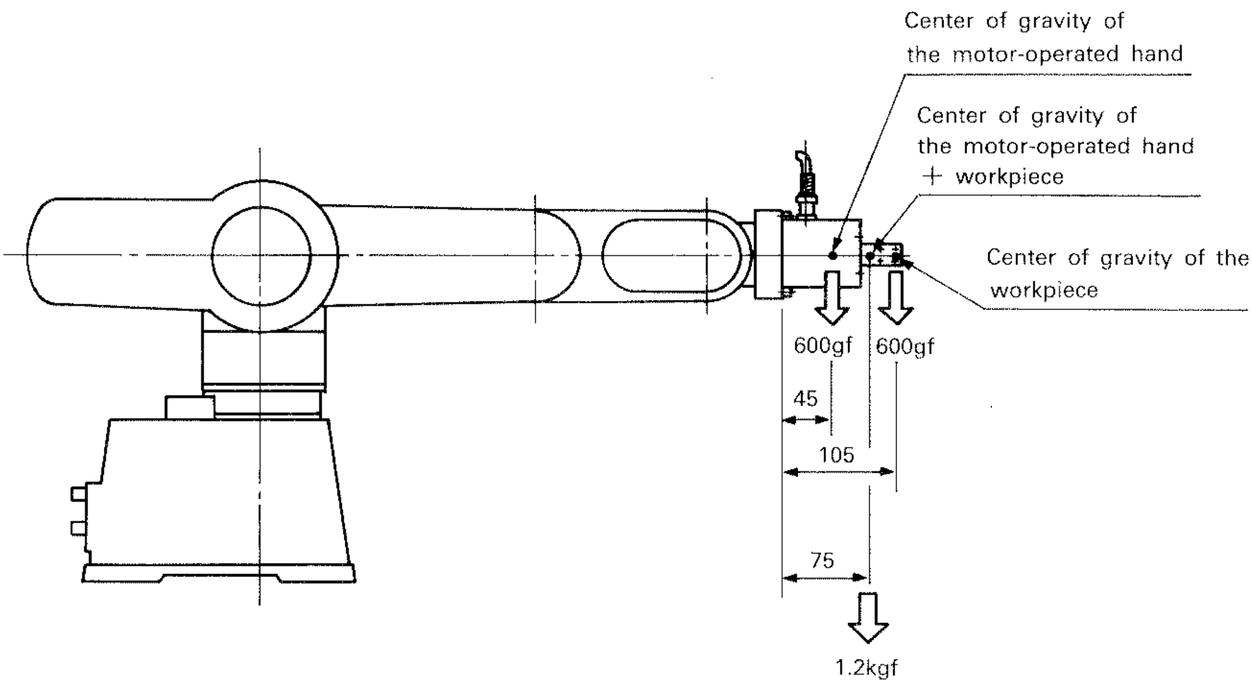


Fig. 1.3.14 Definition of Weight Capacity of the Robot Used with Motor-Operated Hand (Option)

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 x 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

WARNING

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

(3) Any warranty card without the delivery data 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
②Robot base ····················· Munsell 10Y3/1 70% gloss
③Drive unit ······················· Munsell 10Y8/1 70% gloss
④Teaching box ··················· Munsell 5.2G2.5/0.2
⑤Motor-operated hand ········ Munsell N2.5 metallic 70% gloss

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

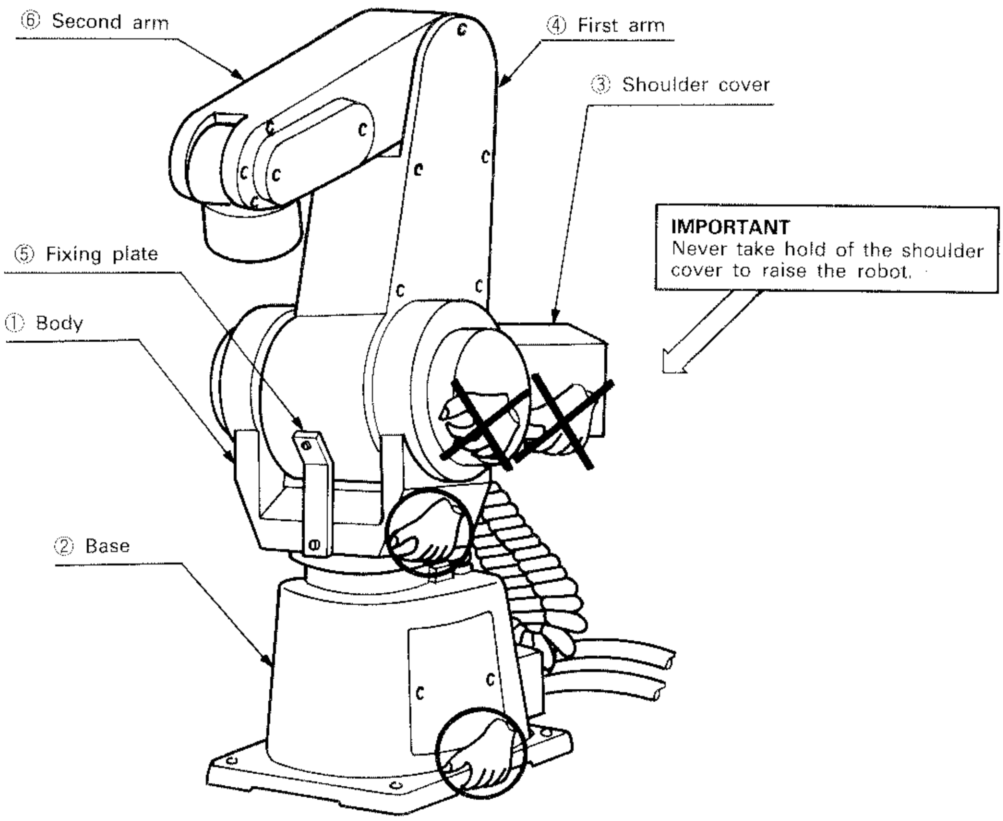


Fig. 2.1.1 Transportation of the Robot

(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 first arm (4) in the positive direction and the second arm (6) in the negative direction until they are pressed against the stoppers as shown in Fig. 2.1.1, and install the arm fixing plate (5) to the first arm (4) and body (1).

(3) When raising the robot, take hold of the underside of body (1) or base (2) with your both hands.

(4) Never take hold the sides or back of shoulder cover (3), or damage to the cover or personal accident could result.

(5) The fixing plate (5) must be reinstalled before transportation to protect the arm (4) (6) during transportation. The robot must be transported in the attitude shown in Fig. 2.1.1.

(6) The above paragraph (2) must be followed when the robot is installed on the other equipment and transported together with that equipment.

WARNING

The robot must be transported with the fixing plate installed.

1.2 Installation of the Robot

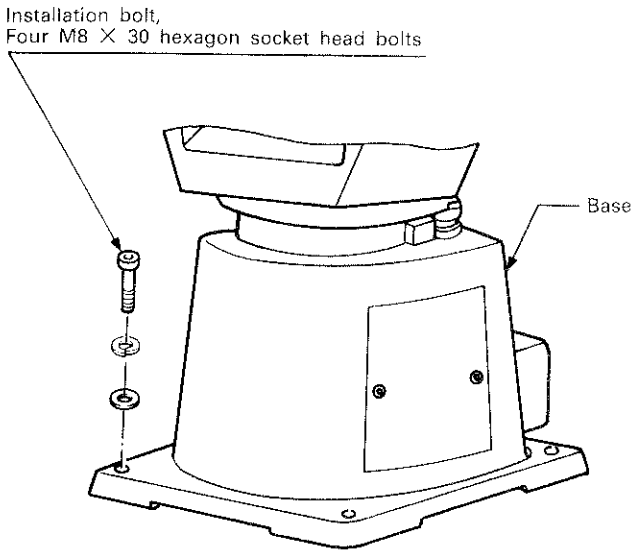


Fig. 2.1.2 Installation of the Robot

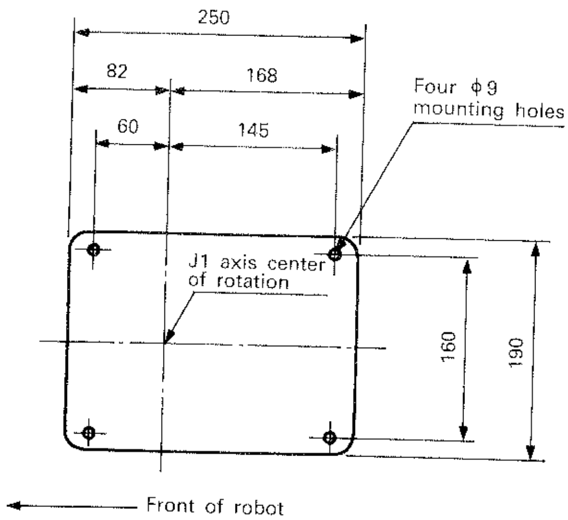


Fig. 2.1.3 Robot Installed Dimensions

(1) Fig. 2.1.2 shows how the robot is installed.

(2) The robot mounting surface has been machined for accurate installation. Using the ϕ9 mounting holes provided at four corners of the base and the installation bolts (M8 × 30 hexagon socket head bolts) furnished, secure the robot onto the floor.

(3) 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 works.

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

(5) The fixing plate (4) is installed before shipment as shown in Fig. 2.1.1 to protect the arm (5) during transportation. Remove this plate before operating the robot.

(6) The customer should keep the fixing plate as it must be used when transporting the robot.

(7) Fig. 2.1.3 shows the installed dimensions of the robot.

(8) Remove the arm fixing plate after the installation is complete. When the robot is transported on a surface plate, this plate must be kept fit on the robot until the robot is finally installed.

WARNING

The customer should keep the arm fixing plate together with the bolts as it must be used when transporting the robot.

1.3 Transportation and Installation of the Drive Unit

(1) Use care not to subject the drive unit to excessive shocks and vibrations and not to let it topple over or tilt during transportation.

(2) Wherever feasible, keep level the drive unit when installed. The tilting angle should be kept within 20°.

(3) Keep the ventilation opening in the drive unit open to 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 installed in a cabinet, make sure that it is well ventilated for ample heat dissipation.

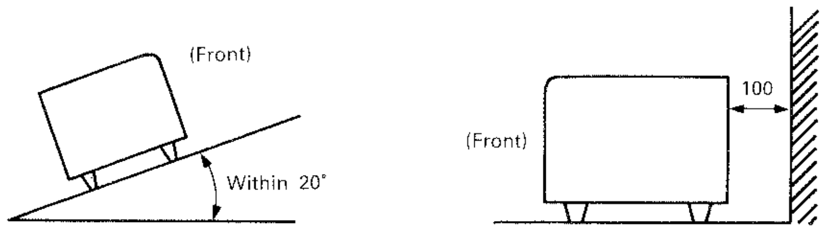


Fig. 2.1.4 Installation of the Drive Unit

1.4 Installation of the I/O Card

Insert your I/O card into the empty slot (the second one from the left) in the rear of the drive unit. At this time, ensure that the card fits in the upper and lower guide rails provided in the slot, slide the card all the way into the slot, and secure it by tightening the upper and lower screws. Fig. 2.1.5 shows the installation procedure. The type of your I/O card is identified by a unique symbol as follows: A8, #2A; B8, #2B; A16, #2C; and, B16, #2D. The symbol is silk-screen-printed on the sheetmetal of each card. (Set bit 3 of the side panel switch SW1 in accordance with the I/O card used. See Section 2.1.2.)

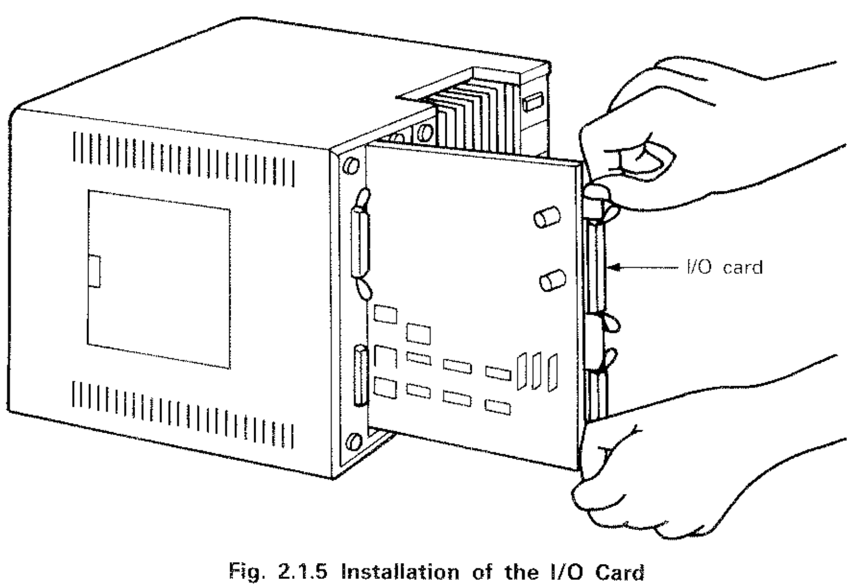


Fig. 2.1.5 Installation of the I/O Card

1.5 Grounding

(1) Wherever feasible, provide the separate grounding dedicated, respectively, to the robot and the drive unit. 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 2-mm^2-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.

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

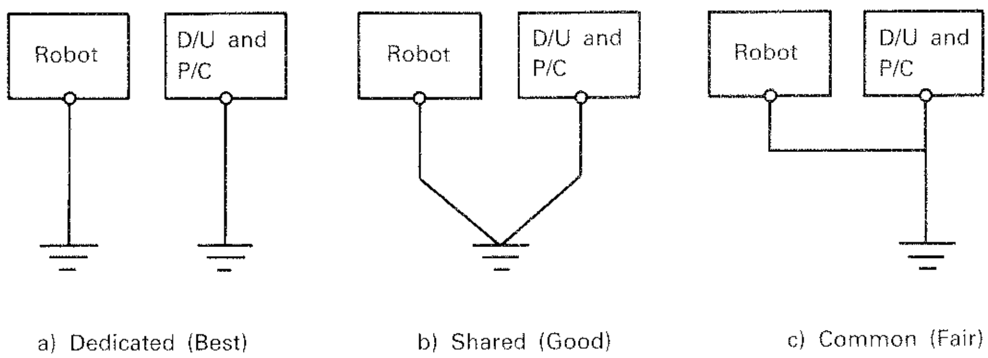


Fig. 2.1.6 Grounding Methods

1.6 Cable Connections

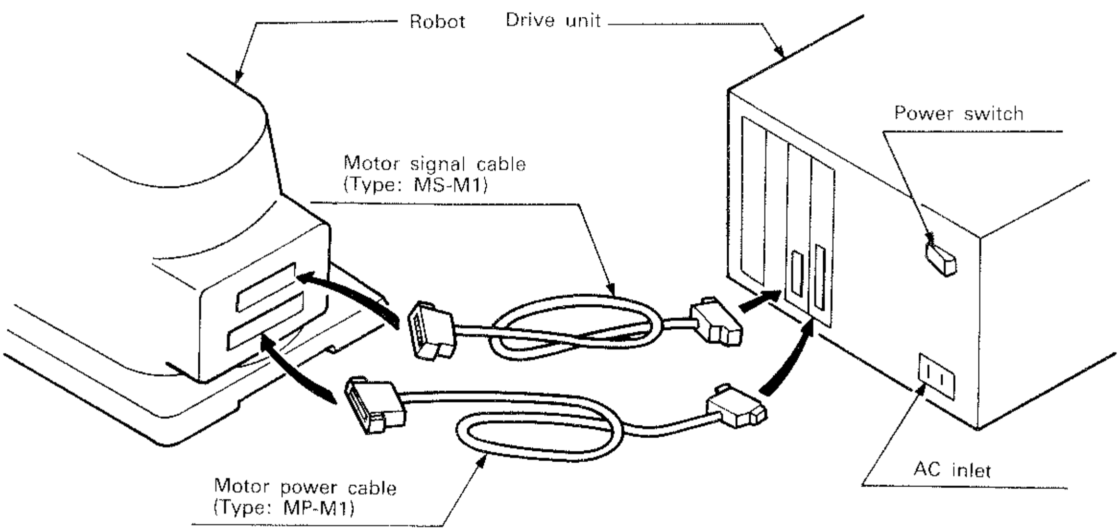


Fig. 2.1.7

(1) Fig. 2.1.7 shows the cable connections.

(2) Before attempting to connect cables, make sure that the power switch on the rear panel of the drive unit is in the OFF position.

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

(4) Connect the motor signal cable and motor power cable to the corresponding connectors in 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 securing the connector, be sure to raise the spring latches on both sides of the connector as illustrated in Fig. 2.1.8.

(7) Use special care when routing the cables. Do not pull hard or bend the cables, as an open-circuited cable or damaged connector could result.

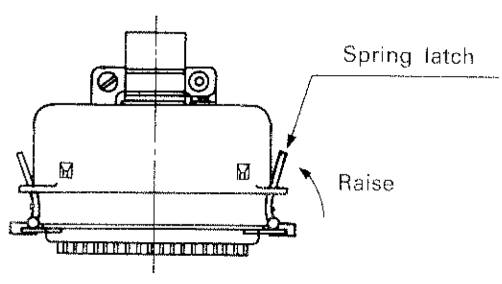


Fig. 2.1.8 Securing the Connector

1.7 Installation of the Hand (Option)

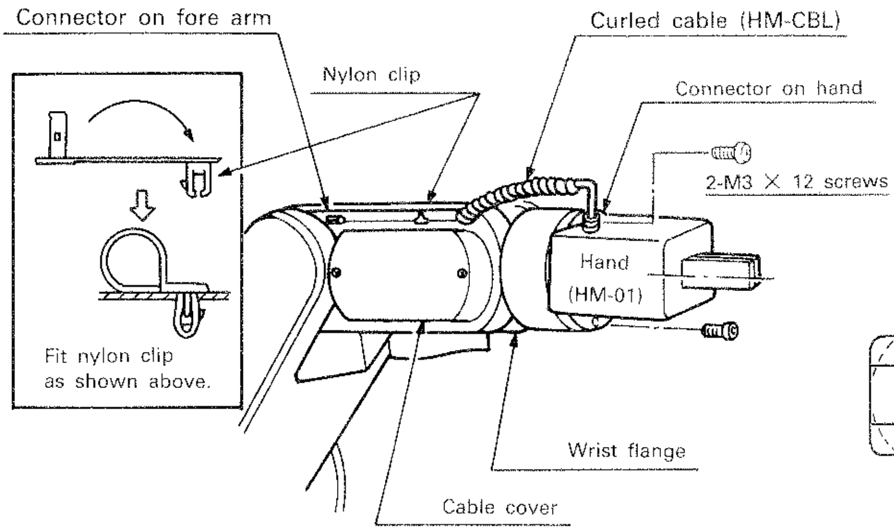


Fig. 2.1.9 Installation of the Hand (Option)

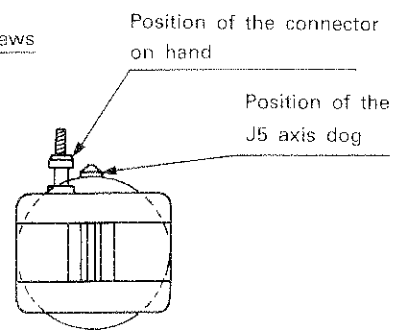


Fig. 2.1.10 Hand Position When Installed

(1) Fig. 2.1.9 shows how the motor-operated hand (option) is to be installed.

(2) Fig. 2.1.10 shows the position of the hand with reference to that of the hand mounding surface where the hand is installed. (For installation, use two M3 x 12 hexagon socket head bolts.)

(3) After the hand has been installed, secure the curled cable onto the connector on the hand and that on the fore arm as shown in Fig. 2.1.9. Using the nylon clip furnished with the hand, secure the straight section of the curled cable to the cable cover.

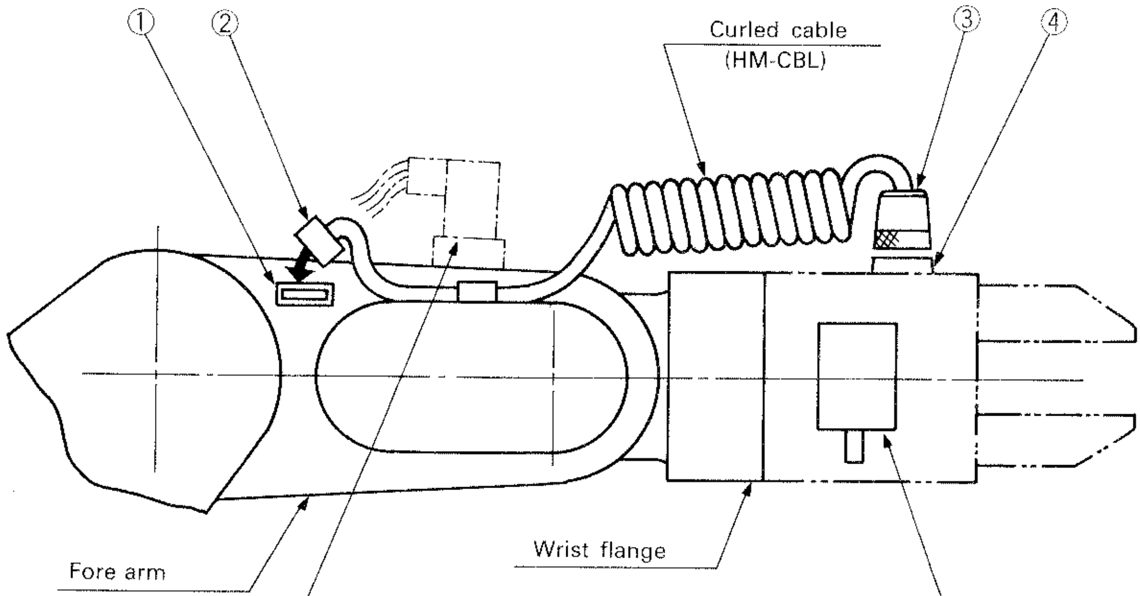
(4) If the hand is customized by the customer, make sure that the hand weighs 1.2 kgf or less including the workpiece to be handled. (At this time, ensure that the center of gravity is located 75mm or lower than the hand mounting surface.)

(5) Table 2.1.1 lists the specifications of the motor, solenoid valve, and connectors for the custom-built hand.

WARNING

The curled cable may get caught between wrist housing and fore arm during operation of the robot equipped with a hand, depending on the position of wrist pitch (J4 axis) and wrist roll (J5 axis). Avoid using with the robot in such a position.

Key No.	Description	Connector		Connector pin	Manufacturer
		Name	Type	Type	
①	Connector on fore arm	SM connector	SMP-02V-BC	SHF-001T-0.8SS	JAPAN SOLDERLESS TERMINAL MFG. CO., LTD.
②	Cable connector (on arm side)		SMR-02V-BC	SYM-001T-0.6	
③	Cable connector (on hand side)	SR13 connector	SR13-10R-2P (01)	Furnished with connector	HIROSE ELECTRIC CO., LTD.
④	Connector on hand		SR13-10P-2S (01)	Furnished with connector	



*Specifications for custom-built hand

Pneumatically-operated hand

Drive source: AC solenoid valve
Power requirements: 120, 220, 230, 240VAC
(Depending on source voltage in the country)
Allowable current: Max. 0.5A (120V AC),
0.25A (220 to 240V AC)

NOTE:
- Install valve on top of forearm and connect line wire to connector on forearm (1).
- The load capacity will be reduced by the weight of the valve.

Motor-operated hand

Drive source: DC motor
Power requirement: 24VDC
Capacity: 6.5W

NOTE:
- Model 106-3001 (SM26) motor manufactured by Sanyo Electric is recommended.

WARNING
The hand select switch located on the rear panel of the drive unit must be set properly according to the type of hand to be used (AC or DC). Wrong setting can be a cause of a burnt drive source or internal transistor. When AC has been selected, 100V AC is supplied from the forearm connector (① in the above figure).

Table 2.1.1 Specifications for the Motor, Solenoid Valve, and Connectors for Hand

1.8 Installation of the Teaching Box (Option)

When using the teaching box (option), be sure to plug the teaching box cable connector into the mating connector on the rear panel of the drive unit (the lower connector on the I/O card) before turning power ON. At this time, do not forget to raise the spring latches to secure the connector into position.
Whenever the connector is plugged and unplugged, make sure that the drive unit is OFF.

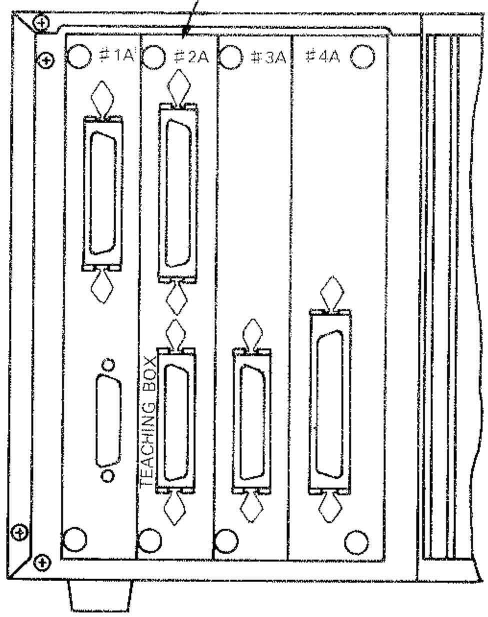


Fig. 2.1.11 Connecting the Teaching Box

1.9 Installation of the Backup Battery (Option)

To install the memory backup battery (option), proceed as follows. It is suggested that important programs and position data be stored on EPROM to protect them from accidental erasure.

(1) After ensuring that the power is OFF, loosen the upper and lower screws on the card in the first slot from the left in the rear of the drive unit. Then, slide the card all the way out.

(2) Pass the Insulock Tie (Teflon band) furnished into the mounting hole in the empty area located on the right bottom of the card and secure the battery in position as shown in Fig. 2.1.12.

(3) Pick up the both ends of the battery socket and plug the socket into the connector (PS1) on the card. At this time, make sure that the side of the socket where the electrode is visible faces down (the card surface) to ensure correct polarity.

(4) After the above procedures have been complete, reinsert the card into the first slot from the left and secure it by tightening the upper and lower screws.

WARNING

Before the drive unit is first turned ON after the battery has been installed, be sure to flip down (to the OFF position) bit 2 of SW1 located inside the side door of the unit.
After the power has been turned ON, flip up (to the ON position) bit 2.

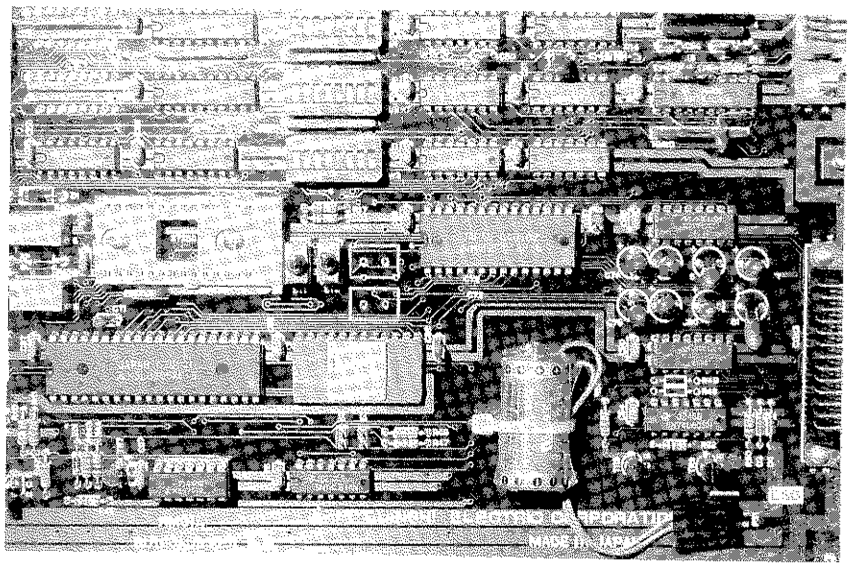


Fig. 2.1.12.

1.10 Installation of the Emergency Stop Switch

From a safety viewpoint, be sure to install the emergency stop switch at a location which is readily accessible.

(1) Use the "EXT. EMG. STOP" terminals on the rear panel of the drive unit for connection.

(2) Remove the strap furnished and connect the COM terminal and NC terminal (normally-closed) of the emergency stop switch to the terminals. Use care not to lose the strap.

(3) Electrical ratings for the emergency stop are 12V DC, 25mA. Select wire that offers sufficient strength and good resistance to environmental changes.

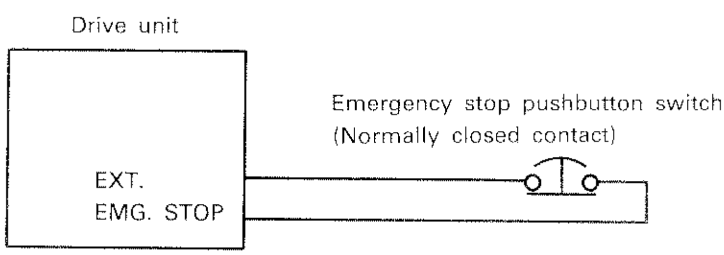


Fig. 2.1.13 Installation of Emergency Stop Switch

1.11 External Emergency Stop Circuit

In addition to the emergency stop switch installed to the emergency stop terminal on the rear panel of the drive unit, the robot can be emergency stopped by the following circuit which is located outside the drive unit and shuts off the power supplied to the drive unit.

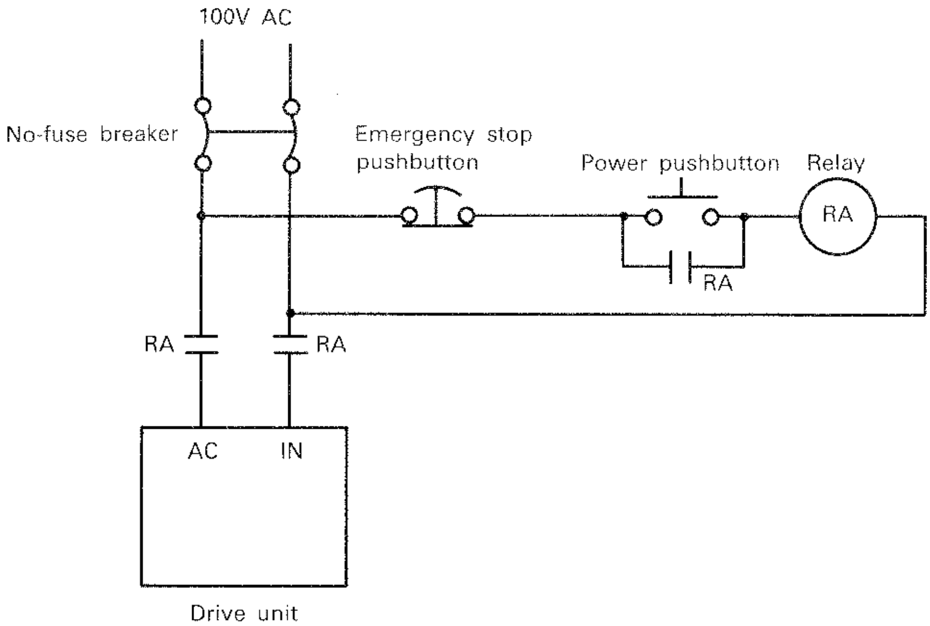


Fig. 2.1.14 External Emergency Stop Circuit

When the emergency stop button in the above circuit has been pressed, no error signal is output even if type A16 or B16 I/O card is used.

2. BASIC FUNCTIONS OF SYSTEM COMPONENTS

2.1 Drive Unit

2.1.1 Functions of front control swtiches and LEDs

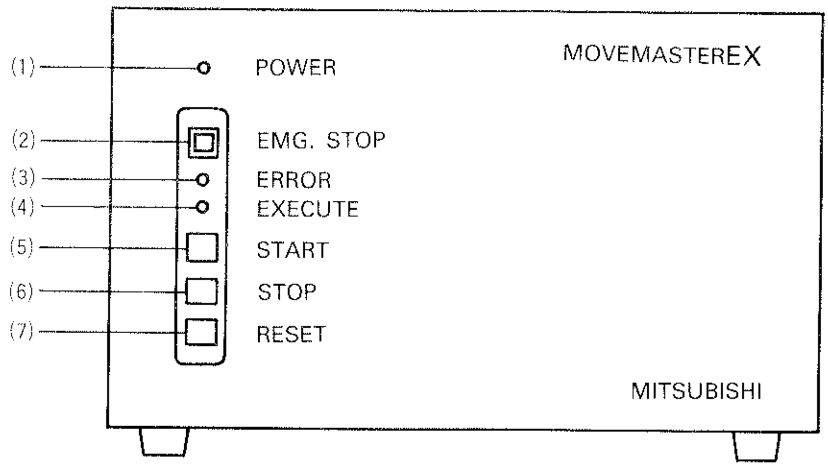


Fig. 2.2.1 Drive Unit Front Panel

(1) POWER (Power indicator LED, yellow)
Lights up when the drive unit is turned ON. It is OFF when the fuse in the rear panel is blown.

(2) EMG. STOP (Emergency stop switch, red)
The emergency stop pushbutton switch. When this switch is pressed, the servo system is cut off and the brakes are applied, thus bringing the robot to an immediate stop. At the same time, the error indicator LED (3) flashes OFF and ON with LED3 located inside the drive unit side door lighting up.

(3) ERROR (Error indicator LED, red)
Blinks or lights up to warn that an error has occurred in the system. In error mode Ⅰ, it blinks at 0.5-second intervals, while in error mode Ⅱ, it lights up and stays lit. The buzzer sounds in phase with the ON timing of this LED if it has been set to ON (bit 8 of SW1 inside the side door in the upper position).

(4) EXECUTE (Command execution indicator LED, green)
Lights up and stays lit while a command is being executed by operaing the drive unit or teaching box. It goes out when the execution has been completed. It stay lit while a program is running.

(5) START (Start switch, green)
Starts the program or restarts it from an suspended state.

(6) STOP (Stop switch, red)
Stops the program being executed. At this time, the robot completes executing the current command before it stops. This means that, if the robot is executing a move command when the switch is pressed, it will complete reaching the destination point.

(7) RESET (Reset switch, white)
Resets the program suspended by the deperssion of the stop switch and error mode II. When the switch is pressed, the program returns to its beginning and, if an error has occurred, the error indicator LED goes out. Note, however, that the general-purpose I/O outputs are not reset at this time.

2.1.2 Functions of side setting switches and LEDs

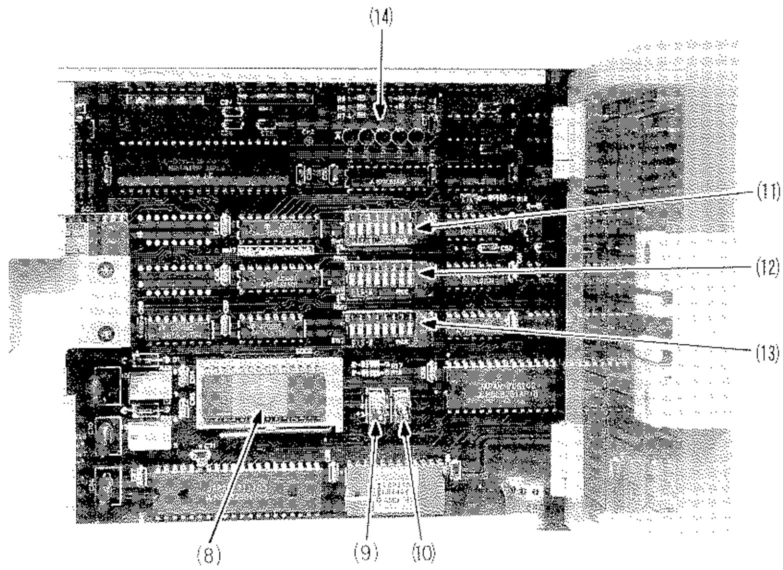


Fig. 2.2.2 Drive Unit Side Panel

(8) SOC2 (EPROM socket for program and position data)
A user socket for installation of an EPROM into which the program and position data have been written. The EPROM can be inserted into, or removed from, the socket by just operating the lever provided under the socket. When inserting the EPROM, make sure that the indentation is positioned on the left.

(9) ST1 (Control mode setting switch)
Sets the control mode of the drive unit (upper position: drive unit mode; lower position: personal computer mode).

(10) ST2 (Select switch whether to transfer data from EPROM to RAM upon power-up)
Selects whether to transfer data from EPROM to drive unit memory RAM when the power is turned ON (upper position: EPROM data is transferred to RAM; lower position: EPROM data is not transferred to RAM). The RAM data is used to run a program from the drive unit. It is therefore necessary to previously set this switch to the upper position when running a program by means of data written in EPROM after the power has been turned ON. When a program is to be run by the battery-backed-up RAM, set this switch to the lower position.

(11) SW1 (Function select DIP switch; Bits are 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). Flip this bit down to the lower position unless you use MULTI16.

Bit 2: Selects whether to check if the contents of the RAM data are retained upon power-up (upper position: Check is performed; lower position: Check is not performed). Set this bit to the lower position if the battery (option) is not used.
If the battery is to be used, set the bit to the lower position when the system is first turned ON after the battery has been installed; thereafter, place the bit in the upper position. (The RAM data is cleared by switching on the power with bit 2 at the lower position.)
Then, error mode I is caused when power is turned ON if the contents of memory are not properly retained during power-down due to a battery failure or other cause. (At this time, LED5 inside the drive unit side door lights up.)

Bit 3: Selects the type of I/O card used (upper position: type A16 or B16; lower position: type A8 or B8).

Bit 4: Selects whether or not to set, change, or delete the reference position data in the cartesian coordinate system (upper position: enabled; lower position: disabled). Set this bit to the upper position when setting the cartesian coordinate system reference positions and when loading the reference position data written in the EPROM into the RAM; otherwise, set it to the lower position to avoid erroneous setting.

Bit 5: Selects to enable either the drive unit front control switches or external signals for running the program while type A16 or B16 I/O card is being used (upper position: external signals; lower position: front control switches). When this bit is set to the upper position, operation becomes possible by means of the dedicated signal lines on the external I/O equipment connector on the drive unit rear panel, while the front control switches (except the emergency stop switch) are disabled. With the bit set to the lower position, the front control switches are enabled, while the external dedicated signal line is disabled. Place the bit in the lower position when type A8 or B8 I/O card is to be used.

Bit 6: Selects whether to enable the [ENT] key on the teaching box to release robot brakes. Normally, set this bit to the lower position.
(See 2.2.4 Releasing the brakes.)

Bit 7: Not used.

Bit 8: Selects whether to turn ON or OFF the buzzer (upper position: buzzer sounds when an error occurs; lower position: buzzer does not sound when an error occurs).

(12) SW2 (RS232C communication format setting switch)
See CHAPTER 2 INTERFACE WITH THE PERSONAL COMPUTER (RS232C), APPENDIX.

(13) SW3 (RS232C baud rate setting switch)
See CHAPTER 2 INTERFACE WITH THE PERSONAL COMPUTER (RS232C), APPENDIX.

(14) LEDs 1 — 5 (Hardware error displays)
Indicate the cause of the corresponding hardware error (error mode Ⅰ) when it occurs.

LED1 ······ Excessive servo system errors (1st LED from the left)

LED2 ······ Open or disconnected motor signal cable (2nd LED from the left)

LED3 ······ Drive unit emergency stop input (3rd LED from the left)

LED4 ······ Teaching box emergency stop input (4th LED from the left)

LED5 ······ Backup battery failure (5th LED from the left)

2.1.3 Functions of connectors, switches, and terminal block on rear panel

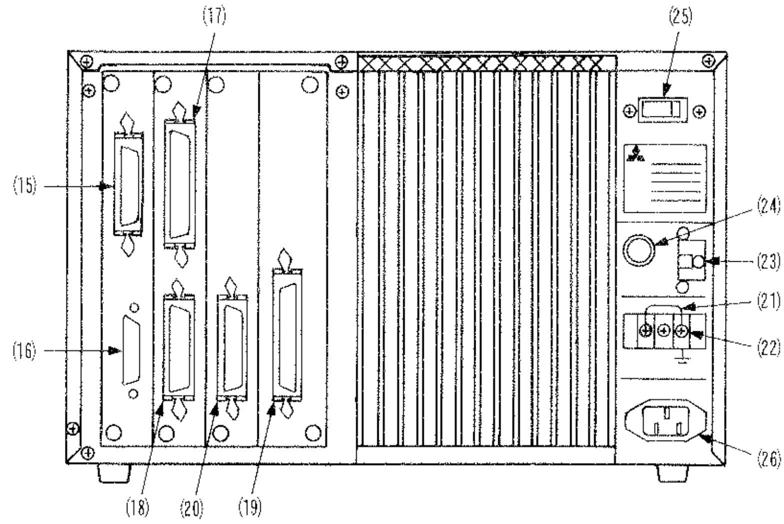


Fig. 2.2.3 Drive Unit Rear Panel

(15) CENTRONICS (Centronics connector)
Serves as the connector for the Centronics interface that connects a personal computer and drive unit.

(16) RS-232C (RS232C connector)
Serves as the connector for the RS232C interface that connects a personal computer and drive unit.

(17) EXTERNAL I/O (External I/O equipment connector)
Serves as the connector for connecting external I/O equipment (limit switches, LEDs, programmable controllers, etc.) and drive unit.

(18) TEACHING BOX (Teaching box connector)
Serves as the connector for connecting the teaching box (option) and drive unit.

(19) MOTOR POWER (Motor power cable connector)
Serves as the connector for the power line between the drive unit and robot.

(20) MOTOR SIGNAL (Motor signal cable connector)
Serves as the connector for the signal line between the drive unit and robot.

(21) EXT EMG. STP (Emergency stop input terminal)
Input terminal for connecting an external emergency stop switch (12V DC, 25mA, N/C contact terminal).

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

(23) HAND AC/DC (Hand select switch)
Selects either DC or AC depending on the type of drive of the hand attached to the robot. Select DC when a motor-operated hand (option) is used; select AC when using pneumatically-operated hand employing AC solenoid valves. Make doubly sure of the correct setting, as wrong setting damages the internal circuitry.

(24) FUSE (Fuse)
Fuse holder containing a fuse (250V AC, 10A) for the drive unit.

(25) POWER (Power switch)
Power ON/OFF switch for the drive unit.

(26) AC (AC inlet)
Inlet into which the drive unit power cord is plugged. (120, 220, 230, or 240V AC depending on the source power voltage in your country)

2.2 Teaching Box

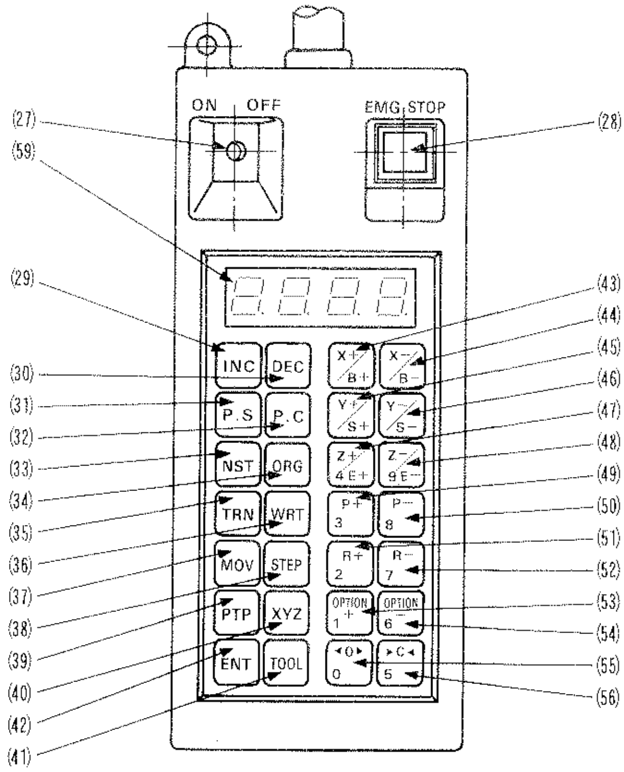


Fig. 2.2.4 Teaching Box

2.2.1 Functions of the switches

(27) ON/OFF (Power switch)
Selects whether to enable or disable keys on the teaching box. When the robot is to be operated using the teaching box, turn this switch ON. During program run or when controlling the robot by means of commands sent from a personal computer, turn the switch OFF. An erroneous key entry can also be cleared by turning the switch OFF. Operation from the teaching box cannot be performed if this switch is set to ON during programmed operation.

(28) EMG. STOP (Emergency stop switch)
Pushbutton switch used for emergency stop of the robot (signal is internally latched when this switch is pressed). When the switch is pressed, the robot is immediately brought to a stop and the error indicator LED blinks (error mode I). LED4 inside the drive unit side door also comes on.

2.2.2 Functions of each key

(29) [INC] (+ [ENT])
Moves the robot to a predefined position with a position number greater than the current one. To move the robot through a certain sequence, repeat the keying-in sequence. (See command "IP.")

(30) [DEC] (+ [ENT])
Moves the robot to a predefined position with a position number smaller than the current one. To move the robot through a certain sequence, repeat the keying-in sequence. (See command "DP.")

(31) [P. S] (+ [Number] + [ENT])
Defines the coordinates of the current position of the robot into a position with the number specified. If a single number is assigned to two different positions, the one defined last takes precedence. (See command "HE.") To prevent an error, do not set the robot to the attitude in which any axis is adjacent to the operation limit.

(32) [P. C] (+ [Number] + [ENT])
Deletes the contents of a position with the number specified. (See command "PC.")

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

(34) [ORG] (+ [ENT])
Moves the robot to the reference position in the cartesian coordinate system. (See command "OG.")

(35) [TRN] (+ [ENT])
Transfers the contents of the user EPROM (program and position data) installed in SOC2 of the drive unit side panel to the drive unit RAM. (See command "TR.")

(36) [WRT] (+ [ENT])
Writes the program and position data written in the drive unit RAM into the user EPROM installed in SOC2 of the drive unit side panel. (See command "CR.")

(37) [MOV] (+ [Number] + [ENT])
Moves the end of the hand to a specified position. (See command "MO.") The moving speed is equivalent to SP4.

(38) [STEP] (+ [Number] + [ENT])
Executes the program step by step starting with the line number specified. To cause the program to be executed sequentially from one step to another, repeat the keying-in sequence. Note that, at this time, no number entry is necessary. Error mode II is caused if an error occurs while the steps are being executed.

(39) [PTP]
Selects the articulated jog operation. When this key is pressed, operation of any jog key thereafter effects a motion in each joint. In the initial condition when the teaching box is turned ON, this PTP state is set.

(40) [XYZ]
Selects the cartesian jog operation. When this key is pressed, operation of any jog key thereafter effects an axis motion in the cartesian coordinate system.

(41) [TOOL]
Selects the tool jog operation. When this key is pressed, operation of any jog key thereafter effects an axis motion in the tool coordinate system (advance/retract motion in the hand direction).

(42) [ENT]
Completes each key entry from (29) through (38) to effect corresponding operation.

(43) [X+/B+]
Moves the end of the hand in positive X-axis (to the left looking toward the front of robot) in the cartesian jog operation and sweeps the waist in the positive direction (clockwise as viewed from the top of robot) in the articulated jog operation.

(44) [X−/B−]
Moves the end of the hand in negative X-axis (to the right looking toward the front of robot) in the cartesian jog and sweeps the waist in the negative direction (counterclockwise as viewed from the top of robot) in the articulated jog.

(45) [Y+/S+]
Moves the end of the hand in positive Y-axis (to the front of the robot) in the cartesian jog and swivels the shoulder in the positive direction (upward) in the articulated jog.

(46) [Y−/S−]
Moves the end of the hand in negative Y-axis (to the rear of the robot) in the cartesian jog and swivels the shoulder in the negative direction (downward) in the articulated jog.

(47) [Z+/E+ 4]
Moves the end of the hand in positive Z-axis (straight upward) in the cartesian jog, turns the elbow in the positive direction (upward) in the articulated jog, and advances the hand in the tool jog. It serves also as the numeric key "4."

(48) [Z−/E− 9]
Moves the end of the hand in negative Z-axis (straight downward) in the cartesian jog, turns the elbow in the negative direction (downward) in the articulated jog, and retracts the hand in the tool jog. It serves also as the numeric key "9."

(49) [P+ 3]
Turns the end of the hand, while maintaining its current position determined by the "TL" command, in the positive direction (upward) in the cartesian jog and bends the wrist (wrist pitch) in the positive direction (upward) in the articulated jog. It also serves as the numeric key "3.".

(50) [P− 8]
Turns the end of the hand, while maintaining its current position determined by the "TL" command, in the negative direction (downward) in the cartesian jog and bends the wrist (wrist pitch) in the negative direction (downward) in the articulated jog. It also serves as the numeric key "8.".

(51) [R+ 2]
Twists the wrist (wrist roll) in the positive direction (clockwise looking toward the hand mounting surface). It also serves as the numeric key "2."

(52) [R− 7]
Twists the wrist (wrist roll) in the negative direction (counterclockwise looking toward the hand mounding surface). It also serves as the numeric key "7."

(53) [OPTION+ 1]
Moves the optional axis in the positive direction. It also serves as the numeric key "1."

(54) [OPTION− 6]
Moves the optional axis in the negative direction. It also serves as the numeric key "6."

(55) [◀O▶ 0]
Opens the hand gripper. It also serves as the numeric key "0."

(56) [▶C◀ 5]
Closes the hand gripper. It also serves as the numeric key "5."

2.2.3 Functions of the indicator LED

The 4-digit LED shows the following information.

(57) Position number
Shows the position number in 3 digits when [INC], [DEC], [P. S], [P. C], or [MOV] key is being used.

(58) Program line number
Shows the program line number in 4 digits when [STEP] key is being used or when program is running.

(59) Teaching box status indicator (the first digit from the left)
"⊔" means processing invoked by depression of ENT key is either in progress or at an end.
"⊏" means processing invoked by depression of ENT key cannot be carried out.

2.2.4 Releasing the brakes

When the system power is OFF or when error mode I occurs, brakes are applied to J2 and J3 axes of the robot. This means that these axes of motion cannot be achieved externally. The following steps may, however, be performed to release the brakes and allow for the axis motion externally. The procedures can be used when servicing the robot or positioning it for packing. (Remember, you must have the teaching box to perform the procedure.)

(1) After power has been turned ON, press the emergency stop pushbutton switch on the front panel of the drive unit to cause error mode I.

(2) Flip up (to ON position) bit 6 of SW1 located inside the side door of the drive unit.

(3) Set the teaching box ON/OFF switch to ON and press the [ENT] key to release brakes in J2 and J3 axes. At this time, be sure to hold the robot arm with your hands so that it will not drop down by its own weight. Note that the brakes are released for J2 and J3 axes at the same time; brakes are reapplied as soon as the [ENT] key is released.

(4) After the required job has been completed, be sure to flip bit 6 of SW1 down (to OFF position).

2.2.5 Intelligent commands corresponding to each key

The functions of the keys on the teaching box correspond to the functions invoked by intelligent commands sent from the personal computer as follows.

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

3. BEFORE STARTING THE OPERATION

This chapter gives the overview of the operation and programming as initiation to the robotic system.

3.1 System Configuration

The Movemaster system may be configured in two different ways. The following paragraphs describe the unique features of, and the basic idea behind, each system configuration.

3.1.1 System configuration centering around a personal computer

This system configures the Movemaster with a personal computer. The personal computer invokes the robot's axis motions by the intelligent commands provided in the Movemaster. In this configuration, the personal computer acts as the brain that causes the robot to perform a variety of tasks including assembly and experimentation.
When configured with a whole range of peripheral equipment, such as the printer, X-Y plotter, external storage, and sensors, the system becomes a highly expandable, enhanced one. 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 the 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 decribed later in this manual.

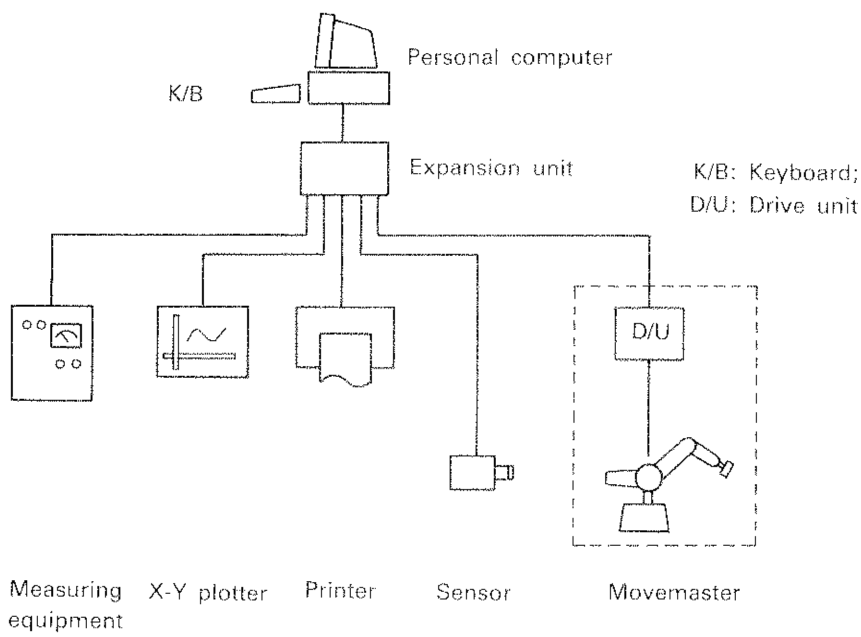


Fig. 2.3.1 Typical System Configuration Centering Around a Personal Computer

3.1.2 System configuration centering around the drive unit

This configuration uses the drive unit to drive the Movemaster and the personal computer is used only for programming purposes. The program written with the personal computer is transferred to the drive unit for later running the robot. This means that you do not need to install a personal computer on the actual production floor. Signal exchange between the robot and peripheral equipment such as limit switches, relays, LEDs, and programmable controllers is accomplished through the external I/O port in the drive unit. The program, which is 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.

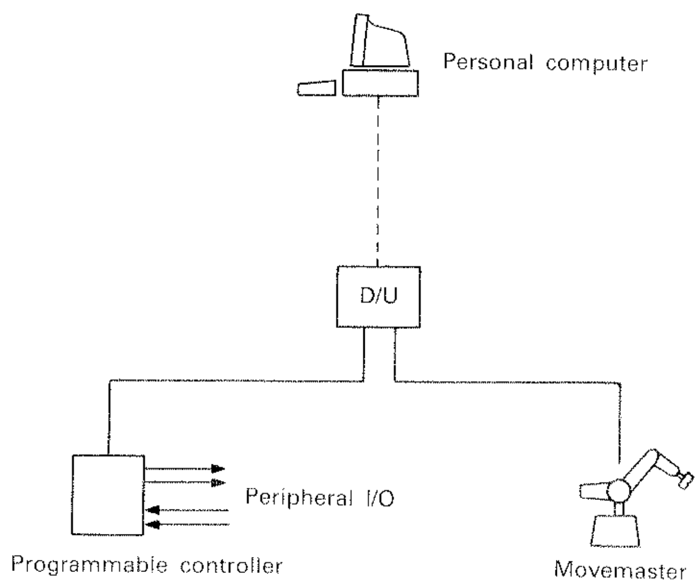


Fig. 2.3.2 Typical System Configuration Centering Around the Drive unit

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 descriptions and use the appropriate interface according to your personal computer and application. For more details, see CHAPTERS 1 and 2 INTERFACE WITH THE PERSONAL COMPUTER, APPENDIX.

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, meaning that 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 are 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

These paragraphs describe the two control modes available with the Movemaster, i.e., the personal computer mode and drive unit mode.

3.3.1 Personal computer mode

[Setting procedure]
Set the toggle switch (ST1) located inside 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 (the mode corresponding to 3.1.1 System configuration 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, be sure to keep the ON/OFF switch of the teaching box 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 code. Lower-case characters are automatically converted into upper-case characters. 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 code; where the number at the beginning, "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 the order of the line number. Be sure, therefore, to assign the line number at the beginning when writing a program. Line numbers are possible from 1 up to 2048. Any line number greater than 2048 leads to 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.

(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".

Now, let us study some of the typical programs.

(Example 1) Direct execution (Centronics)

```
100 LPRINT "NT"            ; Origin setting (nesting).
110 LPRINT "SP 7"          ; Set speed at 7.
120 LPRINT "MO 10, O"      ; Move to position 10 with hand opened.
130 LPRINT "GC"            ; Close hand (grip close).
140 LPRINT "MO 11, C"      ; Move to position 11 with hand closed.
150 END                    ; End BASIC program.
RUN                        ; Run BASIC program.
OK
```

In this example, hitting "RUN" causes each line (numbered 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 robotic motions).

(Example 2) Program generation to execution (Centronics)

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

In this example, hitting "RUN" causes each line (numbered 100 to 160) of the BASIC program to be executed, which in turn causes the Movemaster program (lines numbered 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) located inside the drive unit side door to the upper position.

[Explanation]
This mode allows the program stored in the drive unit EPROM or battery-backed-up RAM to be executed (corresponding to 3.1.2 System configuration centering around the drive unit). In this mode, the front control switches of the drive unit are used to start, stop, and reset the program. If type A16 or B16 I/O card is being used, operation is possible by means of external signals. Any command sent from the personal computer is not honored.

4. FROM POWER-UP TO ORIGIN SETTING

This chapter describes the procedures that must be performed after delivery and installation of the system before initiating the actual robotic operations.

4.1 Setting the Side Setting Switches

Before turning the system power ON, set the side setting switches as follows by referring to 2.1.2 Functions of side setting switches and LEDs.

ST1: Lower position (Personal computer mode)
ST2: Lower position (EPROM data is not transferred to RAM)

4.2 Turning Power ON

Turn ON the power switch located on the rear panel of the drive unit. When the drive unit is turned ON, the power indicator LED (POWER) on the front panel of the drive unit lights up.

4.3 Origin Setting

The robot must be returned to origin as follows after power is turned ON. This is done to match the robot's mechanical origin with the control system's origin and is required only once after the power is turned ON. Here is the motion of each axis:

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

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

Be alert, at this time, that the arm could interfere with the objects surrounding the robot. It must therefore be moved as appropriate to a safe position using the teaching box before attempting to return the robot to origin. The command execution indicator LED (EXECUTE) on the drive unit front panel stays lit during origin setting and goes out as soon as it completes.

Using the teaching box

1) Turn ON the teaching box ON/OFF switch.

2) Press [NST] and [ENT], successively in that order.

Using intelligent commands through personal computer

1) Turn OFF the teaching box ON/OFF switch.

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

5. POSITION SETTING PROCEDURE

This chapter describes the position-setting procedure following the origin-setting procedure.

5.1 Setting the Cartesian Coordinate System Reference Position

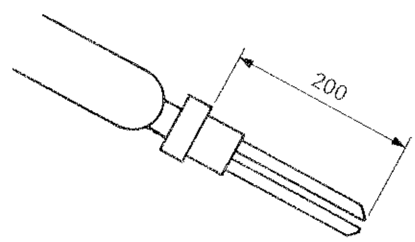
This setting is made to effect as accurate axis motions as possible and is not necessary if the robot is only moved through a series of taught points. If the commands in the cartesian coordinate system, such as pallet commands, are to be used, however, this setting must be made prior to teaching. Once this setting is made and is stored in EPROM, the setting procedure is no longer necessary. (For more details, see CHAPTER 4 CARTESIAN COORDINATE SYSTEM REFERENCE POSITION SETTING, APPENDIX.)

5.2 Setting the Tool Length

This setting defines the position of the hand end to move the robot axes as accurately as possible and is not required when the robot is only moved through a series of taught points. This procedure is not necessary either when using the robot with the optional motor-operated hand because the position of the hand end represents a point about 107mm away from the hand installation surface and defaults to the standard value (107mm).
However, a new end position must be defined if the robot is operated by a command in the cartesian coordinate system (e.g. pallet command) using any other hand than the above or using the motor-operated hand with an additional tool.
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 command "TL" through the personal computer. After setting the tool length, teach the robot through the desired positions. (See Command "TL," DESCRIPTION OF THE COMMANDS.)

Example: Tool length 200mm

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



NOTE

- When positions are taught with a tool length other than the standard (107mm) set, be sure to define the same tool length at the beginning of the program in which those taught positions are used. If this setting is wrong, the robot hand may travel too far to cause a hit. Note that the tool length defaults to 107mm.

  Example: (Teaching) Position 1 is set with a 200-mm tool length.
  (Programming) 1 TL 200 2 NT 3 MO 1
  (Direct execution) TL 200 ↓ MO 1 ↓

5.3 Defining, Verifying, Changing, and Deleting the Positions

Here, let us actually operate the teaching box to define, verify, change, and delete positions.
Now, turn ON the teaching box ON/OFF switch.

(1) Defining positions
Let us define three different positions as follows.

1) Press the appropriate jog key or keys to move the end of the arm to an appropriate position.

2) Suppose this is position "10." Now, hit the following keys successively:

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

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

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

(2) Verifying positions
Let us verify if positions have been correctly defined.

1) To verify position 10, hit the following keys successively:

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

If the position has been correctly defined, the end of the arm moves to the point mentioned above.

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

(3) Changing positions
Let us change, or redefine, previously defined positions.

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

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

This clears the old position data and redefines position 10.

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

(4) Deleting positions
You may even wish to delete some positions.

1) To delete position 10, hit the following keys successively:

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

   This clears position 10 making it available for new definition.

2) To verify that position 10 has been properly deleted, hit the following keys successively:

   [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.

6. PROGRAM GENERATION AND EXECUTION

This chapter describes the procedures required to generate a program in the personal computer mode by using the positions previously defined and to execute it.
The description that follows assumes the teaching box ON/OFF switch is in the OFF position.

6.1 Generating and Transferring a Program

We will here generate a model program in the personal computer mode described earlier. Now, why don't we write a simple program using three positions (numbered 10, 11, and 12)? The following lists the program sequence, where the numbers at the beginning represent the Movemaster program line numbers. For details of the commands, see DESCRIPTION OF THE COMMANDS.

Movemaster program

```
10 NT              ; Origin setting (nesting).
12 SP 7            ; Set speed at 7.
14 MO 10, O        ; Move to position 10 with hand opened.
16 MO 11, C        ; Move to position 11 with hand closed.
18 MO 12, C        ; Move to position 12 with hand closed.
20 TI 30           ; Stop for 3 seconds.
22 GT 14           ; Jump 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                        ; End BASIC program.
RUN                            ; Run BASIC program.
OK
```

For the RS232C interface, execute the following BASIC program. For details of the settings to be made, see CHAPTER 2 INTERFACE WITH THE PERSONAL COMPUTER (RS232C), APPENDIX.

```
100 OPEN "COM1 : 9600, E, 7, 2" AS #1 ; For Mitsubishi MULTI16.
110 PRINT #1, "10 NT"
120 PRINT #1, "12 SP 7"
130 PRINT #1, "14 MO 10"
140 PRINT #1, "16 MO 11"
150 PRINT #1, "18 MO 12"
160 PRINT #1, "20 TI 30"
170 PRINT #1, "22 GT 14"
180 END                        ; End BASIC program.
RUN ↲                          ; Run BASIC program.
OK
```

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

6.2 Executing the Program

Let us then execute the generated program using the following procedure.

6.2.1 Step execution

The generated program may be executed, line by line, by operating the teaching box keys for verification. Here is the procedure.

1) Turn ON the teaching box ON/OFF switch.

2) To execute the program starting with line number 10, hit the following keys successively in that order.

   [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, key in:

   [STEP] [ENT]

   This causes the command "SP7" on line number 12 to be executed.

4) Repeat step 3) to verify the program, line by line. Note that no entry of line number is necessary.
   The operating speed for step execution is automatically set to SP4 or less.

6.2.2 Starting the program

In the personal computer mode, the generated program can be initiated by the personal computer. Here is the procedure.

1) Turn OFF the teaching box ON/OFF switch.

2) Execute the command "RN" to directly run the program.

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

While the program is running, the current line number is shown on the teaching box LED.

6.2.3 Stopping/restarting the program

The program currently running can be stopped and restarted by operating certain switches on the front control panel of the drive unit.

Stopping : Press the [STOP] switch, and the robot stops after completing executing the current line number. This means that, if the robot is executing a move command when the switch is pressed, it will complete reaching the destination point. At this time, the teaching box LED shows the line number in which the robot was stopped.

Restarting : Press the [START] switch. This causes the program to restart from the line number following the one stopped.

Turning ON the teaching box ON/OFF switch while the program is being halted enables the operation from the teaching box. Be sure, however, to turn the switch OFF before pressing the [START] switch to restart the program. Note also that no command can be executed from the personal computer even in the personal computer mode while the program is being halted. To execute commands, perform the following resetting operation.

6.2.4 Stopping/resetting the program

The program currently running can be stopped and reset by operating certain switches on the front control 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. The operation is the same as described before.

Resetting : Press the [RESET] switch after the [STOP] switch has been pressed, and the program is reset.

To restart the program following the resetting operation in the personal computer mode, the command "RN" must be used. At this time, pressing 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 IN EPROM (PERSONAL COMPUTER MODE)

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

7.1 Inserting Erased EPROM

Insert a new EPROM or one erased by the EPROM eraser (the recommened EPROM model being Mitsubishi M5L27256K with 250-ns access time, write voltage: 12.5V) into the user socket on 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

We shall now write the contents of the drive unit RAM into the EPROM.

Using the teaching box

1) Turn ON the teaching box ON/OFF witch.
2) Press [WRT] and [ENT], successively in that order.

Using intelligent commands through the personal computer

1) Turn OFF the teaching box ON/OFF switch.
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 up green. The LED goes out in about 100 seconds after the data has been written correctly. Do not remove the EPROM from the socket until the LED goes out. Should the data be written incorrectly (due to an EPROM failure or write error), it causes error mode II. If the error condition occur, reset the condition and try another EPROM.

7.3 Precautions for Storage of EPROM

Be sure to affix an ultraviolet rays shielding seal onto the glass window of the EPROM into which data has been written 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

This chapter describes the procedure to operate the robot in the drive unit mode using the program and position data written in the EPROM.
The procedure given in the following starts with the condition before turning power ON.

8.1 Inserting the EPROM

Insert the EPROM, into which the program and position data have been written, into the user socket on the drive unit. Make sure of the correct installation direction of the EPROM, the indentation being positioned on the left.

8.2 Setting the Side Setting Switches

Before turning power ON, make the following switch settings by referring to 2.1.2 Functions of side setting switches and LEDs.

ST1 ······ Upper position (Drive unit mode)
ST2 ······ Upper position (EPROM data to be transferred to RAM)

8.3 Turning Power ON

Turn ON the power switch on the rear panel of the drive unit. Then, the EPROM data is transferred to the drive unit memory RAM according to the switch setting made in the preceding step.

8.4 Executing the Program

Now, let us execute the program transferred to RAM in the following procedure.

8.4.1 Step execution

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

8.4.2 Starting the program

In the drive unit mode, the transferred program may be initiated by operating a certain switch on the drive unit front control panel. Here is the procedure.

1) Turn OFF the teaching box ON/OFF switch.

2) Press the [START] switch on the front control panel.

The teaching box LED shows the current line number being executed.

8.4.3 Stopping/restarting the program

The program currently running can be stopped and restarted by operating certain switches on the front control panel of the drive unit. The procedure is the same as that for the personal computer mode. (See 6.2.3 Stopping/restarting the program.)

8.4.4 Stopping/resetting the program

The program currently running can be stopped and reset by operating certain switches on the front control 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. The operation is the same as described before.

Resetting : Press the [RESET] switch after the [STOP] switch has been pressed, and the program is reset.

Pressing the [START] switch following resetting causes the program to be executed starting with the first line number. Note that the general-purpose I/O outputs are not reset.

9. OPERATION USING THE EXTERNAL SIGNALS

In the drive unit mode, the program is run by operating the front control switches of the drive unit as we have seen. This chapter outlines the procedure to effect these operations using the external signal lines with an I/O card (type A16 or B16) inserted in position. For more details, see CHAPTER 3 INTERFACE WITH EXTERNAL I/O EQUIPMENT, APPENDIX.

9.1 Setting the Switches

Make the following switch settings by referring to 2.1.2 Functions of side setting switches and LEDs.

ST1 ······· Upper position (Drive unit mode)
SW1 ······ Bits 3 and 5 in upper position (Type A16 or B16 I/O card selected; external signals enabled)

When power is turned ON with the above switch settings made, the external signals are enabled for the execution of the program in the drive unit mode. In these settings, the personal computer, and drive unit front control switches are all disabled, except the emergency stop switch.

9.2 Executing the Program

9.2.1 Starting the program

The program is initiated by a START signal input.

9.2.2 Stopping/restarting the program

Stopping : The program is stopped by a STOP signal input.

Restarting : The program is restarted by a START signal input following the STOP signal.

9.2.3 Stopping/resetting the program

Stopping : The program is stopped by a STOP signal input.

Resetting : The program is reset by a RESET signal input following the STOP signal.

10. ERROR CONDITIONS

This chapter describes various error conditions which may occur during the operation of the Movemaster, including the warning indicators, conditions when the errors occur, possible causes, and remedial action procedures.

10.1 Error Mode I

Error mode I is concerned mainly with hardware errors. The cause of the error may be known by the lit LED inside the side door of the drive unit. (See 2.1.2 Functions of side setting switches and LEDs.)

〈Warning indicators〉
The error indicator LED (red) on the drive unit front panel flashes OFF and ON at 0.5-second intervals. The buzzer, if set to ON, also sounds in phase with the ON timing of the LED. IF type A16 or B16 I/O card is being used, an error signal is output from the dedicated signal line of external I/O equipment connector.

〈Condition〉
Current to motors for all axes (including the hand) is immediately cut off (servo OFF) and brakes are applied to J2 and J3 axes. As a result, the robot is brought to an immediate stop.

〈Possible causes〉
(1) LED1 ON ······ Excessive servo system errors. More precisely, causes include excessive load, encoder signal failure, and robot crashing into an obstacle.
(2) LED2 ON ······ Open or disconnected signal cable between the robot and drive unit.
(3) LED3 ON ······ Drive unit emergency stop input. More precisely, when the emergency stop switch on the drive unit front panel is pressed or when there is an input to the external emergency stop input terminal (N/C contact) on the drive unit rear panel.
(4) LED4 ON ······ Teaching box emergency stop input. More specifically, when the emergency stop push-button switch on the teaching box is pressed, when the teaching box connector comes off while the teaching box is in use, or when the teaching box connector is plugged when power is turned ON.
(5) LED5 ON ······ Backup battery failure. This error occurs only when the battery check facility is enabled (bit 2 of SW1 inside the drive unit side door in ON position).

〈Remedial action〉
Turn power OFF, and then eliminate the cause of the error occurred. When restarting the robot, make sure that the robot has been returned to origin with the arm moved to a safe position.

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 lights up steadily. The buzzer, if set to ON, also sounds continuously. If type A16 or B16 I/O card is being used, 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 enters the mode waiting for error reset. If the error occurs while a program is running, the program stops at the line number on which the error occurs, the line number being shown on the teaching box LED.

<Possible causes>

(1) Command transfer error by the personal computer. More specifically, an undefined command, input format error, or transmission error.

(2) Command is being unexecutable. More precisely, an entry of parameter exceeding the specified range or move command to an undefined position.

(3) EPROM write error. More specifically, the EPROM unerased or EPROM failure.

<Remedial action>
Perform either of the following resetting operations.

Using drive unit switch

Press the [RESET] switch.

Using intelligent commands through the personal computer (in personal computer mode)

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

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

Using dedicated signal line of I/O connector

Turn ON the RESET input signal.

The error indicator LED goes out as soon as the above resetting procedure has been completed. The program, if it has been being executed, is reset and it will be started from the first line number if restarted. Note that general-purpose I/O outputs are not reset by this resetting operation.

1. SPECIFICATIONS

2. OPERATION

3. DESCRIPTION OF THE COMMANDS

4. MAINTENANCE AND INSPECTION

5. APPENDICES

CONTENTS (DESCRIPTION OF THE COMMANDS)

2. DESCRIPTION OF EACH COMMAND

The following pages explain each command in the format shown below. Note that those commands marked with ※ can only be executed directly by the 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, 〈 〉 indicating the command parameter and [ ] indicating 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 meanings of each line and/or some footnotes.

2.1 Position/Motion Control Instructions

DP (Decrement Position)

[Function]

Moves the robot to a predefined position with a position number smaller than the current one.

[Input Format]

```
DP
```

[Sample Input]

DP

[Explanation]

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

(2) If there is no predefined position with a position number smaller than the current one, error mode II is caused.

[Sample Program]

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

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.

[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 open/close state of the gripper, do not change before and after the movement. Error mode Ⅱ is caused if the distance of travel exceeds the robot's operational space.

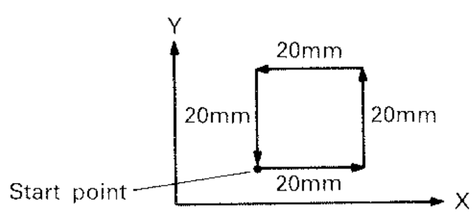
(3) The default distance of travel is 0.

(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 hand is determined by the tool length currently established. (See command TL.)

[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 to reach the start point finally.

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 ≦ 629

[Sample Input]

HE 5

[Explanation]

(1) The coordinates of the current position is calculated based on the currently established tool length (see command TL). In the initial condition, the tool length is 107 mm, a point 107 mm 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) Execution of this command when any of the axes of the robot is positioned near the boundary of its operational space may cause error mode II. In such cases, avoid such attitude of the robot.

(5) Error mode II is caused if this command is executed before the robot returns to the origin.

[Sample Program]

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

HO (Home)

[Function]

Establishes the reference position in the cartesian coordinate system.

[Input Format]

```
HO
```

[Sample Input]

HO

[Explanation]

(1) This command establishes the reference position (in X-, Y-, and Z-axis and pitch/roll angles) in the cartesian coordinate system. All subsequent motions are based on this reference coordinates. (For details, see CHAPTER 4 CARTESIAN COORDINATE SYSTEM REFERENCE POSITION SETTING, APPENDIX.)

(2) This command must be executed to repeat the robotic motion through a series points previously taught after the robot has been disassembled and reassembled for mechanical adjustments.

(3) This command cannot be executed when bit 4 of SW1 located inside the side door of the drive unit is in the lower position (OFF).

[Sample Program]

10 LPRINT "HO"

IP (Increment Position)

[Function]

Moves the robot to a predefined position with a position number greater than the current one.

[Input Format]

```
IP
```

[Sample Input]

IP

[Explanation]

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

(2) If there is no predefined position with a position number greater than the current one, error mode II is caused.

[Sample Program]

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

MA (Move Approach)

[Function]

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

[Input Format]

MA 〈Position number (a)〉 , 〈Position number (b)〉 [, 〈O or C〉]

Where, 1 ≦ Position number (a), (b) ≦ 629

[Sample Input]

MA 2, 3, C

[Explanation]

(1) This command causes the end of the hand to move 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 command SF.)
(Each coordinate of position (b) is temporary 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 is caused before the robot moves.

(4) Error mode II is also caused if positions (a) and (b) have not been defined yet.

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

[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 the two specified position numbers.

[Input Format]

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

Where, 1 ≦ Position number (a), (b) ≦ 629
|Position number (a) − Position number (b)| ≦ 99

[Sample Input]

MC 101, 200

[Explanation]

(1) This command causes the robot to move at a given speed from the current position via position (a) to position (b), 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 as it reaches the end position.

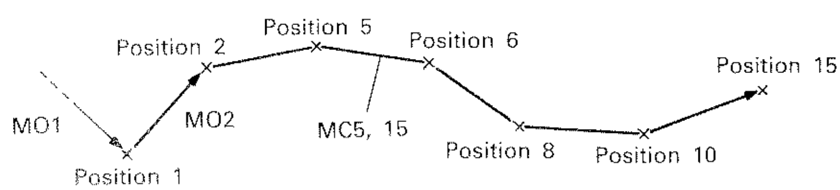
(3) The open/close state of the hand before the robot starts moving remains valid throughout the entire movement and the open/close definitions at intermediate points have no bearing on it.

(4) Since the robot does not 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, the resultant oscillations adversely affect the robot mechanical. (The maximum speed of travel using command MC is equivalent to SP4.)

(5) Error mode II is caused if the specified positions (a) and (b) have not been defined, the path connecting the predefined intermediate points defies physical movement of the robot, or if the difference between the position numbers (a) and (b) exceeds 99.

[Sample Program]

```
10 LPRINT "MO 1"
20 LPRINT "MO 2"
30 LPRINT "SP 3"
40 LPRINT "MC 5, 15"
```



※ In the above example, line numbers 10 and 20 cause the robot to move to positions 1 and 2, respectively, and line number 40 causes the robot to move continuously through the predefined intermediate points 6, 8, and 10 between positions 5 and 15 until reaching position 15.

MJ※ (Move Joint)

[Function]

Turns each joint the specified angle from the current position.

[Input Format]

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

[Sample Input]

MJ 10, 20, −30, 40, −50

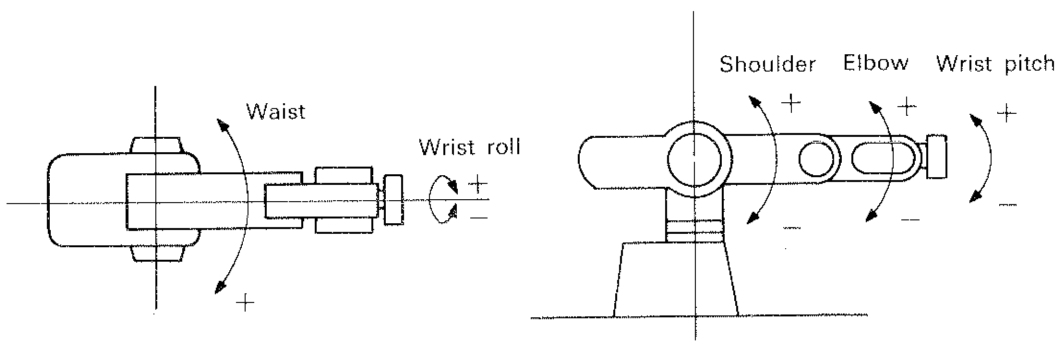
[Explanation]

(1) The least input increment of the turning angle is 0.1°.

(2) The open/close state of the hand does not change before and after the movement. Error move II is caused before the axis motion if the 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 10 causes the waist to sweep 90° in the + direction, line number 20 causes the shoulder to swivel 30° in the − direction, and line number 30 causes the wrist to bend 20° in the + direction.

MO (Move)

[Function]

Moves the end of the hand to a specified position.

[Input Format]

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

Where, 1 ≦ Position number ≦ 629

[Sample Input]

MO 2, C

[Explanation]

(1) This command causes the end of the hand to move 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 command TL.)

(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 is caused if the specified position has not been predefined or the movement exceeds the robot's operational space.

[Sample Program]

```
10 LPRINT "SP 5"            ; Set speed at 5.
20 LPRINT "MO 20, C"        ; Move to position 20 with hand closed.
30 LPRINT "MO 30, O"        ; Move to position 30 with hand open.
```

MP※ (Move Position)

[Function]

Moves the end of the hand to a position whose coordinates (position and angle) are specified.

[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 is caused if the specified coordinates exceed the robot's operational space.

(3) The default coordinate value is 0.

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

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

[Sample Program]

```
10 LPRINT "PD 1, 0, 380, 300, −70, −40"
20 LPRINT "MO 1"
30 LPRINT "MP 0, 380, 280, −70, −40"
```

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

MS (Move Straight)

[Function]

Moves the robot to a specified position number through the specified number of intermediate points on a straight line.

[Input Format]

MS 〈Position number〉 , 〈No. of intermediate points〉 [, 〈O or C〉]

Where, 1 ≦ Position number ≦ 629
1 ≦ No. of intermediate points ≦ 99

[Sample Input]

MS 2, 5, C

[Explanation]

(1) The number of intermediate points between the current position and the specified position number is calculated by equally dividing the distance of travel and position angle (pitch/roll angle) between the two positions (the number of divisions equaling the number of intermediate points plus 1). Acceleration and deceleration during the movement are omitted.

(2) The more the number of specified intermediate points, the smoother the straight line for the movement path, but the more time required for calculation before the robot starts moving (99 intermediate points requires about 2.4 seconds). It is therefore recommended that the number of intermediate points be specified according to the distance of travel and the required path accuracy.

(3) If any of the intermediate points specified exceeds the robot's operational space, the robot stops in mid-motion and error mode II is caused. The hand end velocity during movement changes in accordance with the attitude.

(4) 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 definition of the specified position is executed.

(5) Error mode II is caused if the specified position has not been predefined.

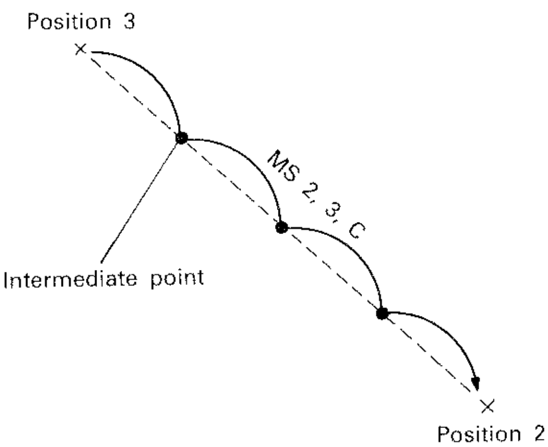
(6) The positions of the intermediate points are calculated based on the tool length currently established. (See command TL.)

(7) Certain positions of the robot may cause oscillations. In such cases, keep the speed low. (The speed should be set to SP5 or lower. 50mm/second max.)

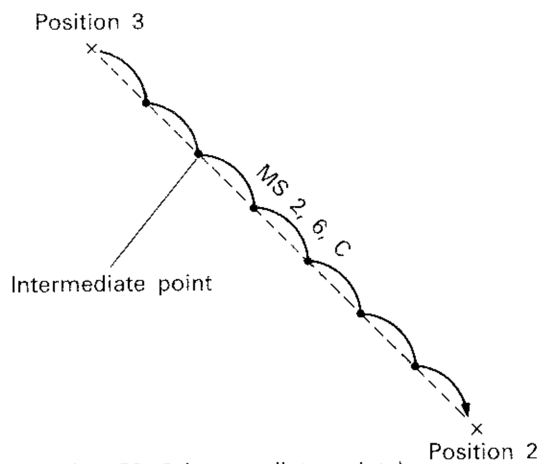
【Sample Program】

```
10 LPRINT "HE 2"
20 LPRINT "MO 3"
30 LPRINT "MS 2, 3, C"
40 LPRINT "MO 3"
50 LPRINT "MS 2, 6, C"
```

※ In the above example, line number 20 causes the robot to move to position 3 by articulated interpolation and line number 30 causes it to move to position 2 through 3 intermediate points with the hand closed. Line number 50 causes the robot to move to position 2 through 6 intermediate points.



(Line number 30, 3 intermediate points)



(Line number 50, 6 intermediate points)

MT (Move Tool)

[Function]

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

[Input Format]

MT 〈Position number〉 , [〈Travel distance〉] [, 〈O or C〉]

Where, 1 ≦ Position number ≦ 629

[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 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 is caused before the end of the hand starts moving when the destination exceeds the robot's operational space. It is also caused 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 command TL.)

[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 is advanced 30mm, from the current position in the tool direction, with the hand closed.

NT (Nest)

[Function]

Returns the robot to mechanical origin.

[Input Format]

```
NT
```

[Sample Input]

NT

[Explanation]

(1) This command causes the robot to return to origin, which must be performed immediately after the power is turned ON. Execution of this command is necessary before any move command can be executed. Origin setting is performed automatically by the limit switches and phase-Z of encoders provided in each axis.

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

(3) Care must be taken to prevent personal injury if the hand holds a workpiece, as the hand opens as soon as the origin setting operation is initiated.

(4) Do not touch the limit switches and robot body until origin setting completes.

[Sample Program]

```
10 LPRINT "NT"              ; Execute origin setting.
20 LPRINT "MO 10"           ; Move to position 10.
```

OG (Origin)

[Function]

Moves the robot to the reference position in the cartesian coordinate system.

[Input Format]

```
OG
```

[Sample Input]

OG

[Explanation]

(1) This command causes the robot to move to the cartesian coordinate system reference position established by the HO command or by keying in [P. S] and [0] on the teaching box. (See CHAPTER 4 CARTESIAN COORDINATE SYSTEM REFERENCE POSITION SETTING, APPENDIX.)

(2) If the reference position is yet to be defined, this command causes the robot to a position as determined by tentative data stored in the system ROM.

(3) Execution of this command prior to origin setting results in error mode II.

[Sample Program]

```
10 LPRINT "NT"        ; Return the robot to origin.
20 LPRINT "OG"        ; Move to reference position in the cartesian
                        coordinate system.
```

PA (Pallet Assign)

[Function]

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

[Input Format]

PA 〈Pallet number〉, 〈No. of column grid points〉, 〈No. of row grid points〉

Where, 1 ≦ Pallet number ≦ 9
1 ≦ No. of column grid points ≦ 255
1 ≦ No. of row grid points ≦ 255

[Sample Input]

PA 3, 20, 30

[Explanation]

(1) This command must be executed before the pallet calculation command (see command PT) 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 command PT.)

[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.

PL (Position Load)

[Function]

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

[Input Format]

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

Where, 1 ≦ Position number (a), (b) ≦ 629

[Sample Input]

PL 5, 7

[Explanation]

(1) After this command has been executed, the coordinates of position (a) becomes equivalent to those of position (b), the old position (a) coordinates being cleared.

(2) After the command has been executed, the position of the hand at position (b) is assigned to that at position (a).

(3) Error mode Ⅱ is caused if position (b) is yet to be defined.

[Sample Program]

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

※ In the above example, after the current coordinates and hand position have been defined as position 2, the position data of position 2 is copied to position 3.

PT (Pallet)

[Function]

Calculates the coordinates of a grid point on a specified pallet number 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) This command calculates the coordinates of a grid point on a specified pallet number and identifies the coordinates as the position number corresponding to the specified pallet number. The pallet definition command (PA) must be executed for the pallet to be used before executing this command. After the PT command has been executed, the position data previously defined for the target position number is cleared.

(2) In order for this 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. 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 command PT allows the coordinates of a grid point to be defined as the position number equivalent to the pallet number.

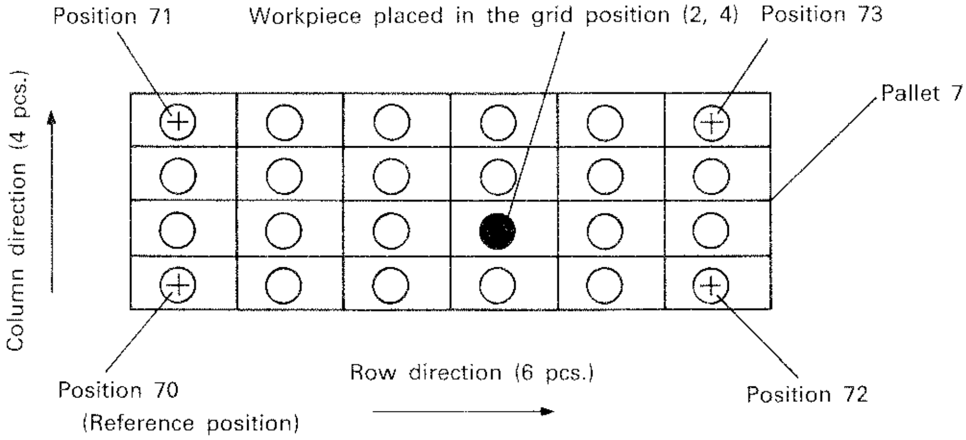
(3) Error mode II is caused if the pallet positions have not been defined and the pallet counters have not been set or have been set to have values exceeding those defined by command PA. An error does not occur, however, even when the coordinates obtained for the grid point exceed the robot's operational 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 command TL. The robot must also be taught through the pallet positions (four corners) using the predefined correct tool length.

[Sample Program]

Suppose you have a pallet on which a total of 24 workpieces are arranged, 4 in the column direction and 6 in the row direction. Now, let us 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"
```

Procedure

(1) We now have pallet 7.
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.

(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)

(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)

(4) Now, 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 a specified position number for those of another specified position number.

[Input Format]

PX ⟨Position number (a)⟩ , ⟨Position number (b)⟩

Where, 1 ≦ Position number (a), (b) ≦ 629

[Sample Input]

PX 2, 3

[Explanation]

(1) After this command has been executed, the coordinates of position (a) are exchanged for the coordinates 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 is caused 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.

SF (Shift)

[Function]

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

[Input Format]

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

Where, 1 ≦ Position number (a), (b) ≦ 629

[Sample Input]

SF 10, 100

[Explanation]

(1) Each coordinate of position (b) is added to the corresponding coordinate of position (a).

(2) Error mode Ⅱ is caused if positions (a) and/or (b) have not been predefined.

(3) This command 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, line number 10 causes the robot to move to that point.

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 ≦ 9

[Sample Input]

SP 7, H

[Explanation]

(1) This command sets the operating velocity and the acceleration/deceleration time upon starting and stopping. The velocity is variable in 10 steps, 9 being the maximum speed and 0 the minimum. Acceleration/deceleration time may be selected from among H or L. The acceleration time is 0.35 seconds for H and 0.5 seconds for L, while deceleration time is 0.4 seconds for H and 0.6 seconds for L. When H is selected, the acceleration and deceleration are constant from SP0 to SP9. When L is selected, the acceleration and deceleration times are constant from SP0 to SP9. (See Appendix 8.1 "Relation between Speed Parameters and Velocities".)

(2) When two or more axes of motion are involved, this command sets the operating velocity of the joint having the greatest number of motor pulses.

(3) When the velocity and acceleration/deceleration time are set, the acceleration and deceleration distance required for movement is predetermined. This means that the set speed may not be reached if the distance of travel is small.

(4) Error mode I may be caused if a high speed and H time are set to effect a backward motion or when the robot's load capacity is large. In such cases, set a low speed and L time.

(5) A speed and acceleration/deceleration time, once set, remain valid until new ones are set. In the initial condition, setting is "SP 4, L". (The last acceleration/deceleration time remains valid when it is omitted.)

(6) If the speed parameter is omitted, it defaults to 0.

[Sample Program]

```
10 LPRINT "SP 3"         ; Set speed at 3.
20 LPRINT "MO 10"        ; Move to position 10.
30 LPRINT "SP 6, L"      ; Set speed at 6 and time L.
40 LPRINT "MO 12"        ; Move to position 12.
50 LPRINT "MO 15"        ; Move to position 15.
```

TI (Timer)

[Function]

Halts the motion for a specified period of time.

[Input Format]

```
TI <Timer counter>
```

Where, 0 ≦ Timer counter ≦ 32767

[Sample Input]

TI 20

[Explanation]

(1) This command causes the robot to halt its motion for the following period of time: specified timer counter value × 0.1 seconds (max. 3,276.7 seconds).

(2) The command can be 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"        ; Move to position 1 (with hand open).
20 LPRINT "TI 5"           ; Set timer at 0.5 sec.
30 LPRINT "GC"             ; Close hand (to grip workpiece).
40 LPRINT "TI 5"           ; Set timer at 0.5 sec.
50 LPRINT "MO 2, C"        ; Move to position 2 (with hand closed).
```

TL (Tool)

[Function]

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

[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) The tool length, once established, remains valid until a new one is defined. When a tool length is changed, the current position is also changed accordingly, which, however, does not involve any robotic motion. (Initial setting of the tool length is 107mm.)

(3) The default value is 0.

(4) Since the point defined by this 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 run, be sure to set the same tool length as that established during teaching 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.

2.2 Program Control
Instructions

CP (Compare Counter)

[Function]

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

[Input Format]

```
CP <Counter number>
```

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

CP 20

[Explanation]

(1) This command must be executed before a conditional jump instruction (see EQ, NE, LG, and SM) is to be executed if the value in the counter whose number is specified is used as the condition for the jump. These conditional jump instructions cause a jump to occur when certain conditions are met involving comparison with the contents of the internal register loaded by the CP command.

(2) Even when the value of the specified counter changes after the execution of the command, the contents of the internal register are not affected. If the conditional jump is specified under the condition specifying comparison with the counter value, therefore, this command must be executed after the counter value has changed.

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

(4) The contents of the counter can be changed or read by means of the counter-related instructions. (See commands SC, IC, DC, and CR.)

[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.

DA (Disable Act)

[Function]

Disables the interrupt by a signal through the specified bit of the external input terminal.

[Input Format]

```
DA <Bit number>
```

Where, $0 \leqq$ Bit number $\leqq 7$ (15)
Figure in ( ) is for type A16 or B16 I/O card.

[Sample Input]

DA 3

[Explanation]

(1) This command clears the interrupt enabled state of the bit defined by the interrupt enable instruction (see command EA).

(2) Once this command has been executed, no interrupt occurs even when a signal is input to the specified bit while the robot is in motion. Note, however, that the execution of this command does not affect the interrupt enabled condition of other bits.

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

[Sample Program]

See command EA.

DC (Decrement Counter)

[Function]

Subtracts 1 from the value in a specified counter.

[Input Format]

DC <Counter number>

Where, 1 $\leqq$ Counter number $\leqq$ 99

[Sample Input]

DC 35

[Explanation]

(1) Error mode II is caused if the counter value becomes less than −32767.

(2) The command can be 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 counter-related instructions. (See commands SC, IC, CP, and CR.)

[Sample Program]

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

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

DL※ (Delete Line)

[Function]

Deletes the contents of a specified line number or numbers.

[Input Format]

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

Where, 1 ≦ Line number (a), (b) ≦ 2048

[Sample Input]

DL 200, 300

[Explanation]

(1) This command deletes all contents from line number (a) up to and including line number (b).

(2) If line number (b) is omitted, the contents of line number (a) only are deleted.

[Sample Program]

```
10 LPRINT "100 MO 10"
20 LPRINT "110 MO 12"
30 LPRINT "120 MO 15"
40 LPRINT "130 MO 17"
50 LPRINT "140 MO 20"
60 LPRINT "DL 130"              ; Delete line number 130.
```

EA (Enable Act)

[Function]

Enables the interrupt by a signal through the specified bit of the external input terminal, and specifies line number to which the program jumps when interrupt occurs.

[Input Format]

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

Where, $0 \leqq$ Bit number $\leqq 7$ (15)
$1 \leqq$ Line number $\leqq 2048$
Figure in ( ) is for type A16 or B16 I/O card.

[Sample Input]

EA +7, 1024

[Explanation]

(1) This command causes an interrupt by an external input signal to be serviced while the robot is in motion. When the specified signal is input while the robot is in motion after this command has been executed, the robot is brought to an immediate stop and the program jumps to the specified line number. Avoid using the command while the robot is moving at high speed, as mechanical failure could result. (Set speed below SP5)

(2) The program jumps when the specified input bit is ON if the parameter + has been specified, and when the input bit is OFF if the parameter − has been specified.

(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 takes precedence.

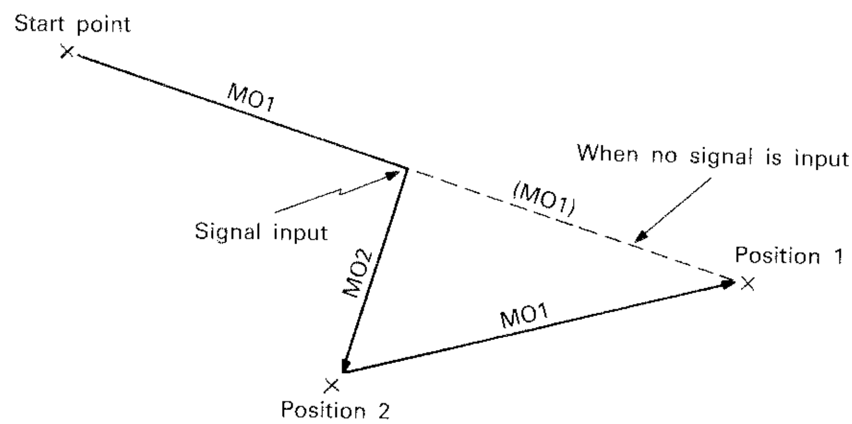
(4) Once this command is executed, the interrupt enabled condition is retained until the interrupt disable instruction (DA), program end instruction (ED), or reset instruction (RS) is executed.

(5) This command is not honored during jogging by means of the teaching box and while the robot is in motion as a result of 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 brought to a halt 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) This command marks the end of a program.

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

[Sample Program]

```
10 LPRINT "100 SP 3"        ; Set speed at 3.
20 LPRINT "110 MO 3"        ; Move to position 3.
30 LPRINT "120 MO 5"        ; Move to position 5.
40 LPRINT "130 ED"          ; End the program.
```

EQ (If Equal)

[Function]

Causes a jump to occur if the contents of the internal register equal a specified value when compared.

[Input Format]

EQ 〈Compared value〉 , 〈Jumping line number〉

Where, $-32767 \leqq$ Compared value (decimal) $\leqq +32767$
or &8001 $\leqq$ Compared value (hexadecimal) $\leqq$ &7FFF
$1 \leqq$ Jumping line number $\leqq 2048$

[Sample Input]

EQ 128, 1024; or EQ &80, 1024

[Explanation]

(1) This command causes a jump to occur conditionally by means of an external input data or the contents of the internal counter.

(2) If the contents of the internal comparison register 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.

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

(4) The compared value may be defined either in decimal or hexadecimal. When a hexadecimal number is used, be sure to append "&" at the beginning of the number.

[Sample Program]

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

GS (Go Sub)

[Function]

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

[Input Format]

GS <Line number>

Where, $1 \leqq$ Line number $\leqq 2048$

[Sample Input]

GS 1024

[Explanation]

(1) This command permits the instruction sequence to jump to a specified line number. The command RT is used to return subsequently to the main program after the subroutine has been completed.

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

(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"
      (
200 LPRINT "90 ED"          ; End the program.
210 LPRINT "100 MO 11"      ; Move to position 11.  )
220 LPRINT "110 MO 12"      ; Move to position 12.  } Subroutine
230 LPRINT "120 MO 13"      ; Move to position 13.  }
240 LPRINT "130 RT"         ; End the subroutine.   )
```

GT (Go To)

[Function]

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

[Input Format]

GT 〈Line number〉

Where, 1 ≦ Line number ≦ 2048

[Sample Input]

GT 1024

[Explanation]

(1) This command causes the program sequence to jump to a specified line number.

(2) If the specified line number is not available, the first line number following the specified one is executed.

[Sample Program]

```
10 LPRINT "20 MO 1"          ; Move to position 1.
20 LPRINT "30 GT 100"        ; Jump to line number 100 unconditionally,

200 LPRINT "100 MO 12"       ; Move to position 12.
210 LPRINT "110 MO 15"       ; Move to position 15.
```

NE (If Not Equal)

[Function]

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

[Input Format]

NE 〈Compared value〉 , 〈Jumping line number〉

Where, −32767 ≦ Compared value (decimal) ≦ +32767
or &8001 ≦ Compared value (hexadecimal) ≦ &7FFF
1 ≦ Jumping line number ≦ 2048

[Sample Input]

NE 128, 1024; or NE &80, 1024

[Explanation]

(1) This command causes a jump to occur conditionally by means of an external input data or the contents of the internal counter.

(2) If the contents of the internal comparison register do 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. sequence.

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

(4) The compared value may be defined either in decimal or hexadecimal. When a hexadecimal number is used, be sure to append "&" at the beginning of the number.

[Sample Program]

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

NW※ (New)

[Function]

Deletes all program and position data.

[Input Format]

```
NW
```

[Sample Input]

NW

[Explanation]

(1) This command deletes all programs and position data stored in the drive unit RAM.

(2) The command does not, however, delete the reference position data in the cartesian coordinate system.

[Sample Program]

```
10 LPRINT "NW"                ; Delete all programs and position data.
```

NX (Next)

[Function]

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

[Input Format]

```
NX
```

[Sample Input]

NX

[Explanation]

(1) This command, used in combination with the RC command, specifies the range of a loop in a program executed by the RC command.

(2) Error mode II is caused if there is no mating RC command specified.

[Sample Program]

See command RC.

RC (Repeat Cycle)

[Function]

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

[Input Format]

RC 〈No. of repeated cycles〉

Where, 1 ≦ No. of repeated cycles ≦ 32767

[Sample Input]

RC 32

[Explanation]

(1) This command,used in combination with the NX command, causes a loop specified by the NX command to be executed a specified number of times and causes the line number following NX to be subsequently executed.

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

[Sample Program]

```
10 LPRINT "20 MO 1"    ; Move to position 1.
20 LPRINT "30 RC 3"    ; Repeat loop delimited by
                         NX three times.
30 LPRINT "40 MO 2"    ; Move to position 2.  ┐
40 LPRINT "50 MO 3"    ; Move to position 3.  ├  Loop
50 LPRINT "60 MO 4"    ; Move to position 4.  ┘
60 LPRINT "70 NX"      ; Delimit the loop.
70 LPRINT "80 MO 5"    ; Move to position 5.
```

RN※ (Run)

[Function]

Executes a specified part of instructions in a program.

[Input Format]

```
RN [ 〈Starting line number〉 ] [, 〈Ending line number〉 ]
```

Where, 1 ≦ Starting/ending line number ≦ 2048

[Sample Input]

RN 20, 300

[Explanation]

(1) This command causes the program to run starting with a specified starting line number and ending with the line number one ahead a specified ending line number.

(2) If the program is to continue, restart with the ending line number.

(3) If the teaching box is connected, the line number being executed is shown on its LED display. If an ending line number is specified, the program stops with that line number shown on the LED.

(4) If the starting line number is omitted, the program starts with the first line number.

[Sample Program]

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

RT (Return)

[Function]

Completes a subroutine and returns to the main program.

[Input Format]

```
RT
```

[Sample Input]

RT

[Explanation]

(1) This command completes the subroutine called by the GS command and returns to the main program.

(2) Error mode II is caused if the mating GS command is not specified.

[Sample Program]

See command GS.

SC (Set Counter)

[Function]

Loads a specified value in a specified counter.

[Input Format]

```
SC  <Counter number> , [ <Value> ]
```

Where, $1 \leqq$ Counter number $\leqq 99$
$-32767 \leqq$ Value (decimal) $\leqq +32767$
or &8001 $\leqq$ Value (hexadecimal) $\leqq$ &7FFF

[Sample Input]

SC 15, 123

[Explanation]

(1) All counters are initially set to 0.

(2) The command can be used to count the number of workpieces and job sequences and to set the number of grid points in the pallet.

(3) The default value is 0.

(4) The contents of the counter can be changed, compared, or read by the counter-related instructions. (See commands IC, DC, CP, and CR.)

(5) The value loaded in the counter is not changed by the command RS, NW, or ED.

[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.

SM (If Smaller)

[Function]

Causes a jump to occur if the contents of the internal register are smaller than a specified value when compared.

[Input Format]

```
SM <Compared value> , <Jumping line number>
```

Where, $-32767 \leqq$ Compared value (decimal) $\leqq +32767$
or &8001 $\leqq$ Compared value (hexadecimal) $\leqq$ &7FFF
1 $\leqq$ Jumping line number $\leqq$ 2048

[Sample Input]

SM 128, 1024; or SM &80, 1024

[Explanation]

(1) This command causes a jump to occur conditionally by means of an external input data or the contents of the internal register.

(2) If the contents of the internal comparison register 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.

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

(4) The compared value may be defined either in decimal or hexadecimal. When a hexadecimal number is used, be sure to append "&" at the beginning of the number.

[Sample Program]

```
10 LPRINT "100 ID"               ; Fetch data from external input port.
20 LPRINT "110 SM 100, 130"      ; Jump to line number 130 if the input data is
                                   smaller than 100.
30 LPRINT "120 ED"               ; End the program if above condition is not met.
40 LPRINT "130 MO 7"             ; Move to position 7.
```

2.3 Hand Control Instructions

GC (Grip Close)

[Function]

Closes the grip of the hand.

[Input Format]

```
GC
```

[Sample Input]

GC

[Explanation]

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

(2) Pneumatically-operated hand:
The command 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 command TI before and after this command. The execution time of this command is determined by the parameter "starting gripping force retention time" for the GP command. (See command GP.)

[Sample Program]

```
10 LPRINT "100 MO 10, O"     ; Move to position 10 (with hand open).
20 LPRINT "110 TI 5"         ; Set 0.5-sec. timer.
30 LPRINT "120 GC"           ; Close hand (to hold workpiece).
40 LPRINT "130 TI 5"         ; Set 0.5-sec. timer.
50 LPRINT "140 MO 15, C"     ; Move 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 command PD).

[Input Format]

GF 〈Switch〉

Where, switch: 0 or 1

[Sample Input]

GF 1

[Explanation]

(1) This command defines the open or close state of the hand grip used in conjunction with the command PD which defines the coordinates of a specified position.

(2) The grip state is open when the switch is set to 0 and closed when the switch is 1. The switch is set to 0 (grip opened) when power is turned ON.

(3) The switch setting, once made, remains valid until it is newly set.

[Sample Program]

See command PD.

GO (Grip Open)

[Function]

Opens the grip of the hand.

[Input Format]

```
GO
```

[Sample Input]

GO

[Explanation]

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

(2) Pneumatically-operated hand:
The command 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 command TI before and after this command. The execution time of this command is determined by the parameter "starting gripping force retention time" for the GP command. (See command GP.)

[Sample Program]

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

GP (Grip Pressure)

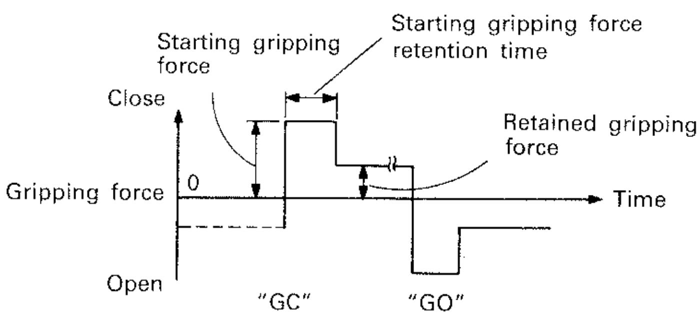
[Function]

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 ≦ 15
0 ≦ Starting gripping force retention time ≦ 99



[Sample Input]

GP 15, 7, 5

[Explanation]

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

(2) The starting and retained gripping forces are 15 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 setting is made.

(3) The initial settings when the power is turned ON are "GP 10, 10, 3."

(4) Parameters, starting and retained gripping forces, are invalid for the pneumatically-operated hand.

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

[Sample Program]

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

OB (Output Bit)

[Function]

Sets the output state of a specified bit through an external output port.

[Input Format]

```
OB <+ or -> <Bit number>
```

Where, 0 ≦ Bit number ≦ 7 (15)
Figure in ( ) is for type A16 or B16 I/O card.

[Sample Input]

OB +1

[Explanation]

(1) Set + to turn ON the specified bit and − to turn 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 command OB, OD, or OT.

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

[Sample Program]

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

※ In the above example, line number 10 sets all external bits (bits 0 to 7) to ON and line number 20 sets bit 0 to OFF.

OD (Output Direct)

[Function]

Outputs specified data unconditionally through the output port.

[Input Format]

OD <Output data>

Where, 0 (−32767) ≦ Output data (decimal) ≦ 255 (+32767)
&0 (&8001) ≦ Output data (hexadecimal) ≦ &FF (&7FFF)
Figure in ( ) is for type A16 or B16 I/O card.

[Sample Input]

OD 7

[Explanation]

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

(2) The output data is defined either in decimal or hexadecimal. For data defined in hexadecimal, be sure to append "&" at the beginning.

(3) For detailed description of connections, see CHAPTER 3 INTERFACE WITH EXTERNAL I/O EQUIPMENT, APPENDIX.

[Sample Program]

```
10 LPRINT "OD &FF"            ; Set all external output port bits (8 bits) to ON.
```

OT (Output)

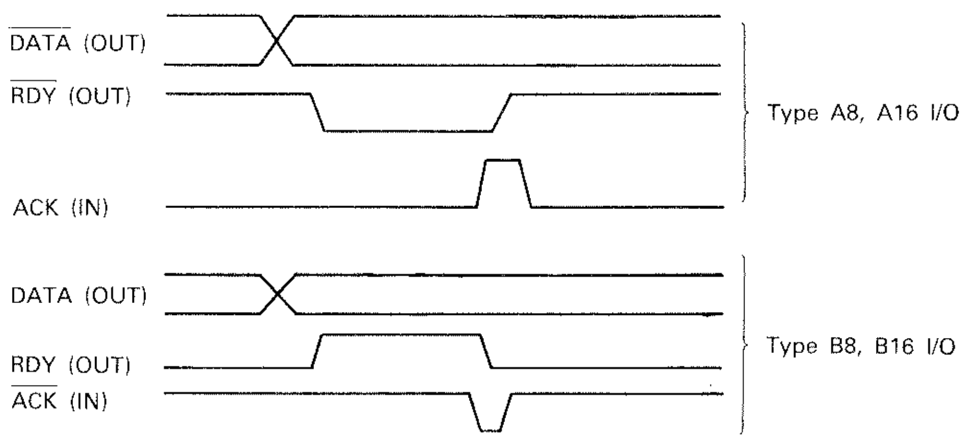
[Function]

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

[Input Format]

```
OT <Output data>
```

Where, 0 (−32767) ≦ Output data (decimal) ≦ 255 (+32767)
&0 (&8001) ≦ Output data (hexadecimal) ≦ &FF (&7FFF)
Figure in ( ) is for type A16 or B16 I/O card.



[Sample Input]

OT 7

[Explanation]

(1) This command causes a signal (parallel data) to be output synchronously through the output port to external equipment such as a programmable controller. At this time, the control signal lines ($\overline{\text{RDY}}$ and ACK signals or RDY and $\overline{\text{ACK}}$ signals) must be previously connected to the external equipment. The data output to external equipment is retained.

(2) The output data is defined either in decimal or hexadecimal. For data defined in hexadecimal, be sure to append "&" at the beginning.

(3) For detailed description of connections, see CHAPTER 3 INTERFACE WITH EXTERNAL I/O EQUIPMENT, APPENDIX.

[Sample Program]

```
10 LPRINT "OT &FF"          ; Set all external output port bits (8 bits) to ON.
```

TB (Test Bit)

[Function]

Causes a jump (or no jump) to occur by means of the contents of a specified bit in the internal register.

[Input Format]

```
TB  <+ or −>  <Bit number> ,  <Jumping line number>
```

Where, 0 ≦ Bit number ≦ 7 (15)
1 ≦ Jumping line number ≦ 2048
Figure in ( ) is for type A16 or B16 I/O card.

[Sample Input]

TB +7, 1024

[Explanation]

(1) This command causes a jump to occur conditionally by means of an external input data or the contents of the internal counter.

(2) The program jumps to the specified line number if the specified bit in the internal comparison register is ON when the parameter + has been defined or is OFF when parameter − has been defined. Otherwise (i.e., if 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 an external input data or by executing the compare counter instruction (see CP) for the counter data. It is therefore necessary to execute previously either of these commands so that a conditional jump can occur.

[Sample Program]

```
10 LPRINT "100 ID"            ; Fetch data from external input port.
20 LPRINT "110 TB +1, 130"    ; Jump to line number 130 if the first bit of the
                                input data is ON.
30 LPRINT "120 ED"            ; End the program if above condition is not met.
40 LPRINT "130 MO 7"          ; Move to position 7.
```

2.5 RS232C Read Instructions

CR (Counter Read)

[Function]

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

[Input Format]

```
CR 〈Counter number〉
```

Where, 1 ≦ Counter number ≦ 99

[Sample Input]

CR 75

[Explanation]

(1) This command causes the contents of a specified counter to be output from the RS232C port.

(2) The output format is ASCII code decimal numbers.

(3) The terminator of output data is 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 0 is returned.

[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, in which Mitsubishi MULTI16 is employed, 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 command SC to define the counter number and data and line number 50 transmits the command CR 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 contents of the data on to the display screen.

DR (Data Read)

[Function]

Reads the contents of the internal register (using RS232C).

[Input Format]

```
DR
```

[Sample Input]

DR

[Explanation]

(1) This command causes the contents of the internal register to be output from the RS232C port. If executed following an input instruction, the command allows the data of the external input port to be read from the RS232C port.

(2) The output format is ASCII code hexadecimal numbers appended with "&H" at the beginning.

(3) The terminator of output data is 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 = &HFF
```

※ In the above example, in which Mitsubishi MULTI16 is employed, 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 next uses the LINE INPUT statement to store the received data in A$ and line number 50 outputs the contents of the data to the display screen.

ER※ (Error Read)

[Function]

Reads the status of the error (using RS232C).

[Input Format]

ER

[Sample Input]

ER

[Explanation]

(1) This command causes the status of the error, as it may or may not be occurring in the robot, to be output from the RS232C port.

(2) The corresponding data is output using ASCII code: 0 when there is no error occurring; 1 in error mode Ⅰ; and, 2 in error mode Ⅱ.

(3) The data can be read not only in the normal condition but also in error mode Ⅰ or Ⅱ.

(4) The terminator of output data is 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) This command is effective when used to check for the occurrence of error before transferring a data stream from a personal computer to the robot.

[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 OCCURS ! "
140 END
```

※ In the above example, in which Mitsubishi MULTI16 is employed, line number 10 opens the RS232C communication file and line numbers from 20 and onward transfer commands to the robot. Note that subroutine 100 is called each time a command is transferred, to check for the occurrence of error.

LR※ (Line Read)

[Function]

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

[Input Format]

```
LR 〈Line number〉
```

Where, 1 ≦ Line number ≦ 2048

[Sample Input]

LR 512

[Explanation]

(1) This command causes the contents of the program on a specified line number to be output from the RS232C port.

(2) The data is output using ASCII code.

(3) The terminator of output data is 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 is read, only 0DH is returned.

(5) The parameters entered in hex. are converted into decimal equivalents which are output. (Example: "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, in which Mitsubishi MULTI16 is employed, 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 a character string. 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.

PR (Position Read)

[Function]

Reads the coordinates of a specified position (using RS232C).

[Input Format]

PR <Position number>

Where, 1 ≦ Position number ≦ 629

[Sample Input]

PR 5

[Explanation]

(1) This command causes the coordinates of a specified position to be output from the RS232C port.

(2) The data is output using the ASCII code as shown below. The least output increment is 0.1mm or 0.1°.
Output format: X-axis coordinate, Y-axis coordinate, Z-axis coordinate, pitch angle, roll angle 0D (Hex.)

(3) The delimiter of the data is a comma (,: Hex. 2C) and the terminator is 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 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
```

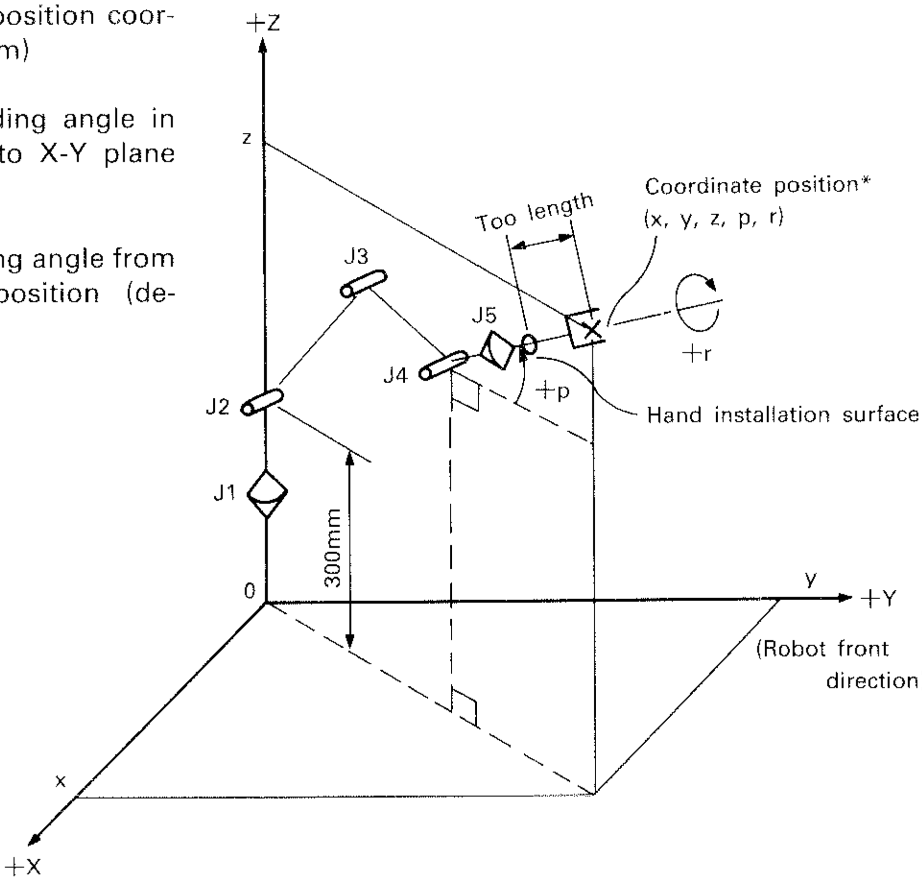
※ In the above example, in which Mitsubishi MULTI16 is employed, line number 10 opens the RS232C communication file and line number 20 enters the position number whose coordinates are to be read. Line number 30 transmits the command PR and the position number. Line number 40 then uses the LINE INPUT # statement to store the received data in A$ and line number 50 outputs the contents of the data to the display screen.

Coordinate System of the Movemaster

*Coordinate position data of the position read using command "PR" or "WH" is defined as shown in the figure on the right.

where

- x, y, z = hand end position coordinates (mm)
- p = hand bending angle in reference to X-Y plane (degree)
- r = hand twisting angle from reference position (degree)



WH (Where)

[Function]

Reads the coordinates of the current position (using RS232C).

[Input Format]

```
WH
```

[Sample Input]

WH

[Explanation]

(1) This command causes the coordinates of the current position of the end of the hand, as determined by the tool length currently established (by command TL), to be output from the RS232C port.

(2) The data is output using the ASCII code as shown below. The least output increment is 0.1mm or 0.1°.
Output format: X-axis coordinate, Y-axis coordinate, Z-axis coordinate, pitch angle, roll angle 0D (Hex.)

(3) The delimiter of the data is a comma (,: Hex. 2C) and the terminator is 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, "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
```

※ In the above example, in which Mitsubishi MULTI16 is employed, line number 10 opens the RS232C communication file and line number 20 transmits the command WH. Line number 30 then uses the LINE INPUT # statement to store the received data in A$ and line number 40 outputs the contents of the data to the display screen.

2.6 Miscellaneous

RS※ (Reset)

[Function]

Resets the program and error condition.

[Input Format]

RS

[Sample Input]

RS

[Explanation]

(1) This command resets the program causing it to return to its beginning.

(2) The command also resets an error condition in error mode II causing the error indicator LED (ERROR) to go off. Note, however, that error mode I cannot be reset by this command. To reset error mode I, turn power OFF.

[Sample Program]

10 LPRINT "RS"

TR※ (Transfer)

[Function]

Transfers the program and position data stored in EPROM to the drive unit RAM.

[Input Format]

```
TR
```

[Sample Input]

TR

[Explanation]

(1) This command causes the contents of the EPROM installed in the user socket inside the drive unit side door to be transferred to the RAM. Once the command has been executed, the old program and position data stored in RAM are all cleared.

(2) The command execution indicator LED (EXECUTE) lights up instantaneously when the data is being transferred.

(3) When bit 4 of SW1 located inside the drive unit side door is in the upper position (ON), the command also causes the cartesian coordinate system reference position data in EPROM to be read into RAM. This transfer of data does not take place when bit 4 is in the lower position (OFF) and the position data in RAM remains valid.

[Sample Program]

```
10 LPRINT "TR"
```

WR※ (Write)

[Function]

Writes the generated program and position data into EPROM.

[Input Format]

WR

[Sample Input]

WR

[Explanation]

(1) This command causes the program and position data generated in the drive unit RAM to be written into the EPROM installed in the user socket inside the drive unit side door. The destination EPROM must have previously been erased before executing the command.

(2) While the data is being written into the EPROM, the command execution indicator LED (EXECUTE) on the drive unit front panel stays lit. It goes out as soon as all data has been written (which takes about 100 seconds).

(3) Error mode II is caused if the EPROM has not been erased or a write error occurs.

[Sample Program]

```
10 LPRINT "WR"
```

' (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) This command allows the programmer to write a comment consisting of up to 7 alphanumeric characters following the ' (apostrophe).

(2) Use this command to write the name and date on the generated program or to mark a subroutine. Comments are helpful in later understanding the program as the command LR 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 '870821"
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 (21 August 1987).

1. SPECIFICATIONS

2. OPERATION

3. DESCRIPTION OF THE COMMANDS

4. MAINTENANCE AND INSPECTION

5. APPENDICES

CONTENTS (MAINTENANCE AND INSPECTION)

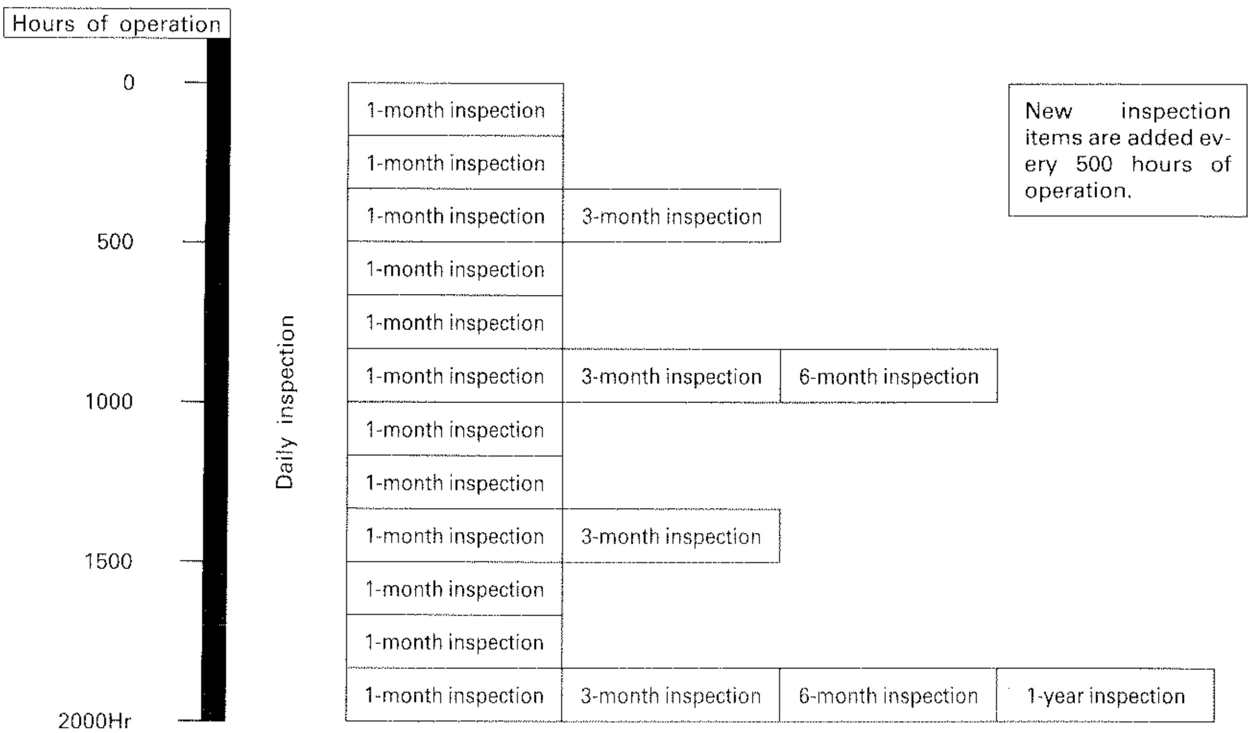
4. MAINTENANCE AND INSPECTION

1. MAINTENANCE AND INSPECTION SCHEDULE

The maintenance and inspection procedures recommended in the following pages are mandatory to ensure best possible performance of the robot for an extended period of time. Be sure to perform the daily as well as periodic inspection procedures. The periodic inspection procedure must be performed independent of the daily inspection procedure.

Inspection Schedule

Different inspection items are covered in different inspection procedures: 1-month, 3-month, 6-month, and 1-year. The corresponding inspection items are added each 500 hours of operation in the recommended inspection schedule.



- Inspection schedule guideline

1-shift:
8 hours/day × 20 days × 3 months = Approx. 500 hours
10 hours/day × 20 days × 3 months = Approx. 600 hours
2-shift:
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 (Procedure)	Remedial Action
(Before power turn-ON)	Verify the following items 1 to 3 before turning power ON.	
1.	Check that the power cable is properly connected. (Visual)	Connect properly.
2.	Check that the cable is connected properly between the robot and drive unit. (Visual)	Connect properly.
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)	Replace or take necessary action.
(Power turn-ON)	Turn power ON being attentive to possible robot motion.	
4.	Ensure the robot does not malfunction or develop unusual noise when power is turned ON. (Visual) NOTE: Note the sound of J2- and J3-axis brakes being released.	Take necessary action according to 2.3 Troubleshooting.
(Operation)	Return the robot to origin and run your program.	
5.	Ensure the point of axis motion is properly aligned. If it is out of alignment, check for followings. 1) Installation bolts left loose (Use a tool to retighten.) 2) Hand mounting bolts left loose (Use a tool to retighten.) 3) Jigs out of alignment (Visual) 4) If the problem persists, use procedures shown in 2.3 Troubleshooting.	Take necessary action according to 2.3 Troubleshooting. Retighten. Retighten. Correct position.
6.	Ensure there is no unusual noise or vibration. (Visual)	Take necessary action according to 2.3 Troubleshooting.

Table 4.2.1 Daily Inspection Items

If no trouble is noted

Verify correct operation of peripherals and start automatic operation.

If trouble is noted

Perform troubleshooting procedure according to 2.3 Troubleshooting.

2.3 Troubleshooting

Table 4.2.3 lists troubleshooting procedures that may be used should the system develop any malfunction upon power-up or during robotic operation.

Symptom	Possible Cause (Checking items)	Remedy
Power does not turn ON.	1) Power cord left unplugged from the outlet 2) Blown fuse 3) Wrong source voltage	• Connect properly. • Replace the blown fuse. • Use correct line voltage.
Robot is not activated.	1) ERROR LED is either ON or flashing. 2) Robot arm in contact with mechanical stoppers or peripherals 3) Brakes not released 4) Motor power cable left unplugged	• Verify correct command. • Connect cables properly. • Verify that the emergency stop switch is not being activated. • Keep the robot clear of obstacles. • If the brakes are left applied, adjust brake gaps. (See 3.6.)
Robot cannot return to origin.	1) Returning to origin not invoked. 2) Returning to origin is not terminated.	• Verify correct command and teaching box key operation. • Verify correct ON/OFF operation of the limit switches. • Encoder Z-phase is possibly out of adjustment; contact our Service Division. (See 3.7.)
Program cannot be written from the personal computer.	1) Teaching box select switch in ON position 2) ERROR LED ON 3) Drive unit mode 4) Reset switch not pressed after stop switch has been pressed during execution of a program. 5) Personal computer connection cable left disconnected.	• Turn OFF switch. • Verify correct input format of the command. (See 10.2 Error Mode II, OPERATION.) • Select personal computer mode. (Place drive unit ST1 in the lower position.) • Press reset switch. • Connect cable properly.
Robot stops during its operation.	1) ERROR LED flashing 2) Line voltage drop 3) Power failure including instantaneous 4) Emergency stop switch pressed inadvertently 5) Motor developing unusual smell 6) Driving parts developing unusual noise or vibration	• Connect connection cables properly, or replace broken cable. • Check for excessive robot load. • Check for possible contact of arm with mechanical stoppers and peripherals. If it checks okay, go to "Poor repeatability." • Use the specified voltage range. • Restart the robot beginning with origin setting. • Turn power OFF, then ON. Now start the robot beginning with origin setting. • Replace burnt motor. • Go to "Unusual noise or vibration."
Poor repeatability	1) Unusual noise developing 2) Robot not secured in position 3) Screws in drive transmitting system left loose 4) Timing belt slack off 5) Unusual return-to-origin operation 6) Repeatability restored by performing origin setting again. 7) Hand attaching screws left loose	• Remove noise source. • Secure robot in position. • Tighten screws securely. • Adjust timing belt tension. (See 3.4.) • Go back to "Robot cannot return to origin." • Tighten screws securely.

Table 4.2.3 Troubleshooting Chart (Continue)

Symptom	Possible Cause (Checking items)	Remedy
Unusual noise or vibration	1) Robot mounting screws left loose 2) Brakes being applied 3) Harmonic drive developing unusual noise	• Tighten screws securely. • If brakes are left applied, adjust brake gaps. (See 3.6.) • Harmonic drive may be damaged. Contact our Service Division.
Hand cannot be opened or closed.	1) Hand select switch in wrong position 2) Wrong parameter defined for GP command 3) Hand reaching its service life	• Select hand select switch to either AC or DC depending on the type of drive of the hand being used. • Correct parameter. • Pack worm gear area with grease. (Consult our Service Division.)
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. (Type: Mitsubishi M54523P)
Robot acted unpredictably.	1) Error lamp flashing. 2) Arm sagged at power-on. 3) Given axis misoperated. 4) Robot cables changed within a year.	• Check all points on the left and contact our Service Division.

Table 4.2.3 Troubleshooting Chart

3. MAINTENANCE AND INSPECTION PROCEDURES

The following parts must be replaced every 2,000 hours of Movemaster EX RV-M1 operation. (It is 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 replacing parts at the customer's site. Read the procedures carefully to make yourselves familialize with them.
The replacement work may be done by our Service Center at customer's expense.
Never attempt to disassemble parts not covered in the following pages.

Parts requiring replacement every 2,000 hours of operation:

1. DC servomotor brush (for waist, shoulder, and elbow)
2. DC servomotor (for wrist pitch and roll)
3. Timing belt (for shoulder, elbow, and wrist pitch)
4. Curled cable (for power and signal)

- The model and quantity of these parts are contained in 4. SERVICE PARTS, MAINTENANCE AND INSPECTION.

WARNING

The replacement procedures when performed 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 construction schematic of the Movemaster EX RV-M1.

(1) Waist (J1 axis) sweep

- The waist (J1 axis) is driven by J1 axis motor ① and harmonic drive reduction gear ② located inside the base.
- J1 axis limit switch ③ is installed on top of the base.

(2) Shoulder (J2 axis) swivel

- The shoulder (J2 axis) is driven by the J2-axis harmonic drive reduction gear ⑥ which is located at the shoulder joint and rotated by the J2-axis timing belt ⑤ from J2-axis motor ④ installed inside the shoulder cover.
- Installed on the input shaft of the harmonic drive reduction gear ⑥ is the electromagnetic brake ⑦ that prevents the shoulder from swiveling down by its own weight when power is turned OFF.
- The J2-axis limit switch ⑧ is installed on the side of the upper arm inside the shoulder cover.

(3) Elbow (J3 axis) extension

- Just as with J2 axis, rotation of the J3-axis motor ⑨ located inside the shoulder cover is transmitted via the J3-axis timing belt ⑩ to the J3-axis harmonic drive reduction gear ㉑.
- Rotation of the J3-axis output shaft of the harmonic drive reduction gear ㉑ is transmitted via the J3-axis drive link to the elbow shaft, which results in the forearm extending.
- As in J2 axis, the electromagnetic brake is installed on the input shaft of the harmonic drive reduction gear ㉑.
- The J3-axis limit switch is installed on the side of the upper arm inside the shoulder cover.

(4) Wrist pitch (J4 axis)

- Rotation of the J4-axis motor located in the forearm is transmitted via the J4-axis timing belt ⑮ to J4-axis harmonic drive reduction gear ⑯; as a result, the wrist housing and end effector are rotated.
- The J4-axis limit switch ⑰ is installed under the forearm.

(5) Wrist roll (J5 axis)

- The hand mounting flange is rotated by the J5-axis motor ⑱ and J5-axis harmonic drive reduction gear ⑲ which are both mounted on a same shaft inside the wrist housing.
- The J5-axis limit switch ⑳ is installed under the forearm.

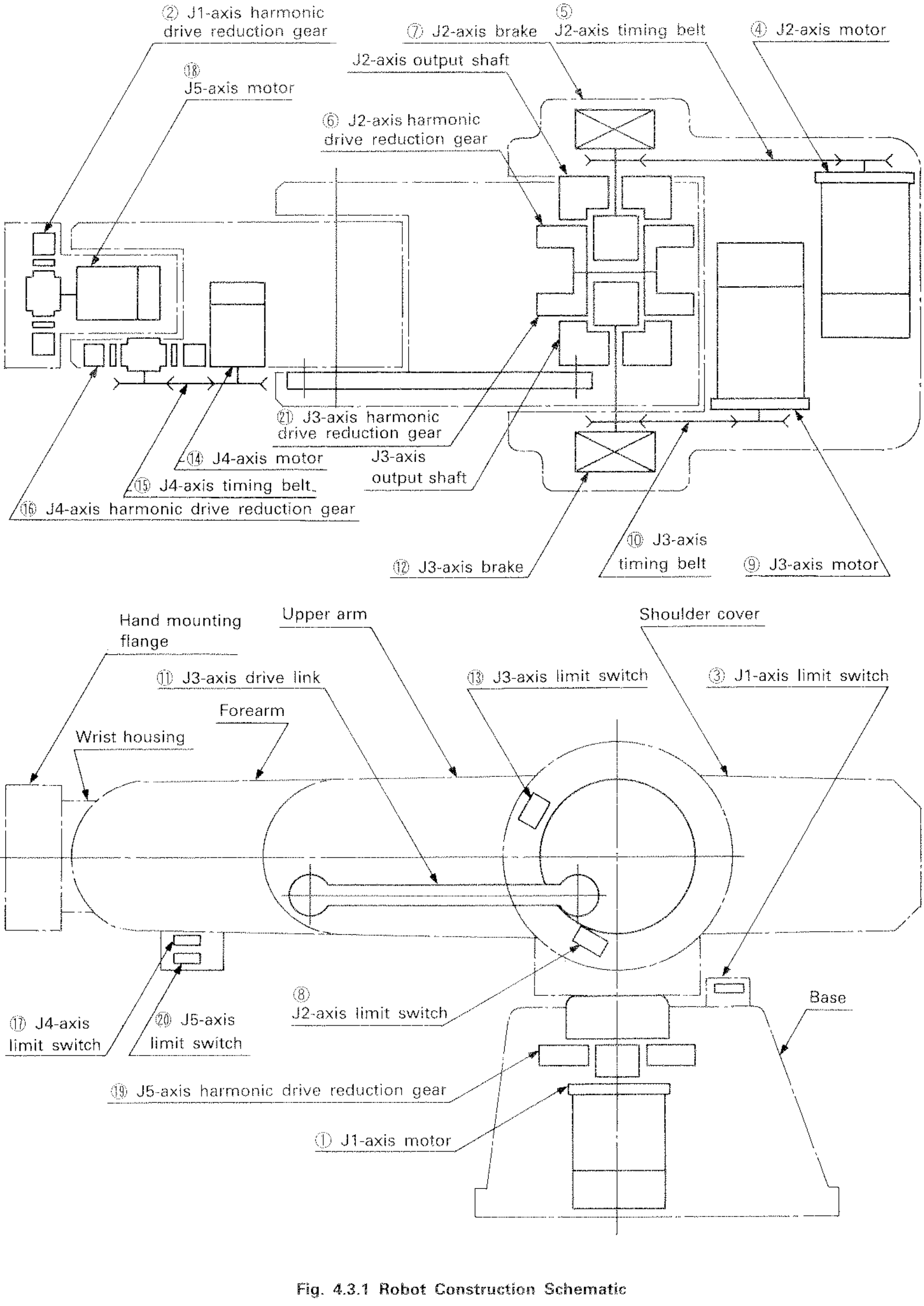


Fig. 4.3.1 Robot Construction Schematic

3.2 Removal of the Covers

(1) Whenever the robot is to be serviced, remove the covers as shown in Fig. 4.3.2.

(2) Table 4.3.1 lists the names of the covers. Table 4.3.2 gives a complete listing of attaching screws.

(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) To replace the covers, reverse the order of removal.

Key	Cover Name	Qty
[1]	Base side cover	2
[2]	Upper shoulder cover	1
[3]	Lower shoulder cover	1
[4]	Upper arm cover	1
[5]	Forearm cover	1
[6]	Pulley cover	1
[7]	Cable cover	1

Table 4.3.1 Cover Names

Key	Screw Name	Qty
(a)	Phillips pan head machine screw M3 × 14	4
(b)	Hexagon socket head bolt M3 × 12	2
(c)	Hexagon socket head bolt M3 × 14	4
(d)	Phillips flat head machine screw M3 × 6	1
(e)	Phillips oval head machine screw M3 × 6	5
(f)	Phillips oval head machine screw M3 × 25	4

Table 4.3.2 Attaching Screws

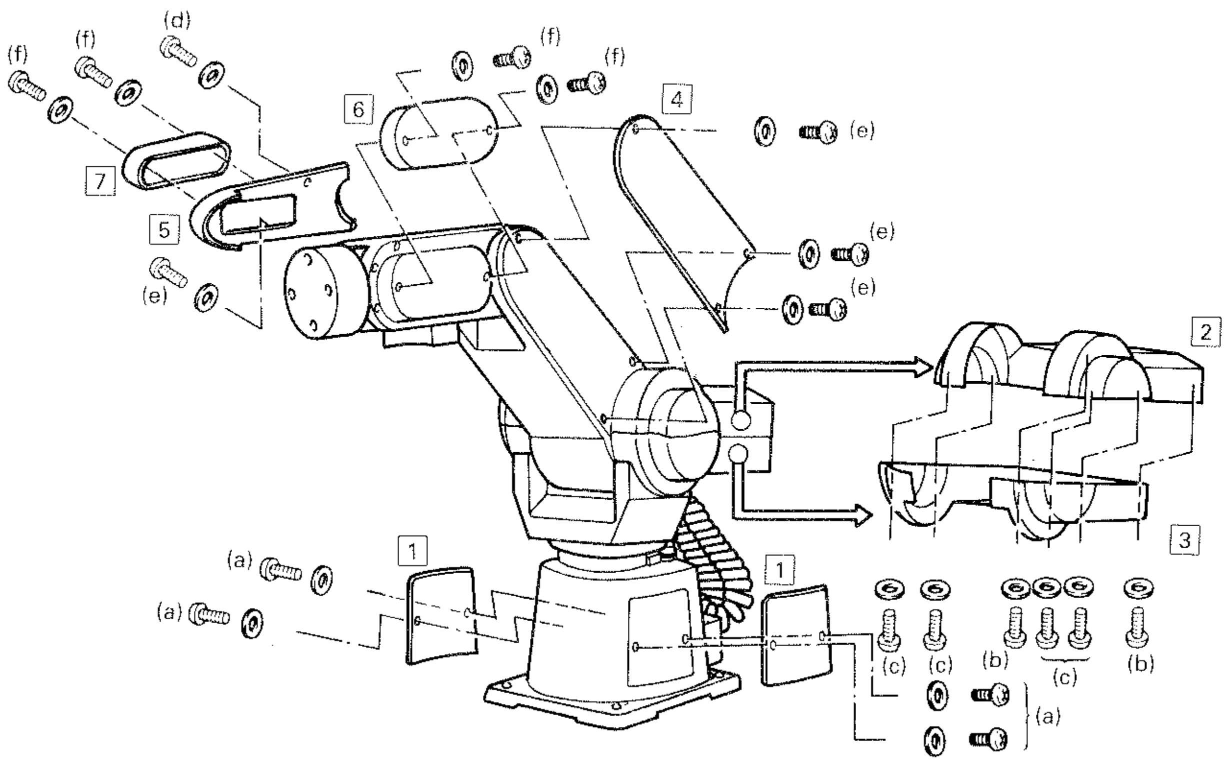


Fig. 4.3.2 Removal of the Covers

3.3 Replacement of the Motor Brush

The RV-M1 robot uses the DC servomotor with brush, which makes the replacement of the brush at given intervals necessary. The DC servomotor with brush is used in all axes; however, for a reason of mechanical construction, replace the whole servomotor assembly for those used as the wrist pitch and roll drives.
As a guideline, replace the motor brush every 2,000 hours of operation.

3.3.1 Checking and replacing 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 the left and right base side covers by referring to 3.2 Removal of the Covers.

(3) Using a flat-blade screwdriver, turn the head of the brush holder cap ① counterclockwise to remove the brush holder cap ①. There are two brush holder caps at right and left.

(4) Fig. 4.3.4 shows the shape of the brush and the brush wear limit length. Check the brush for wear and replace it before the wear limit is reached.

(5) If the brush is damaged, replace it with a new one regardless of the length of the brush.

(6) Check the brush one location at a time. When it is reinstalled, make sure of the correct installed position and direction.

B. Replacement

(1) Following the procedure given above, remove all brushes.

(2) Using a vacuum cleaner, remove all carbon particles from the brush holder pocket.

(3) Carbon particle residual can be a cause of loss of insulation and rectification failure between grounding points. Be sure to remove all particles.

(4) Install the 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 user program.

NOTES:

1) Before installing a new brush, make sure the brush is free from grease, oil, and moisture.
Never handle 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 brush holder cap ① to 4 kgf·cm.

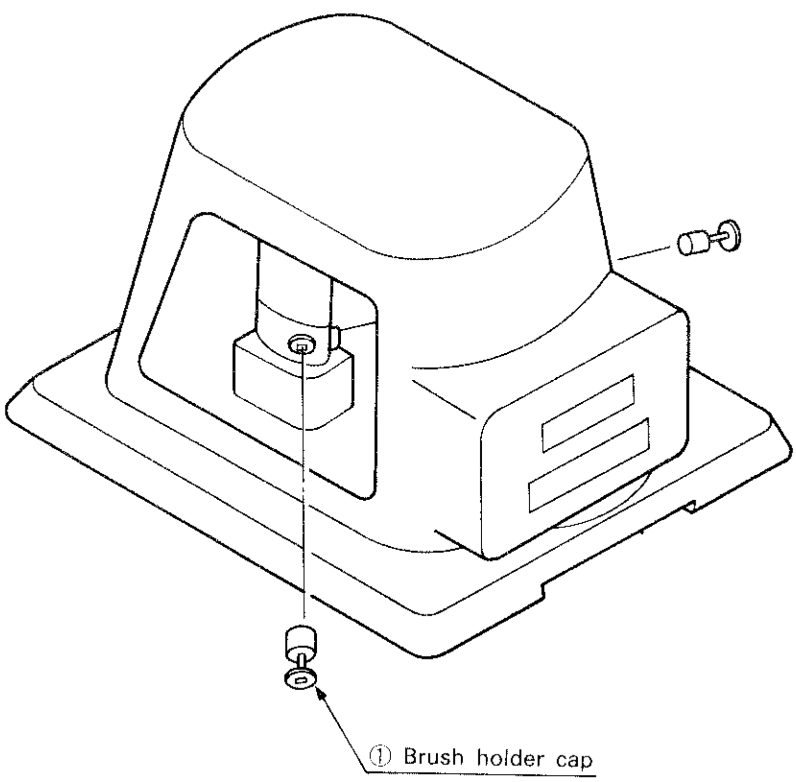


Fig. 4.3.3 Checking and Replacing the Waist Drive Motor Brushes

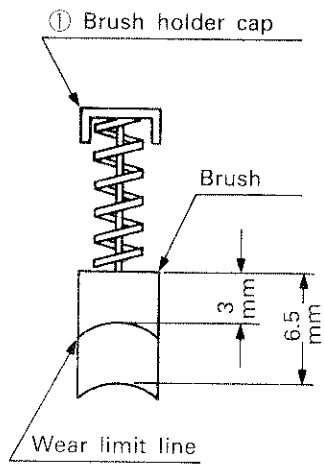


Fig. 4.3.4 Brush Shape

3.3.2 Checking and replacing the shoulder/elbow drive motor brushes

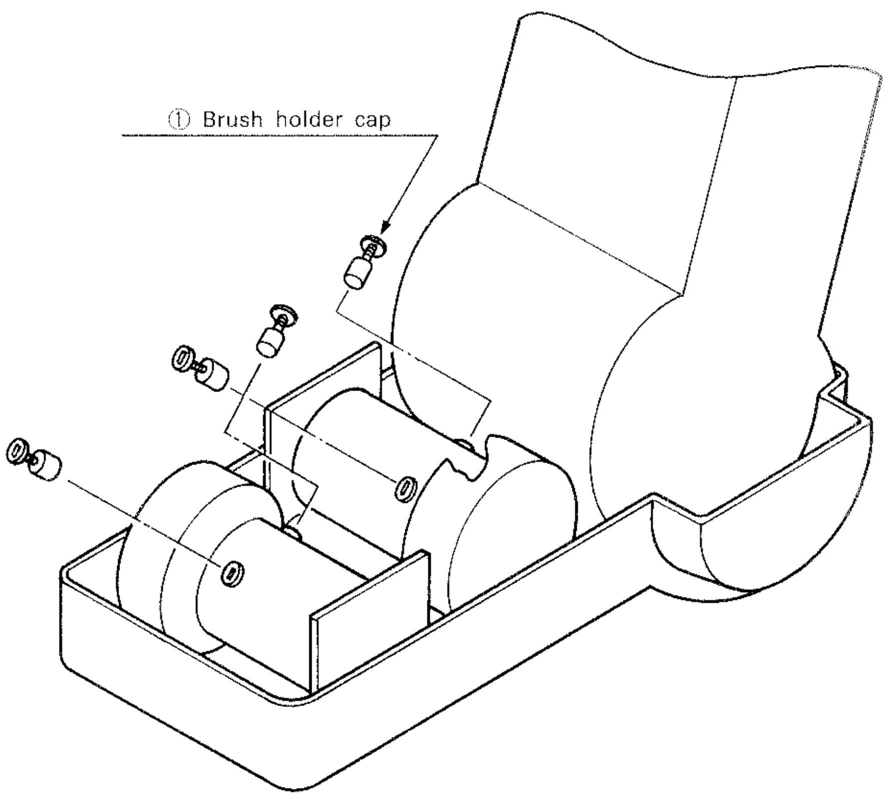


Fig. 4.3.5 Checking and Replacing the Shoulder/Elbow Drive Motor Brushes

A. Inspection

(1) Fig. 4.3.5 outlines the inspection and replacement procedures of the shoulder and elbow drive motor brushes.

(2) Remove the upper shoulder cover by referring to 3.2 Removal of the Covers.

(3) Following steps (3) to (6), A. Inspection, given under 3.3.1 Checking and replacing the waist drive motor brushes, check the brushes.

B. Replacement

(1) Following steps (1) to (5), B. Replacement, given under 3.3.1 Checking and replacing the waist drive motor brushes, replace the brushes.

3.3.3 Replacing the wrist pitch drive motor

A. Replacement

(1) Fig. 4.3.6 outlines the replacement procedure of the wrist pitch drive motor.

(2) Remove the forearm cover and pulley cover by referring to 3.2 Removal of the Covers.

(3) The wrist pitch drive motor ① is installed in the Forearm ②.

(4) The J4-axis encoder connector ③ (connector name: ENC4) and J4-axis motor power connector ④ (connector name: M4) are connected to the motor inside the forearm. Remove these wires from the motor.

(5) Remove the wrist pitch drive timing belt ⑤. (For procedure, see 3.4.4 Checking, adjusting, and replacing the wrist pitch drive timing belt.)

(6) Remove the hexagon socket head setscrew ⑦ that secures the timing pulley ⑥ to the motor shaft and remove the timing pulley ⑥ from the motor shaft.

(7) Remove the two motor mounting screws ⑧ that secure the motor ① to forearm ② and remove the motor ① from the forearm ②.

(8) To install a new motor, reverse the order of removal.

(9) After changing the motor, check the limit pulses in accordance with Section 3.7.

WARNING

The replacement of the motor may result in misaligned mechanical origin. The position data must therefore be reviewed after the motor has been replaced.

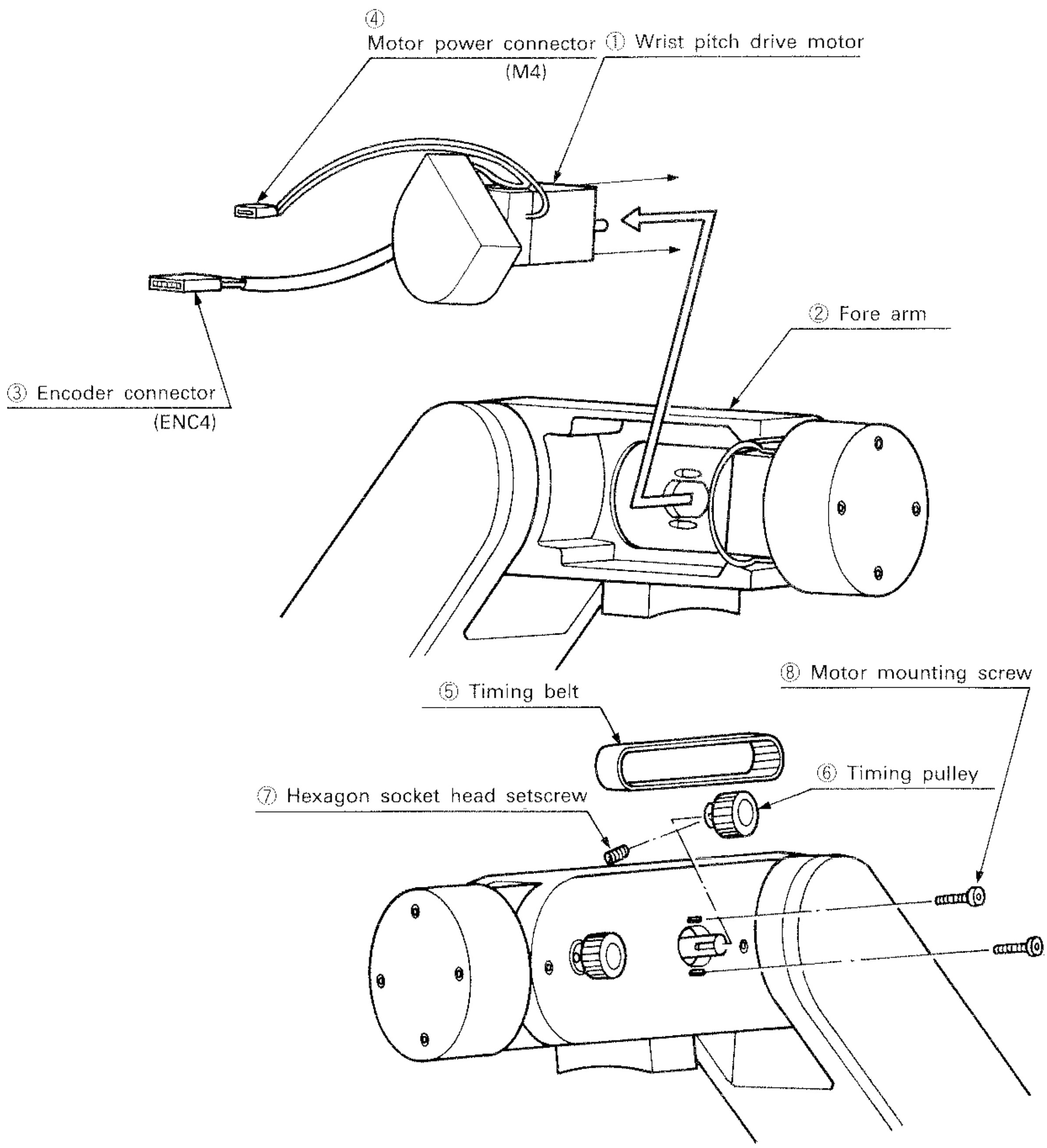


Fig. 4.3.6 Replacing the Wrist Pitch Drive Motor

3.3.4 Replacing the wrist roll drive motor

A. Replacement

(1) Fig. 4.3.7 outlines the replacement procedure of the wrist roll drive motor.

(2) Remove the fore arm cover by referring to 3.2 Removal of the Covers.

(3) The J5-axis encoder connector ① (connector name: ENC5) and J5-axis motor power connector ② (connector name: M5) are connected to the motor inside the fore arm. Remove these wires from the motor.

(4) Remove the motor bracket ④ from the wrist housing ⑤ by removing the four motor bracket mounting screws ③.

(5) Remove the six flange mounting screws ⑥ to remove the flange ⑦ from the motor bracket ④. At this time, the flex-spline shaft ⑧ and wave generator ⑨ of the harmonic drive reduction gear are also removed from the motor bracket. They remain installed on the motor shaft.

(6) Remove the setscrews ⑩ that secure the wave generator ⑨ to the motor shaft and remove the wave generator ⑨ from the motor.

(7) Make sure that the splines in the flex-spline shaft ⑧ and steel balls of the wave generator ⑨ are free from contaminants.

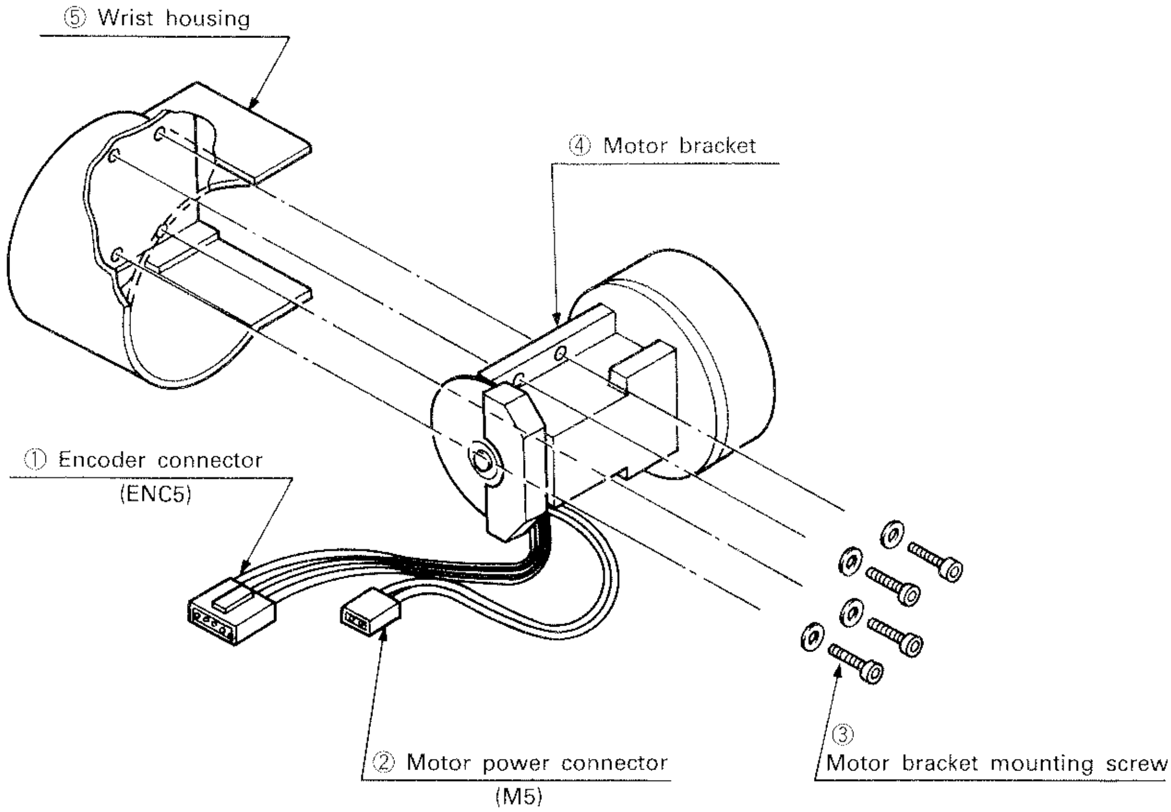
(8) Remove the two motor mounting screws ⑪ from the motor bracket ④.

(9) To install a new motor, reverse the order of removal.

(10) After changing the motor, check the limit pulses in accordance with Section 3.7.

WARNING

The replacement of the motor may result in misaligned mechanical origin. The position data must therefore be reviewed after the motor has been replaced.



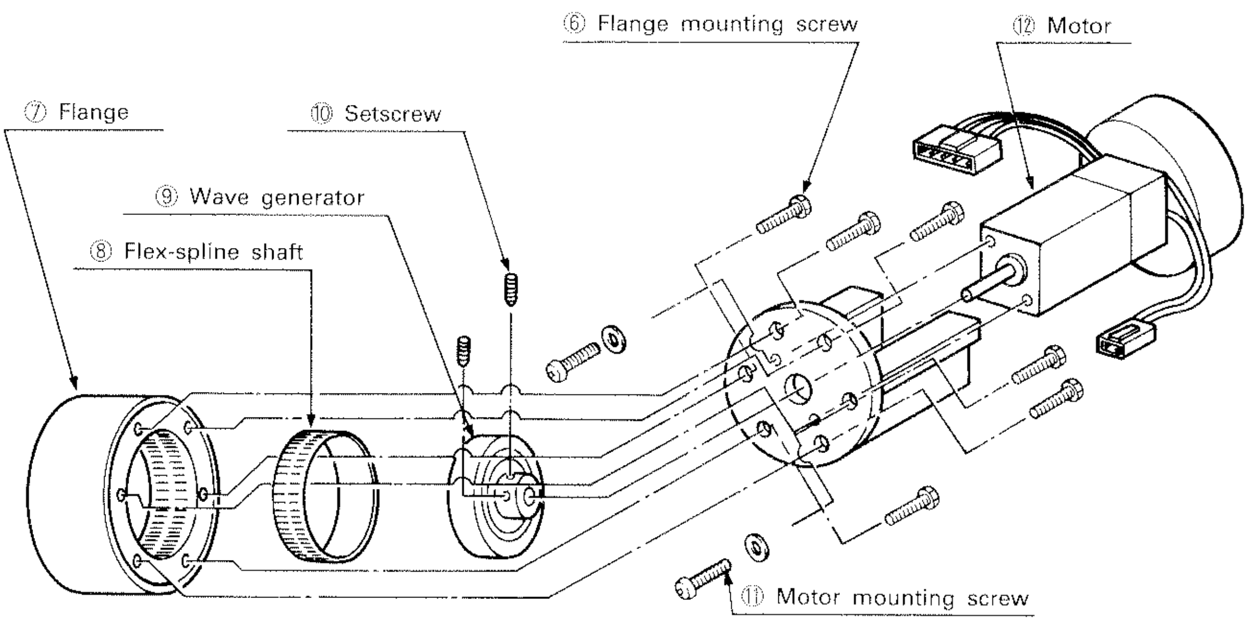


Fig. 4.3.7 Replacing the Wrist Roll Drive Motor

3.4 Adjustment and Replacement of the Timing Belt

Timing belts are used as a drive-transmitting element in the Movemaster EX RV-M1 as shown in Fig. 4.3.1. Unlike chains and gears, the timing belt requires no lubrication and develops low noise. Belt tension, however, is very important in the timing belt. Adjust the belt tension periodically and replace the belt if necessary to ensure long belt life and prevent noise problem.

3.4.1 Replacement frequency

The timing belts used in RV-M1 must be replaced every 2,000 hours of operation under normal condition. More frequent replacement is, however, necessary depending on the operating condition. Use the following guidelines for the replacement of the belt. The timing belt must be replaced with a new one when:

(1) The root or back of a tooth cracks.

(2) The belt swells due to oil and grease on it.

(3) The 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.

NOTE: 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, replace the belt.

3.4.2 Checking, adjusting, and replacing the shoulder drive timing belt

A. Inspection

(1) Fig. 4.3.8 outlines the inspection procedure of the shoulder drive timing belt.

(2) Remove the upper shoulder cover by referring to 3.2 Removal of the Covers.

(3) Visually check the belt to make sure that it does not develop any of the symptoms given in (1) to (5), 3.4.1

(4) Depress the belt at its center with a force of about 20 to 30gf and ensure that it deflects 2mm. See Fig. 4.3.9.

B. Adjustment

(1) Fig. 4.3.8 also outlines the adjustment procedure of the shoulder drive timing belt.

(2) Loosen the four motor mounting screws ①.

(3) While feeling the tension in the timing belt ②, move the motor ③ in the directions shown by arrows "a" and "b". Note that the motor ③ is secured using the four screws in the slots so they can be moved in the slots.

(4) Moving the motor in the direction shown by arrow "a" tightens the belt; moving it in the direction shown by arrow "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 motor mounting screws ① securely. A loose motor mounting screw results in the belt slacking off.

C. Replacement

(1) Fig. 4.3.8 also outlines the replacement procedure of the timing belt.

(2) Remove the brake by referring to 3.6.1 Checking, adjusting, and replacing the shoulder brake.

(3) Remove the three mounting screws ⑥ to remove the brake mounting plate ⑦.

(4) Using a marker, mark alignment pointers on the timing belt ② and timing pulleys ④ and ⑤, while ensuring that the belt teeth are in correct mesh with the pulley teeth. See Fig. 4.3.10.

(5) Loosen the four motor mounting screws ① and remove the timing belt.

(6) Mark the alignment pointers on the new timing belt. During the procedure, keep the belt tightened.

(7) Mount the new belt, getting it around the timing pulleys ④ and ⑤. Make sure that the alignment pointers are lined up.

(8) Adjust the belt tension in accordance with steps (3) to (6) in B and Section 3.4.5.

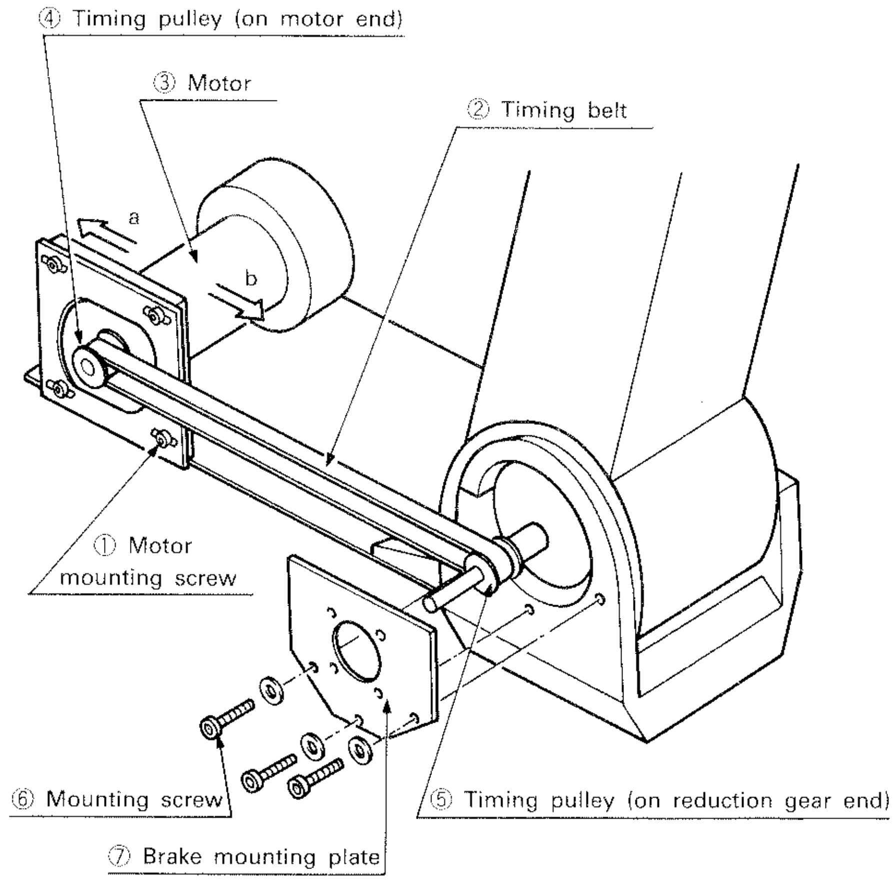


Fig. 4.3.8 Checking, Adjusting, and Replacing the Shoulder Drive Timing Belt

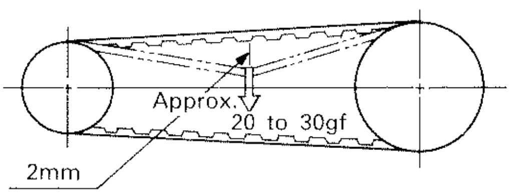


Fig. 4.3.9 Checking the Timing Belt for Proper Tension

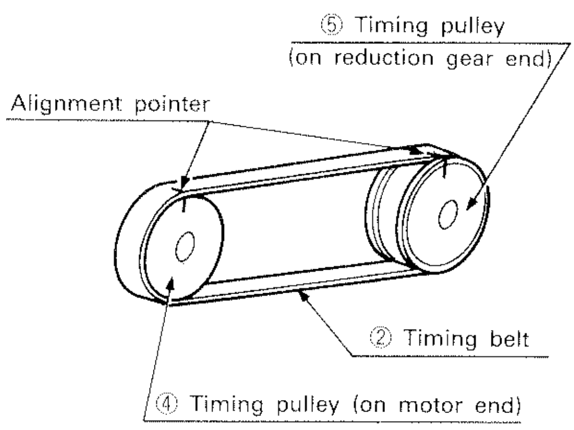


Fig. 4.3.10 Marking the Timing Belt and Pulleys

3.4.3 Checking, adjusting, and replacing the elbow drive timing belt

A. Inspection

(1) Fig. 4.3.11 outlines the inspection procedure of the elbow drive timing belt.

(2) Remove the upper shoulder cover by referring to 3.2 Removal of the Covers.

(3) Visually check the belt to make sure that it does not develop any of the symptoms given in (1) to (5), 3.4.1

(4) Depress the belt at its center with a force of about 20 to 30gf and ensure that it deflects 2mm. See Fig. 4.3.9.

B. Adjustment

(1) Fig. 4.3.11 also outlines the adjustment procedure of the elbow drive timing belt.

(2) Loosen the four motor mounting screws ①.

(3) While feeling the tension in the timing belt ②, move the motor ③ in the directions shown by arrows "a" and "b". Note that the motor ③ is secured using the four screws in the slots so they can be moved in the slots.

(4) Moving the motor in the direction shown by arrow "a" tightens the belt; moving it in the direction shown by arrow "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 motor mounting screws ① securely. A loose motor mounting screw results in the belt slacking off.

C. Replacement

(1) Fig. 4.3.11 also outlines the replacement procedure of the timing belt.

(2) Remove the brake by referring to 3.6.2 Checking, adjusting, and replacing the elbow brake.

(3) Remove the three mounting screws ⑥ to remove the brake mounting plate ⑦.

(4) Be careful, cables are secured to the brake mounting plate ⑦. Do not apply excessive force when removing the mounting plate.

(5) Using a marker, mark alignment pointers on the timing belt ② and timing pulleys ④ and ⑤, while ensuring that the belt teeth are in correct mesh with the pulley teeth. See Fig. 4.3.10.

(6) During adjustment of the belt tension, do not allow the belt to slack off so excessively that the belt comes off the timing pulleys ④ and ⑤.

(7) Mark the alignment pointers on the new timing belt. During the procedure, keep the belt tightened. (See Fig. 4.3.10.)

(8) Mount the new belt, getting it around the timing pulleys ④ and ⑤. Make sure that the alignment pointers are lined up.

(9) Adjust the belt tension in accordance with steps (3) to (6) in B and Section 3.4.5.

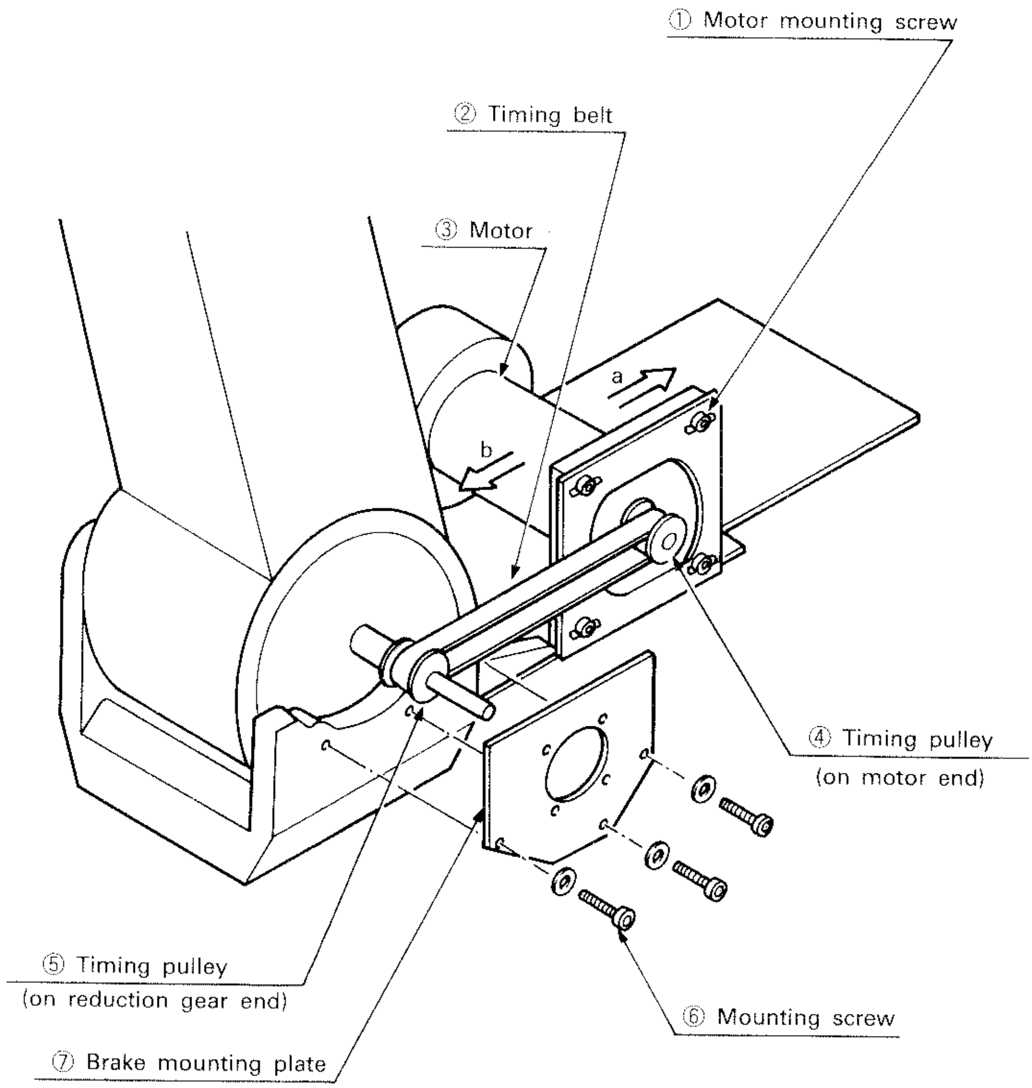


Fig. 4.3.11 Checking, Adjusting, and Replacing the Elbow Drive Timing Belt

3.4.4 Checking, adjusting, and replacing the wrist pitch drive timing belt

A. Inspection

(1) Fig. 4.3.12 outlines the inspection procedure of the wrist pitch drive timing belt.

(2) Remove the upper pulley cover by referring to 3.2 Removal of the Covers.

(3) Visually check the belt to make sure that it does not develop any of the symptoms given in (1) to (5), 3.4.1

(4) Depress the belt at its center with a force of about 20 to 30gf and ensure that it deflects 2mm. See Fig. 4.3.9.

B. Adjustment

(1) Fig. 4.3.12 also outlines the adjustment procedure of the wrist pitch drive timing belt.

(2) Loosen the two motor mounting screws ①.

(3) While feeling the tension in the timing belt ②, move the motor ③ in the directions shown by arrows "a" and "b". Note that the motor ③ is secured using the two screws in the slots so they can be moved in the slots.

(4) Moving the motor in the direction shown by arrow "a" tightens the belt; moving it in the direction shown by arrow "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 motor mounting screws ① securely. A loose motor mounting screw results in the belt slacking off.

C. Replacement

(1) Fig. 4.3.12 also outlines the replacement procedure of the timing belt.

(2) Using a marker, mark alignment pointers on the timing belt ② and timing pulleys ④ and ⑤, while ensuring that the belt teeth are in correct mesh with the pulley teeth. See Fig. 4.3.10.

(3) Loosen the two motor mounting screws ① and remove the timing belt.

(4) Mark the alignment pointers on the new timing belt. During the procedure, keep the belt tightened.

(5) Mount the new belt, getting it around the timing pulleys ④ and ⑤. Make sure that the alignment pointers are lined up.

(6) Adjust the belt tension in accordance with steps (3) to (6) in B and Section 3.4.5.

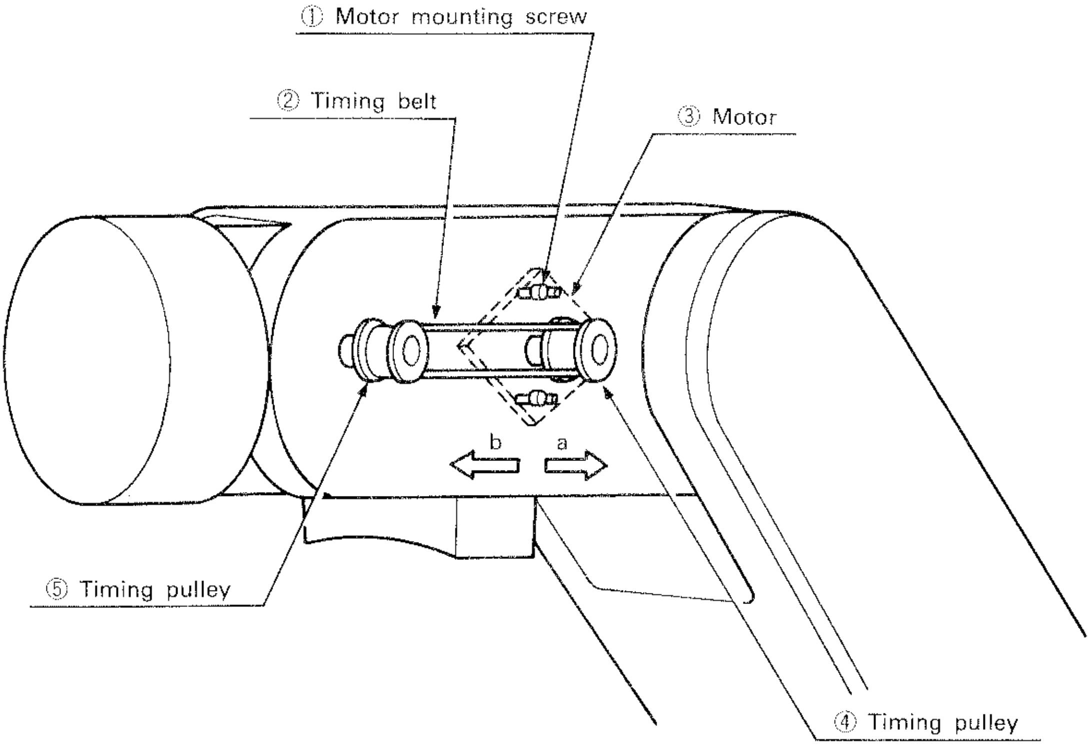


Fig. 4.3.12 Checking, Adjusting, and Replacing the Wrist Pitch Drive Timing Belt

3.4.5 Timing belt tension

Giving an adequate tension to the timing belt is the prerequisite for proper drive transmission and sufficient durability. Do not keep the belt too tight or slack: give it appropriate tension so that you feel 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.

Fig. 4.3.13 and Table 4.3.1 show specifications for belt deflection and load. Ajust the belt tension to obtain the deflection ℓ when load Pk is applied.

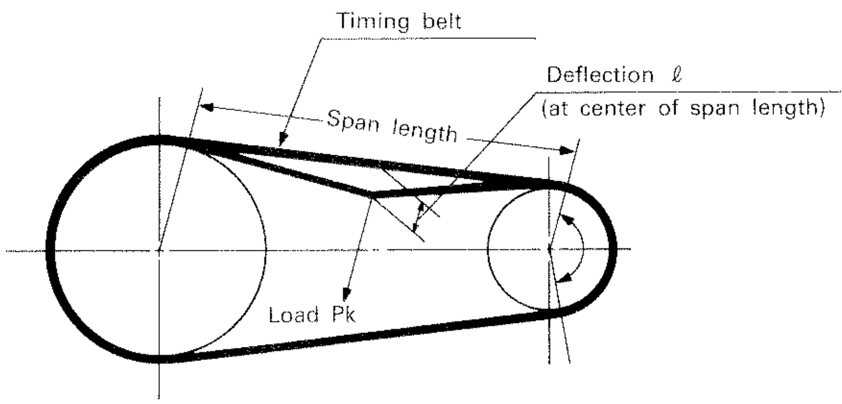


Fig. 4.3.13 Belt Deflection and Load

	Deflection ℓ	Load Pk
Shoulder	2.7 mm	22 to 37 gf
Elbow	1.6 mm	22 to 37 gf
Wrist pitch	1.2 mm	11 to 19 gf

Table 4.3.1 Belt Deflection and Load

3.5 Replacement of the Curled Cables

The curled cables used as the cables connecting the RV-M1 base unit and arm unit must normally be replaced every 2,000 hours of operation.
Two curled cables are used, power and signal. These two cables must be replaced at the same time.

A. Replacement

(1) Fig. 4.3.14 outlines the replacement procedure of the two curled cables.

(2) Make sure that the power switch on the drive unit rear panel is in the OFF position.

(3) Remove the base side covers and upper and lower shoulder covers by referring to 3.2 Removal of the Covers.

(4) Unplug the four connectors (connector names: "CON2", "CON3", "CON4" and "CON3") which are plugged into position inside the base ①.

(5) Unplug the eight connectors (connector names: "PW," "HND," "M2," "M3," "ENC2," "ENC3," "ENC4, 5," and "LS") which are plugged into position inside the shoulder covers.

(6) Each connector is identified by the connector name printed on the marking seal.

(7) Remove the two mounting screws ② and pull off the terminal plate ③ together with the power curled cable ④ and signal curled cable ⑤ from the base ①.

(8) Remove the plastic nuts ⑦, each of which secures the power curled cable ④ and signal curled cable ⑤, respectively, to the motor plate ⑥.

(9) Pull off the two cables ④ and ⑤ from the motor plate ⑥.

(10) To install new cables, reverse the order of removal.

WARNING

When reconnecting the connectors, be sure they are connected in matched pairs identified by the same connector name. A wrong connection can result in a serious trouble.

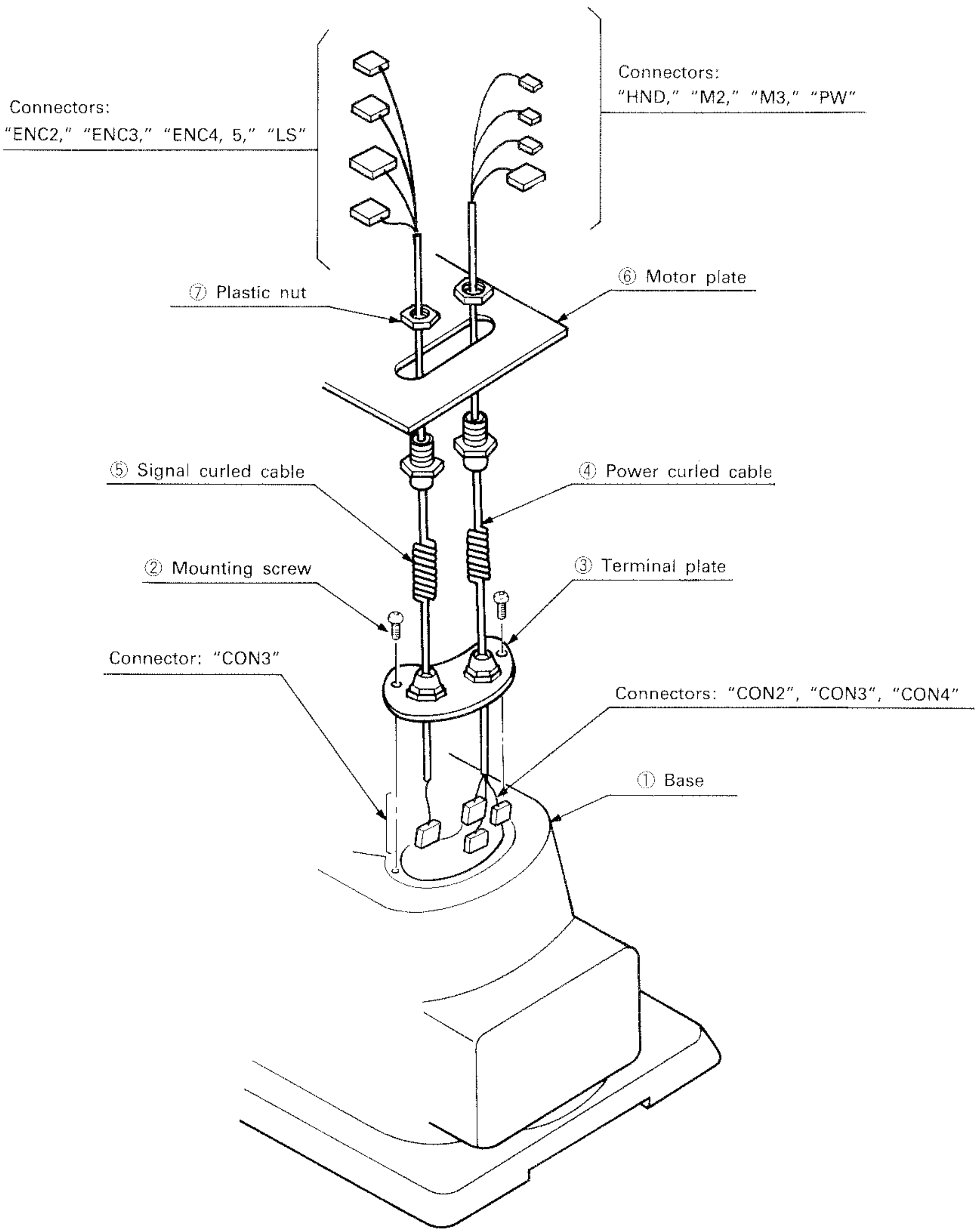


Fig. 4.3.14 Replacing the Curled Cables

3.6 Inspection, Adjustment, and Replacement of the Brakes

The RV-M1 robot has built-in electromagnetic brakes installed in the shoulder swivel and elbow extension axes that prevent the arm from dropping by its own weight when power is OFF. Inspect the brakes at regular intervals by following the inspection and adjustment procedures given below.

3.6.1 Checking, adjusting, and replacing the shoulder brake

A. Inspection

(1) During the normal operation of the robot, check to see if any of the following symptoms develops.

a. The brake is not released when power is turned ON.
  - This is probably due to an open brake coil lead wire or incorrect armature-to-coil gap. If the problem persists even after the gap has been adjusted, replace the brake.

b. The brake squeaks while the arm is being rotated.
  - This is probably due to incorrect gap. Adjust the gap as necessary.

c. The arm drops when power is turned OFF.
  - This is probably due to a worn brake lining or oil and grease trapped inside. Clean or replace the brake lining.

B. Adjustment

(1) Fig. 4.3.15 identifies the shoulder brake parts.

(2) Make sure that the power switch on the drive unit rear panel is in the OFF position.

(3) Remove the upper and lower shoulder covers by referring to 3.2 Removal of the Covers.

(4) Loosen the two setscrews ③. At this time, use care not to lose the copper plates ⑪. Then, using a feeler gauge, adjust the gap between the armature ⑤ and coil ⑦ to 0.1 mm.

(5) After the gap has been properly adjusted, tighten the two setscrews ③ securely. At this time, make sure that the copper plate ⑪ is installed between the setscrew ③ and shaft ⑧.

C. Replacement

(1) Fig. 4.3.16 outlines the replacement procedure of the shoulder brake.

(2) Make sure that the power switch on the drive unit rear panel is in the OFF position.

(3) Remove the upper and lower shoulder covers by referring to 3.2 Removal of the Covers.

(4) The brake connector ① (connector name: "PW") is plugged into position inside the shoulder covers.

(5) Unplug the shoulder brake pin contacts ② (contact Nos. 1 and 2) from the receptacle housing in the connector ①. See Fig. 4.3.17.

(6) Remove the two setscrews ③. At this time, use care not to lose the copper plates ⑪. Then, remove the brake disk ④, armature ⑤, and spring ⑥ from the shaft ⑧.

(7) Remove the four mounting screws ⑨ to remove the coil ⑦ from the brake mounting plate ⑩.

(8) To install a new brake, reverse the order of removal.

(9) Adjust the gap by referring to the procedure given in B.

(10) The pin contacts ② may be plugged into either position, No. 1 and No. 2.

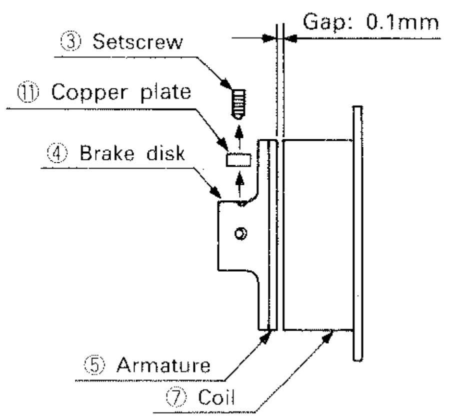


Fig. 4.3.15 Shoulder Brake Parts Identification

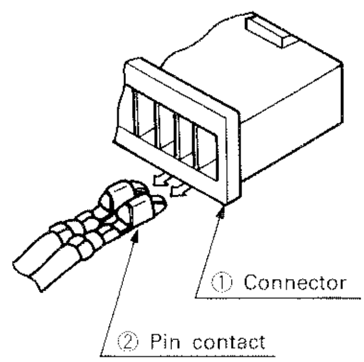


Fig. 4.3.17 Removing the Pin Contacts

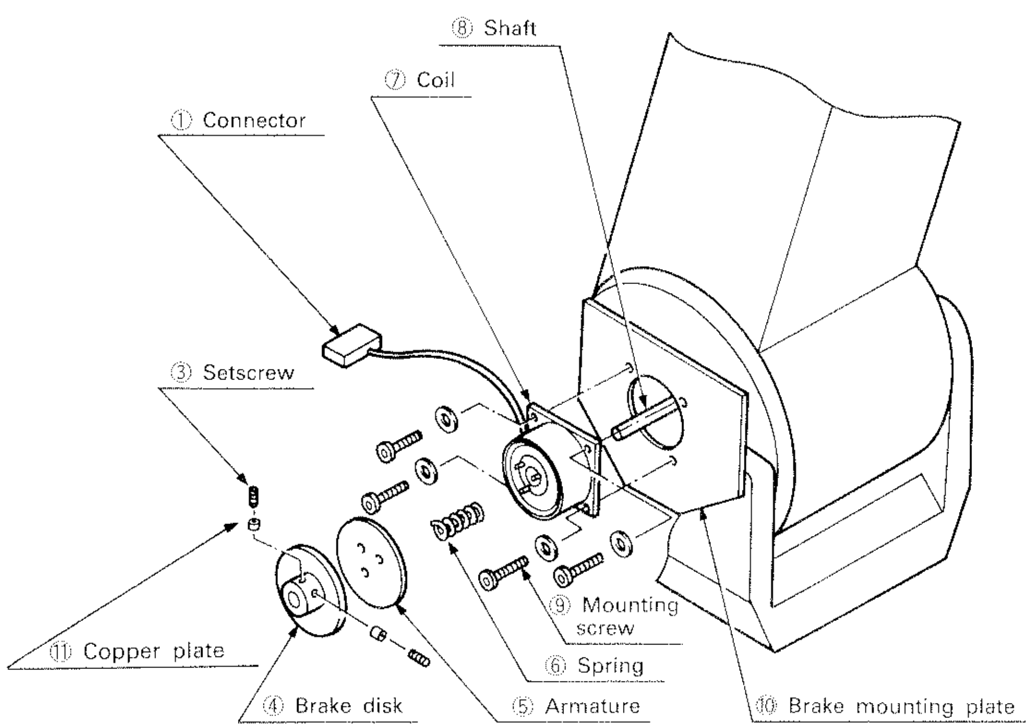


Fig. 4.3.16 Replacing the Shoulder Brake

3.6.2 Checking, adjusting, and replacing the elbow brake

A. Inspection

(1) During the normal operation of the robot, check to see if any of the following symptoms develops.

a. The brake is not released when power is turned ON.
   - This is probably due to an open brake coil or lead wire or incorrect armature-to-coil gap. If the problem persists even after the gap has been adjusted, replace the brake.

b. The brake squeaks while the arm is being rotated.
   - This is probably due to incorrect gap. Adjust the gap as necessary.

c. The arm drops when power is turned OFF.
   - This is probably due to a worn brake lining or oil and grease trapped inside. Clean or replace the brake lining.

B. Adjustment

(1) Fig. 4.3.15 identifies the elbow brake parts.

(2) Make sure that the power switch on the drive unit rear panel is in the OFF position.

(3) Remove the upper and lower shoulder covers by referring to 3.2 Removal of the Covers.

(4) Loosen the two setscrews ③. At this time, use care not to lose the copper plates ⑪. Then, using a feeler gauge, adjust the gap between the armature ⑤ and coil ⑦ to 0.1mm.

(5) After the gap has been properly adjusted, tighten the two setscrews ③ securely. At this time, make sure that the copper plate ⑪ is installed between the setscrew ③ and shaft ⑧.

C. Replacement

(1) Fig. 4.3.18 also outlines the replacement procedure of the elbow brake.

(2) Make sure that the power switch on the drive unit rear panel is in the OFF position.

(3) Remove the upper and lower shoulder covers by referring to 3.2 Removal of the Covers.

(4) The brake connector ① (connector name: "PW") is plugged into position inside the shoulder covers.

(5) Unplug the elbow brake pin contacts ② (contact Nos. 3 and 4) from the receptacle housing in the connector ①. See Fig. 4.3.19.

(6) Remove the two setscrews ③. At this time, use care not to lose the copper plates ⑪. Then, remove the brake disk ④, armature ⑤, and spring ⑥ from the shaft ⑧.

(7) Remove the four mounting screws ⑨ to remove the coil ⑦ from the brake mounting plate ⑩.

(8) To install a new brake, reverse the order of removal.

(9) Adjust the gap by referring to the procedure given in B.

(10) The pin contacts ② may be plugged into either position, No. 3 and No. 4.

WARNING

The arm falls down when the brakes are removed. Be sure to secure the arm in position before attempting to adjust or replace the brakes.

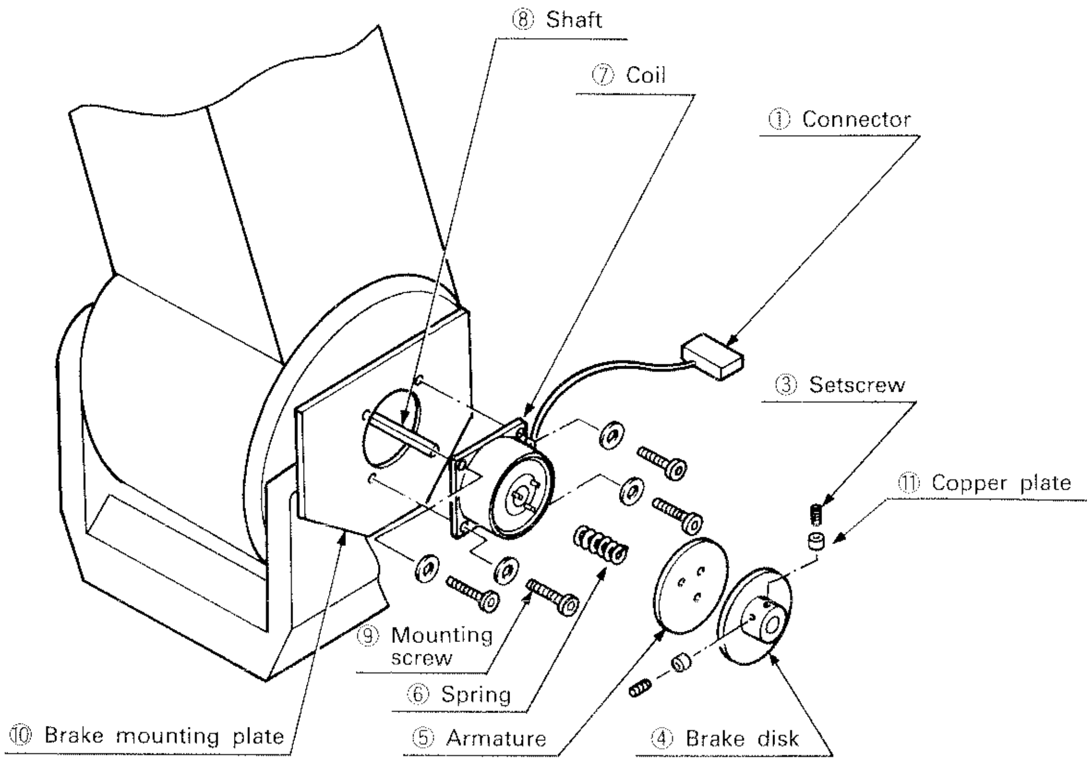


Fig. 4.3.18 Elbow Brake Parts Identification

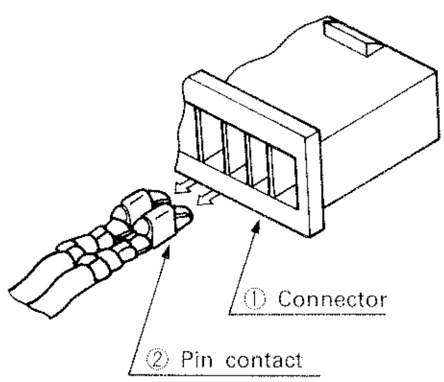


Fig. 4.3.19 Removing the Pin Contacts

3.7 Checking the Limit Pulses

To achieve the correct origin setting, the limit switches and encoder Z phase pulses of all axes should have been adjusted to the appropriate relation. This relation is represented by limit pulses.
The limit pulses are adjusted to the correct value before shipment from the factory. After long use, however, the limit pulses may change because the limit switches will be loose and deteriorate with age. The limit pulses should be within the specified ranges to maintain the robot accuracy. If the limit pulses of any axis exceed its allowed range, operation fault may occur during origin setting or the required position may not be reached after origin setting.
The limit pulses can be checked in the following procedure.
(This operation cannot be performed without the teaching box.)

1) Connect the teaching box to the drive unit and switch power on. Then perform origin setting in accordance with Section 4.3 "Origin Setting."

2) To monitor the required axis status after setting the teaching box ON/OFF switch to ON, press the keys as follows:

   [NST] [9] [9] [*] [ENT]

   * stands for the axis number. For example, type [9], [9], [1] to monitor the status of the J1 axis, and type [9], [9], [2] to monitor that of the J2 axis.

3) At this time, the teaching box LED indicates the limit pulses of the specified axis. Check that the limit pulses are within the corresponding range indicated in Table 3.4.2.

Axis	Minimum Value	Maximum Value
J1	30	170
J2	30	170
J3	30	170
J4	18	78
J5	18	78

Table 3.4.2 Permissible Limit Pulse Ranges

Consult our Service Division if any axis is outside its limit pulse range.

4. SERVICE PARTS

Table 4.4.1 lists those RV-M1 parts which require periodic replacement. Tables 4.4.2 and 4.4.3 are the listings of spare parts that may be required during service procedures.
All parts may be ordered from the authorized manufacturer or our Service Division. Note that the parts recommended by us differ from manufacturer's standard parts: by sure to consult our Service Division before ordering.

4.1 Robot Consumables

No.	Description	Model	Manufacturer	Installed Location	Qty
1	Motor brush	Note1 30W-BRSH-☐	Mitsubishi Electric	Waist, elbow, shoulder; 2 each	6
2	DC servomotor	11W-J4M 11W-J5M		Wrist pitch/roll	2
3	Timing belt	MM-195-9.5	Mitsuboshi Belt	Shoulder	1
4	Timing belt	MM-132-9.5		Elbow	1
5	Timing belt	MM-97-4.8		Wrist pitch	1
6	Curled cable	BU143C243G51	Mitsubishi Electric	Power/signal relay	1 set

Table 4.4.1 Robot Consumables List

Note 1: When ordering the motor brush, specify [D] for the sliver motor frame and [S] for the black frame.

4.2 Robot Spare Parts

No.	Description	Model	Manufacturer	Installed Location	Qty Note 1
1	DC servomotor	30W-J1M 30W-J2M 30W-J3M	Mitsubishi Electric	Waist, elbow, shoulder; 1 each	3
2	Harmonic drive reduction gear	BU143C172H02 (CS-14-100-2A-GR)	Mitsubishi Electric (Harmonic Drive Systems)	Base	1
3	Harmonic drive reduction gear	BU144D555H01 (SH-20-100-2A-GR-SP)		Shoulder	1
4	Harmonic drive reduction gear	BU144D554H01 (SH-14-110-2A-GR-SP)		Elbow	1
5	Harmonic drive reduction gear	BU144D413H01 (FB-14-110-2-BL3-SP)		Wrist pitch	1
6	Harmonic drive reduction gear	BU144D366H01 (FB-14-110-2-BL3)		Wrist roll	1
7	Limit switch assembly	LS-ASSY-[ ] Note 2: Shaft designation	Mitsubishi Electric	Origin-setting signal for all axes	5 sets
8	Electromagnetic brake assembly	BU144D473G51	Mitsubishi Electric	Shoulder, elbow	2
9	Roller follower	NART-6EUUV	THK CO., LTD	Elbow drive link	2
10	Relay card (SIGNAL)	TU811A038G61	Mitsubishi Electric	Relay in base	1
11	Relay card (POWER)	TU811A042G61	Mitsubishi Electric	Relay in base	1
12	Miniature bearing	FL696ZZ	NTN CO., LTD	Elbow	1
13	Miniature bearing	FLW686ZZ	NTN CO., LTD	Elbow	1
14	Miniature bearing	FL626ZZ	NTN CO., LTD	Shoulder	2

Table 4.4.2 Robot Spare Parts List

Note 1: Qty per unit
Note 2: J1 to J5

4.3 Drive Unit Spare Parts

No.	Description	Model	Manufacturer	Installed Location	Qty
1	Fuse	MF60NR10A-05 (250V AC, 10A)	TOYO	Fuse holder in rear of drive unit	1

Note: 10A for 120V line voltage; 5A for 220, 230 or 240V line voltage.

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 shows the pin assignments for the Centronics connector.

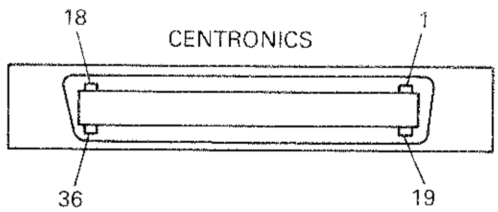


Fig. 5.1.1 Centronics Connector

Pin No.	Signal	Pin No.	Signal
1	$\overline{\text{STB}}$	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	$\overline{\text{ACK}}$	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 model: 57LE-40500-77OO (D3). . . (DDK)
Applicable plug model: 57E-30500. . . (DDK)

1.2 Function of Each Signal Line

Signal	Direction	Function
DB0 to 7 (Data bit)	In	These signals represent information of the 8 bits parallel data, respectively, output from the personal computer.
$\overline{\text{STB}}$ (Strobe)	In	This is the pulse signal indicating that input data from the personal computer is present to be read. Data is read in on the LOW-going negative edge.
BUSY (Busy)	Out	This signal goes LOW when the drive unit is ready for receiving data. It goes HIGH indicating that the drive unit is unable to receive data.
$\overline{\text{ACK}}$ (Acknowledge)	Out	A LOW 2 to 3 μ s pulse indicates that data has been received and the drive unit is ready to accept other data.
GND (Ground)	—	Ground connection for the above signal lines.

Table 5.1.2 Functions of Signal Lines on Centronics Connector

1.3 Signal Line Timing Chart

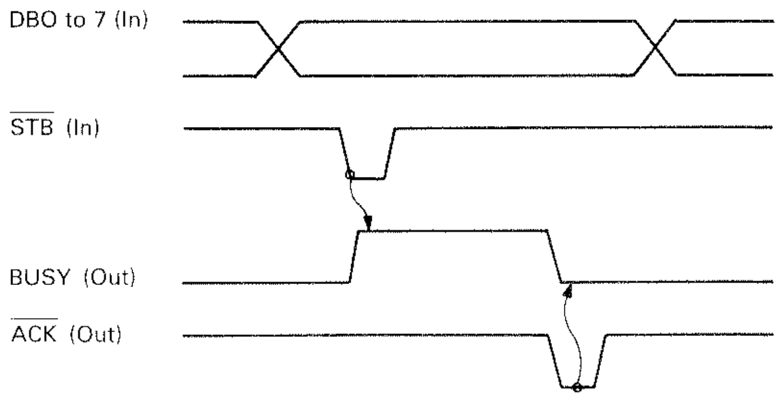


Fig. 5.1.2 Timing Chart

(1) Data from the personal computer is transferred to DB0 to DB7.

(2) The drive unit outputs a HIGH BUSY signal on the negative edge of a $\overline{\text{STB}}$ signal from the personal computer.

(3) When the drive unit has completed receiving data and is ready to accept other data, an $\overline{\text{ACK}}$ signal is sent to the personal computer; at the same time, the BUSY signal goes LOW.

(4) This operating sequence repeats until the terminator, hex. "0D," 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.
Also available are the optional cables for the Mitsubishi MULTI16 and NEC PC9801.

1.5 Centronics Interfacing Examples

Example 1: Using Mitsubishi MULTI16Ⅲ

1) Connecting the Centronics cable

Using the optional cable for Centronics (model: C-MULTI-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer turned OFF). For the connector on the MULTI16Ⅲ, use its built-in printer connector.

2) Setting the personal computer

(Setting switches)

Set the I/O mode switch (I-SW) on the back panel of the personal computer as follows.

Bit 1: In lower position (OFF)
Bit 2: In upper position (ON)
Bit 3: In lower position (OFF)

After the above switch settings have been made, turn ON the personal computer.

(Invoking software)

(1) Activate CP/M-86.
(2) Activate the GENSYS command to make the following settings.

A>GENSYS ↲	Activate GENSYS.
⁞	
Printer? Y ↲	Set printer configuration.
Type? 24 pin 16 ↲	Set 16-pin type.
⁞	
END OF GENSYS (Y/N)? Y ↲	End GENSYS.

(3) Activate MBASIC.

A>MBASIC	Activate MBASIC (or MBASIC2).

3) Setting the drive unit

After the drive unit has been turned ON, place ST1 located inside the side door in the lower position (OFF). This selects the personal computer mode. Make sure that the teaching box is turned 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 (model: C-PC-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer turned OFF).

2) Setting the personal computer

No special settings are required on the personal computer. After N88DISK-BASIC has been activated, turn 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.

2. INTERFACE WITH THE PERSONAL COMPUTER (RS232C)

2.1 RS232C Connector Pin Assignments

Table 5.2.1 shows the pin assignments for the RS232C connector.

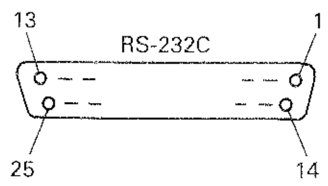


Fig. 5.2.1 RS232C Connector

Pin No.	Signal	Pin No.	Signal
1	FG	14	N. C
2	SD (TXD)	15	N. C
3	RD (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

Applicable connector: JAEDB-25P (male) or equivalent (JAE)

N. C: Not connected

2.2 Function of Each Signal Line

Signal	Direction	Function
FG	—	Frame ground connected to the FG terminal on the drive unit.
SD (TXD)	Out	Provides the lines on which the drive unit presents data to the personal computer.
RD (RXD)	In	Provides the lines on which the personal computer presents data to the drive unit.
RS (RTS)	Out	This signal must be set whenever the personal computer wishes to transmit data.
CS (CTS)	In	This signal is used to authorize the drive unit to transmit data.
DR (DSR)	In	This signal is used to indicate that the personal computer is ready to transmit and receive data.
SG	—	Signal ground for data and control lines.
ER (DTR)	Out	This signal is 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 to be used, the following communication condition settings must be made on 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 paragraphs explain the settings made on the drive unit. For personal computer settings, see Instruction Manual of the personal computer.

(1) Baud rate setting

The baud rate can be set with SW3 located inside the drive unit side door. Place the corresponding bit in the upper position (ON) according to the desired baud rate. (The underlined baud rate in Table 5.2.3 is the standard setting.)

WARNING

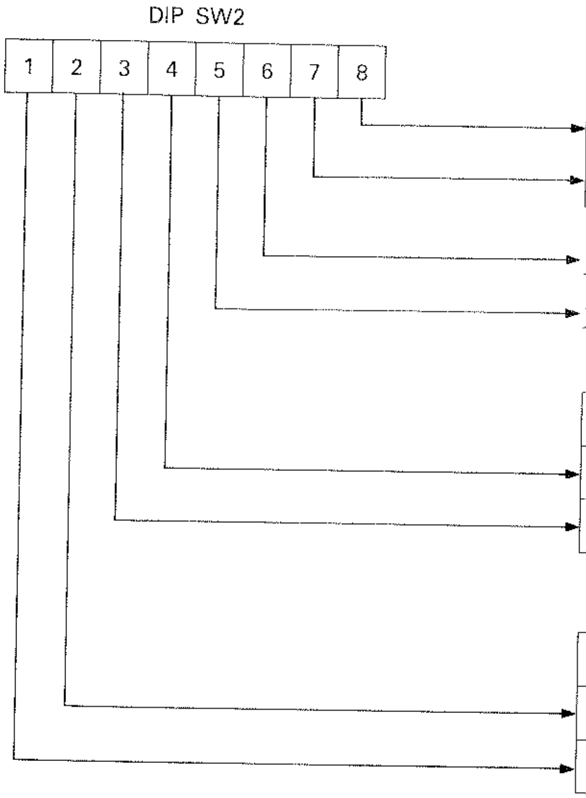
NEVER set to ON two or more bits at one time while the drive unit is ON.

DIP SW3 Bit No.	Baud rate factor		
	×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

Transfer format can be set with SW2 located inside the drive unit side door. Set the bits as required according to the desired format. (The underlined settings are standard.)



No. of stop bits

Null	1 bit	1-1/2 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 pre-scribing 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 the drive unit.

1) Data transfer timing from personal computer to robot

〈Robot〉

1st character:
Signals ER (DTR) and RS (RTS) are turned to HIGH for up to 7ms and 177ms. If any data is input during this period, both ER (DTR) and RS (RTS) are turned to LOW allowing the data to be read in.

2nd character and onward:
The drive unit causes both ER (DTR) and RS (RTS) to go HIGH to be ready for accepting data entry. If any data is input, ER (DTR) and RS (RTS) are turned to LOW allowing the data to be read in. This operation repeats until a hex. "0D" code (CR: carriage return) or a hex, "0A" code (LF: line feed) is input. While the robot is executing the command, both ER (DTR) and RS (RTS) are low.

〈Personal computer〉

Transfer the 1st character while DR (DSR) remains HIGH. Transfer the 2nd and subsequent characters when DR (DSR) later goes LOW and again goes HIGH. When DR (DSR) is high, transferring a series of characters causes an error in the robot.

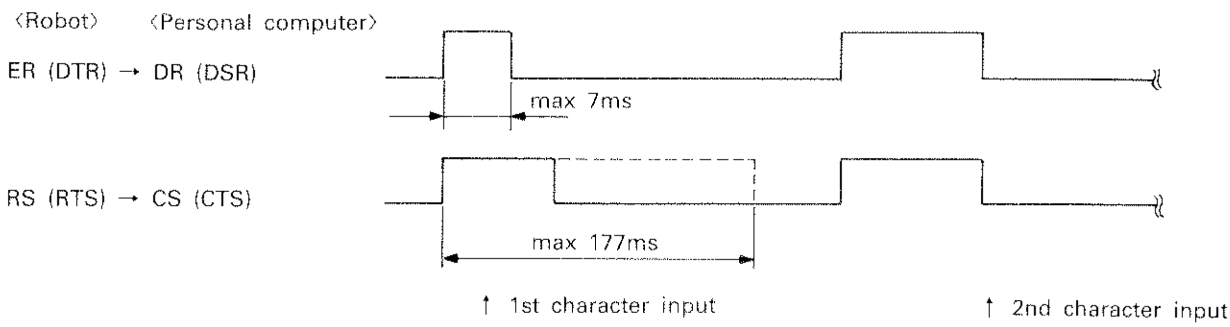


Fig. 5.2.2 Data Transfer Timing from Personal Computer to Robot

2) Data transfer timing from robot to personal computer

〈Robot〉

Data transfer is initiated after causing ER (DTR) to go HIGH. ER (DTR) is turned to LOW after the last hex. "0D" code has been transferred.

〈Personal computer〉

The personal computer causes RS (RTS) to go HIGH waiting 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 position (ON).

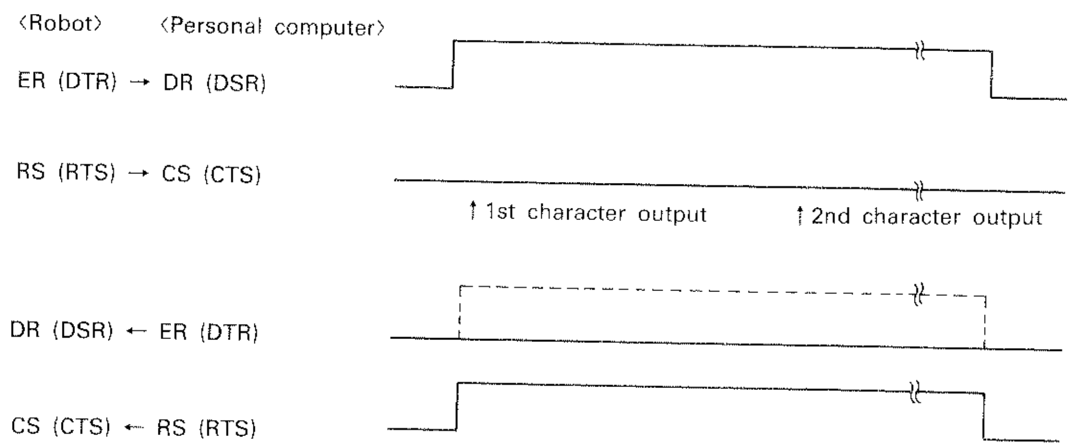


Fig. 5.2.3 Data Transfer Timing from Robot to Personal Computer

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. 5.2.5. Each signal line must, however, meet the timing described earlier.
Also available are the optional cables for the Mitsubishi MULTI16 and NEC PC9801.

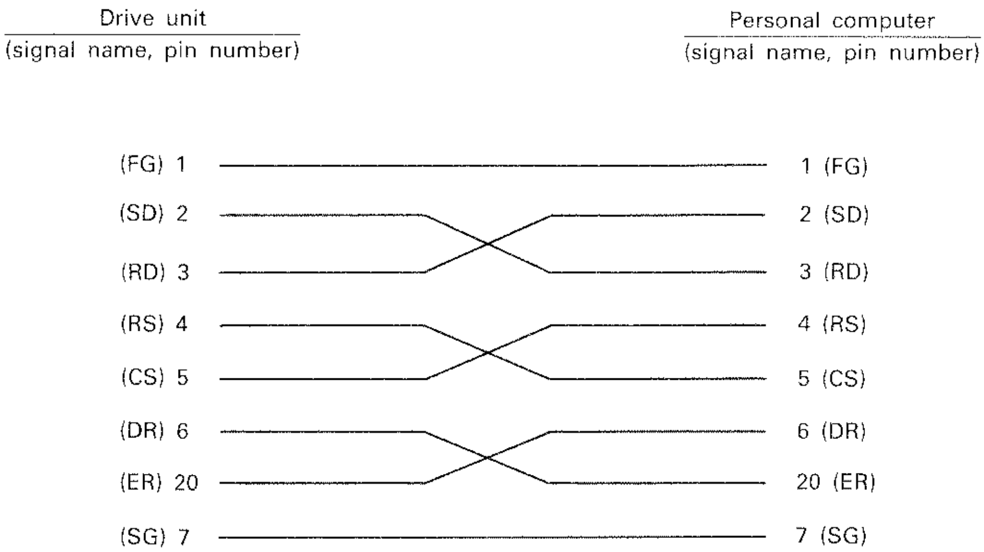


Fig. 5.2.4 RS232C Cable Connection

2.6 RS232C Interfacing Examples

Example 1: Using Mitsubishi MULTI16Ⅲ

1) Connecting the RS232C cable

Using the optional cable for RS232C (model: RS-MULTI-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer turned OFF). For the connector on the MULTI16Ⅲ, use its built-in RS232C connector.

2) Setting the personal computer

(Setting 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 position (OFF).

After the above switch settings have been made, turn ON the personal computer. (For location of S3, see MULTI16Ⅲ Owner's Manual.)

(Invoking software)

(1) Activate CP/M-86.

(2) Activate the GENSYS command to make the following settings.

A > GENSYS ↵ ⁝	Activate GENSYS.
RS232C? Y ↵ Async/Sync? ASYNC ↵ ⁝	Set RS232C configuration.
END OF GENSYS (Y/N)? Y ↵	End GENSYS.

(3) Activate MBASIC.

A > MBASIC ↵	Activate MBASIC (or MBASIC2).

3) Setting the drive unit

After the drive unit has been turned ON, place ST1 located inside the side door in the lower position (OFF). This selects the personal computer mode. Make sure that the teaching box is turned 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 2.3 RS232C Settings.

4) Verifying proper connection

Input and run the following program in MBASIC.

```
10 OPEN "COM1 :9600, E, 7, 2" AS#1
20 PRINT #1, "NT"
30 END                         Above program is executed.
   RUN ↲
   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 (model: RS-PC-CBL), connect the drive unit to the personal computer (with the drive unit and personal computer turned OFF).

2) Setting the personal computer

(Setting switches)

Set DIP switches SW1 and SW2 on the back panel as follows.

SW1: Place bits 7 and 9 in the lower position (ON).
Place bits 8 and 10 in the upper position (OFF).
SW2: Place bit 5 in the lower position (ON).

After the above switch settings have been made, turn ON the personal computer.

(Invoking software)

(1) Activate N88DISK-BASIC.

(2) Set memory switch as follows.

```
OK
mon ↲                Activate machine language monitor.
h]ssw2 ↲
08-08 ↲              Set baud rate (at 9,600 baud).
```

(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

Input and run the following program in N-BASIC.

```
10 OPEN "COM: E73" AS#1
20 PRINT #1, "NT"
30 END

   RUN ↓                          Above program is executed.
   OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

Example 3: A6GPP (Mitsubishi MELSEC-A series general-purpose programmable controller)

1) Connecting the RS232C cable

Using the optional cable for RS232C (RS-PC-CBL), connect the drive unit to the corresponding A6GPP connector (with the drive unit and personal computer turned off.)

2) Setting the A6GPP

(1) Insert the optional 3.5-inch BASIC disk (SW0GHP-BAS) into drive A of the A6GPP and power up the A6GPP.

(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

Input and run the following program in BASIC.

```
10 INIT %1, "6E72"1
20 PRINT %1, "NT"
30 END
   RUN ↓                          Above program is executed.
   OK
```

If the robot performs correct return-to-origin operation, the connection has been made properly.

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.
There are four types of I/O cards available, A8, B8, A16, and B16, each having the following specifics.

Item / Type	General-purpose I/O		Dedicated I/O		Remarks	ID marking (silk-screen-printed)
	Input	Output	Input	Output		
A8	8 points $\overline{\text{STB}}$ $\overline{\text{BUSY}}$	8 points $\overline{\text{RDY}}$ ACK	Not available	Not available	Operation by means of external signals is not possible.	#2A
B8	8 points STB BUSY	8 points RDY $\overline{\text{ACK}}$	Not available	Not available	Operation by means of external signals is not possible.	#2B
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 by means of external signals is possible.	#2C
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 Cards

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 shows the pin assignments for the external I/O connector on type A I/O card. "Wire colors" shown in the Table are keyed to those shown in 3.6 External I/O Cable. Those signals marked with * are not available in type A8 I/O card.

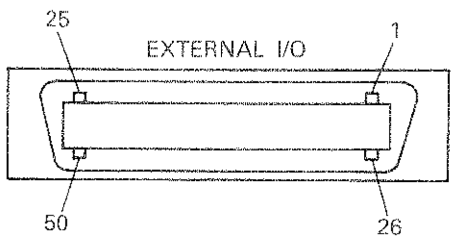


Fig. 5.3.1.A External I/O Connector

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	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	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.A External I/O Connector Pin Assignments (Type A8/16)

Receptacle model: 57LE-40360-77OO (D3). . . (DDK)
Applicable plug model: 57E-30360. . . (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" shown in the Table are keyed to those shown in 3.6 External I/O Cable. Those signals marked with * are not available in type B8 I/O card.

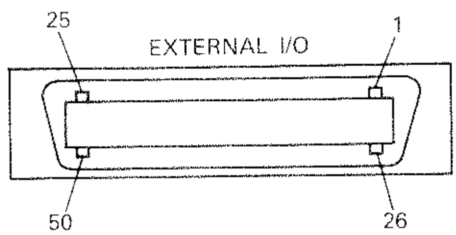


Fig. 5.3.1.B External I/O Connector

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 B8/16)

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 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.
Note that those signals marked with * are not available in type A8 I/O card.

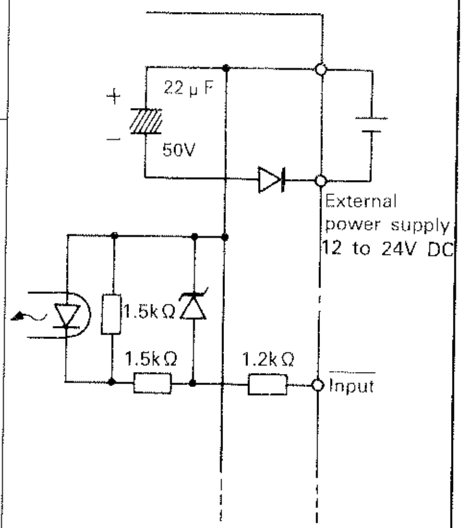
	Signal	Specifications	Internal Circuit
Input Port Block	Input port power input Input port GND output	Regulated power supply 12 to 24V DC	
	Input bits 0 to 7 *8 to 15	Input voltage ON voltage: 9V DC (min.) OFF voltage: 2V DC (max.) Input current 12V DC: 2.5mA (typ) 24V DC: 12.5mA (typ)	
	STB input		
	*START input *STOP input *RESET input		
	BUSY output	Same as output bits	Same as output bits

Table 5.3.3.A Input Port Block Specifications

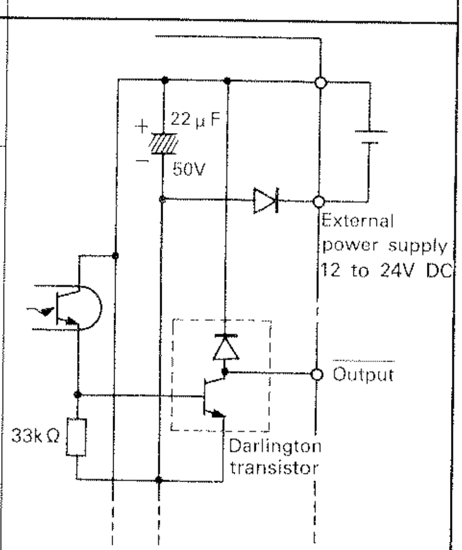
	Signal	Specifications	Internal Circuit
Output Port Block	Output port power input Output port GND output	Regulated power supply 12 to 24V DC	
	Output bits 0 to 7 *8 to 15	Max. applied voltage: 26.4V Max. load current: 0.1A/pin Max. ON voltage: 1.3V Leakage current: 100μA (max.)	
	RDY output		
	*RUN output *WAIT output *ERROR output		
	ACK input	Same as input bits	Same as 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.
Note that those signals marked with * are not available in type B8 I/O card.

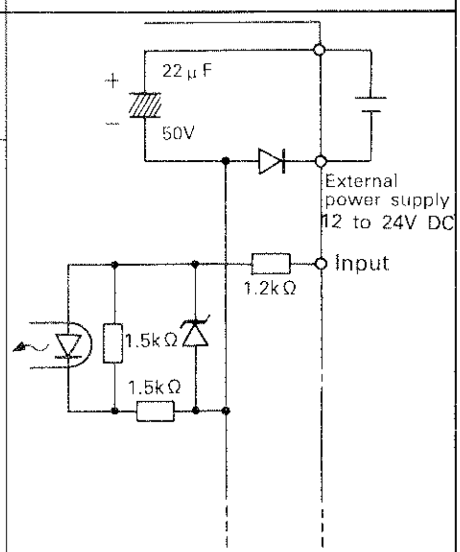
	Signal	Specifications	Internal Circuit
Input Port Block	Input port power input Input port GND output	Regulated power supply 12 to 24V DC	
	Input bits 0 to 7 *8 to 15	Input voltage ON voltage: 9V DC (min.) OFF voltage: 2V DC (max.) Input current 12V DC: 2.5mA (typ) 24V DC: 12.5mA (typ)	
	STB input		
	*START input *STOP input *RESET input		
	BUSY output	Same as output bits	Same as output bits

Table 5.3.3.B Input Port Block Specifications

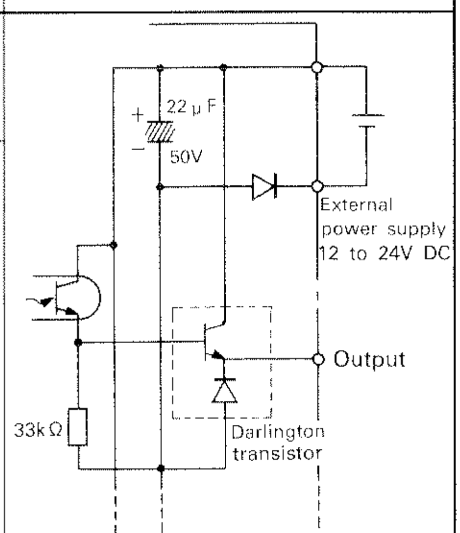
	Signal	Specifications	Internal Circuit
Output Port Block	Output port power input Output port GND output	Regulated power supply 12 to 24V DC	
	Output bits 0 to 7 *8 to 15	Max. applied voltage: 26.4V Max. load current: 0.1A/pin Max. ON voltage: 1.3V Leakage current: 100 μA (max.)	
	RDY output		
	*RUN output *WAIT output *ERROR output		
	$\overline{\text{ACK}}$ 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. Note that those signals marked with * are not available in type A8 I/O card.

	Signal	Function
Input Port Block	Input bits 0 to 7 *8 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 negative 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.A Function of Each Input Signal

	Signal	Function
Output Port Block	Output bits 0 to 7 *8 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.
	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.A Function of Each Output Signal

3.3.B Functions of I/O Signal Lines (Type B I/O Card)

The following Tables show the function of each signal line. Note that those signals marked with * are not available in type B8 I/O card.

	Signal	Function
Input Port Block	Input bits 0 to 7 *8 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 Function of Each Input Signal

	Signal	Function
Output Port Block	Output bits 0 to 7 *8 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 Function of Each Output Signal

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.

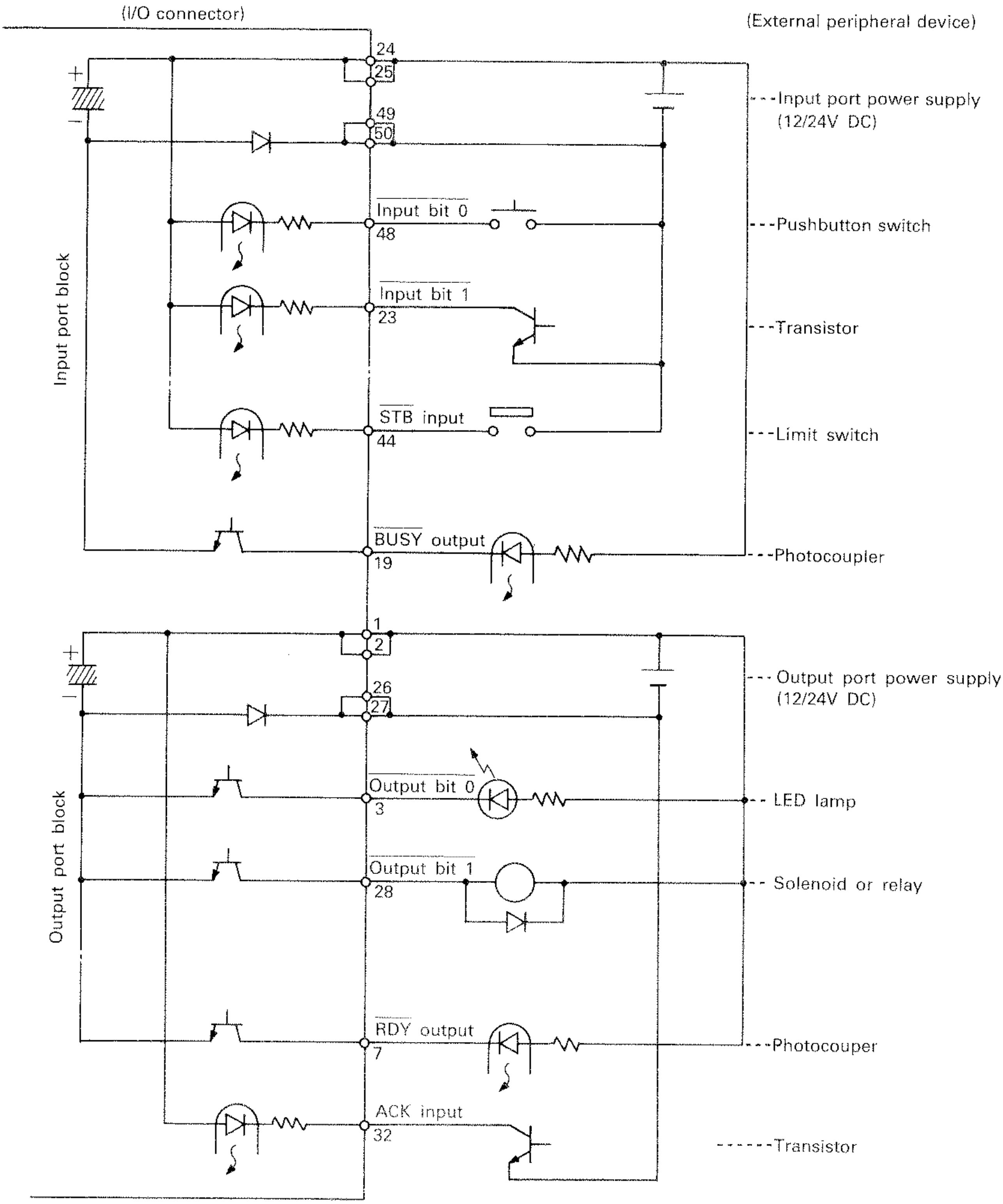


Fig. 5.3.2.A Typical I/O Circuit Connection

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.

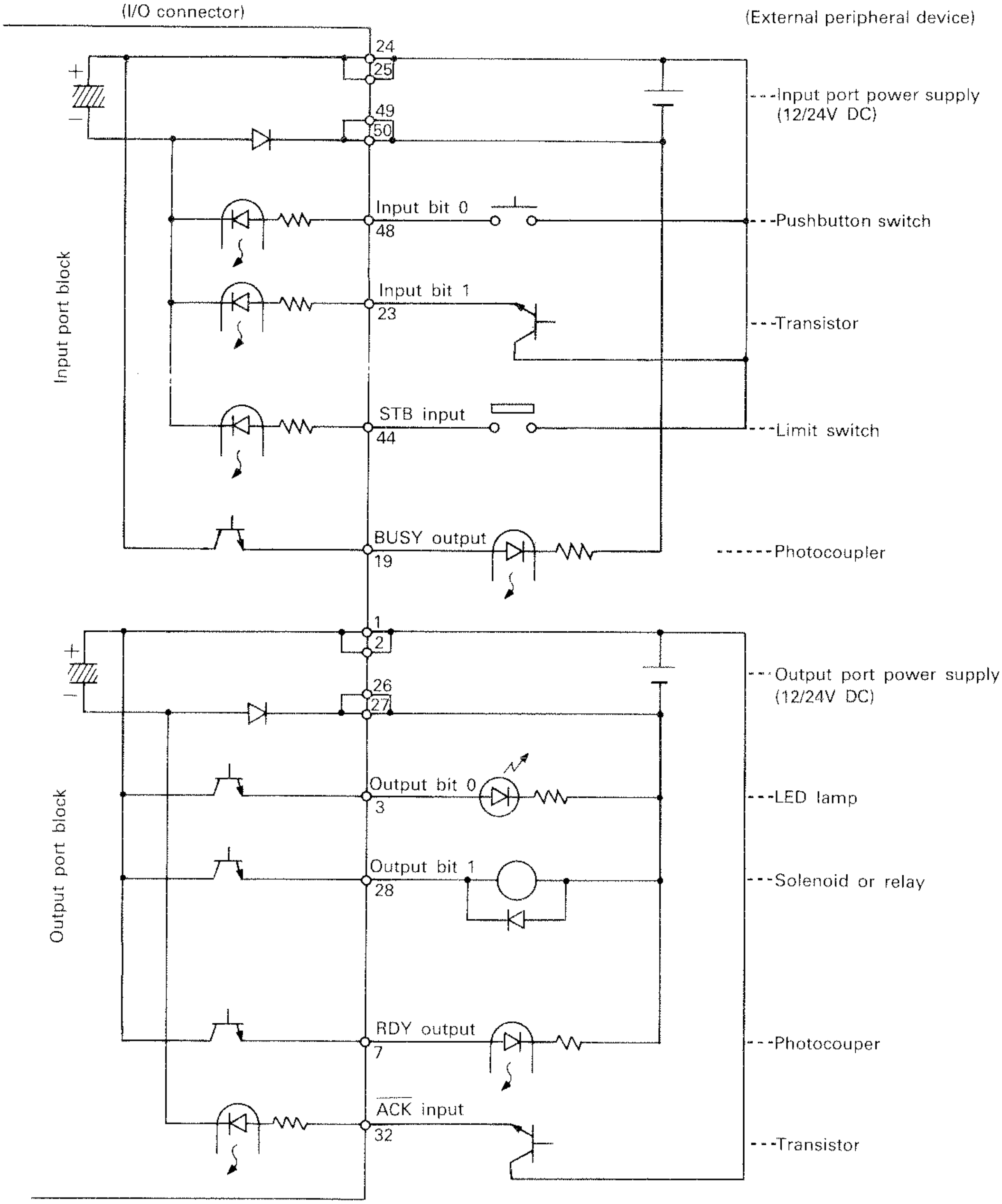


Fig. 5.3.2.B Typical I/O Circuit Connection

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."

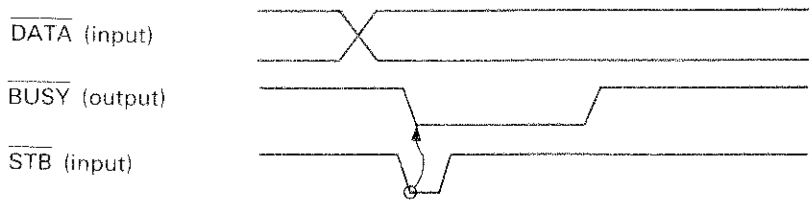


Fig. 5.3.3.A Synchronous Input Timing

(1) The drive unit is ready to receive data from the external device while the $\overline{BUSY}$ signal remains HIGH.

(2) When the external device inputs data and an $\overline{STB}$ signal, the drive unit causes the $\overline{BUSY}$ signal to go LOW and outputs the LOW signal to the external device. The external device should not change the data while the $\overline{BUSY}$ signal remains LOW.

(3) When the drive unit reads the data in, the $\overline{BUSY}$ 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."

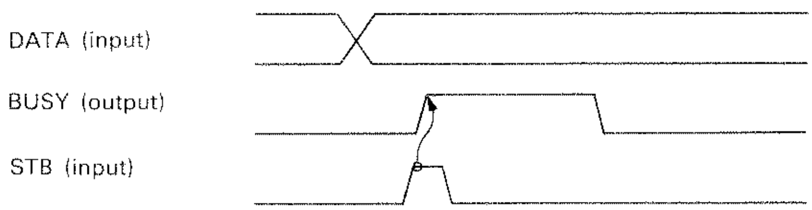


Fig. 5.3.3.B Synchronous Input Timing

(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."

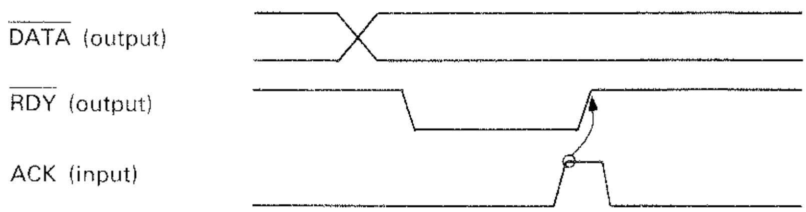


Fig. 5.3.4.A Synchronous Output Timing

(1) The drive unit outputs data as specified by parameters defined in the command "OT" while the $\overline{RDY}$ signal remains HIGH and the ACK signal remains LOW.

(2) As soon as the data is output, the drive unit causes the $\overline{RDY}$ 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 $\overline{RDY}$ 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."

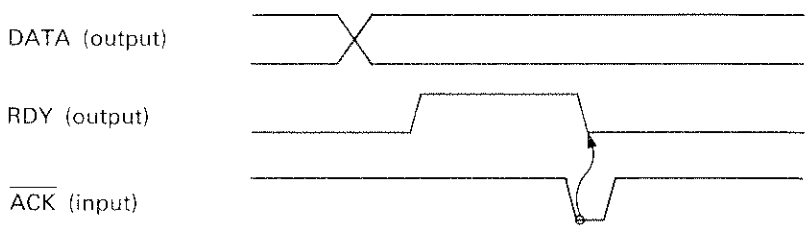


Fig. 5.3.4.B Synchronous Output Timing

(1) The drive unit outputs data as specified by parameters defined in the command "OT" while the RDY signal as well as the $\overline{ACK}$ 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 $\overline{ACK}$ 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 (Type A I/O card)

In a robotic operation environment on the actual production floor, operation is possible (if type A16 I/O card is being used) through the dedicated signal lines of the I/O connector, instead of using the front control switches on the drive unit.
Following describes the necessary settings to be made and signal timing involved in this operation.

(1) Make the following switch settings by referring to 2.1.2 Functions of side setting switches and LEDs, OPERATION.

Bits 3 and 5 of SW1: Upper position (ON)

(2) When the above settings have been made, drive unit front control switches are disabled except for the emergency stop switch.

(3) In these settings, the program can be started, stopped, restarted, and reset (including resetting of error mode II) by means of 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) Output from the dedicated outputs are the signals corresponding to the robot operating conditions (RUN, WAIT, and ERROR outputs). These signals are output in either setting, whether in external operation or front control switch operation mode. Fig. 5.3.5.A shows the dedicated I/O timing.

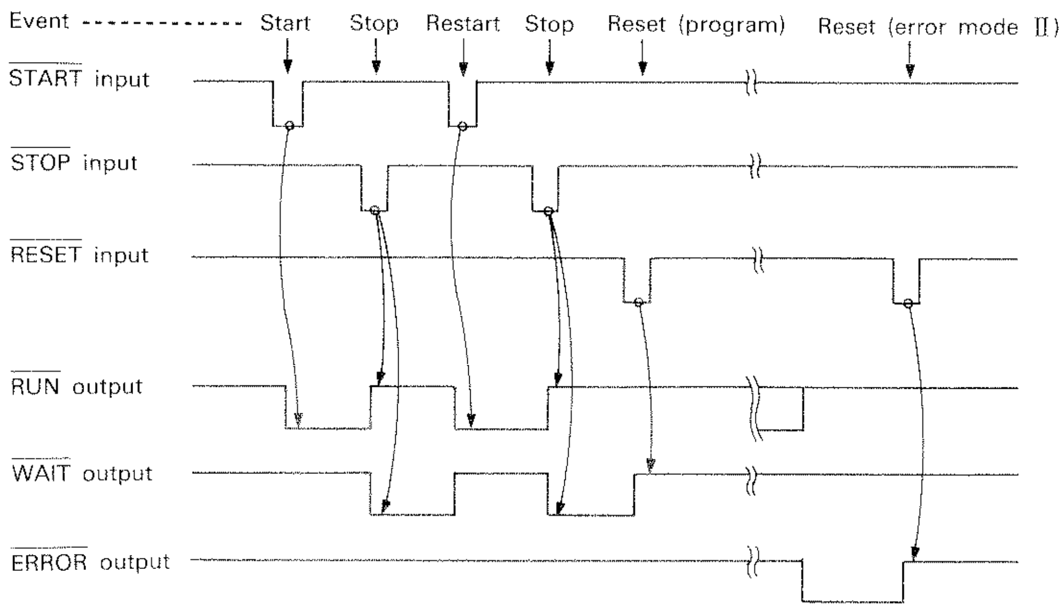


Fig. 5.3.5.A Dedicated I/O Timing

3.5.3.B Dedicated I/O timing (Type B I/O card)

In a robotic operation environment on the actual production floor, operation is possible (if type B16 I/O card is being used) through the dedicated signal lines of the I/O connector, instead of using the front control switches on the drive unit.

Following describes the necessary settings to be made and signal timing involved in this operation.

(1) Make the following switch settings by referring to 2.1.2 Functions of side setting switches and LEDs, OPERATION.

Bits 3 and 5 of SW1: Upper position (ON)

(2) When the above settings have been made, drive unit front control switches are disabled except for the emergency stop switch.

(3) In these settings, the program can be started, stopped, restarted, and reset (including resetting of error mode II) by means of 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) Output from the dedicated outputs are the signals corresponding to the robot operating conditions (RUN, WAIT, and ERROR outputs). These signals are output in either setting, whether in external operation or front control switch operation mode. Fig. 5.3.5.A shows the dedicated I/O timing.

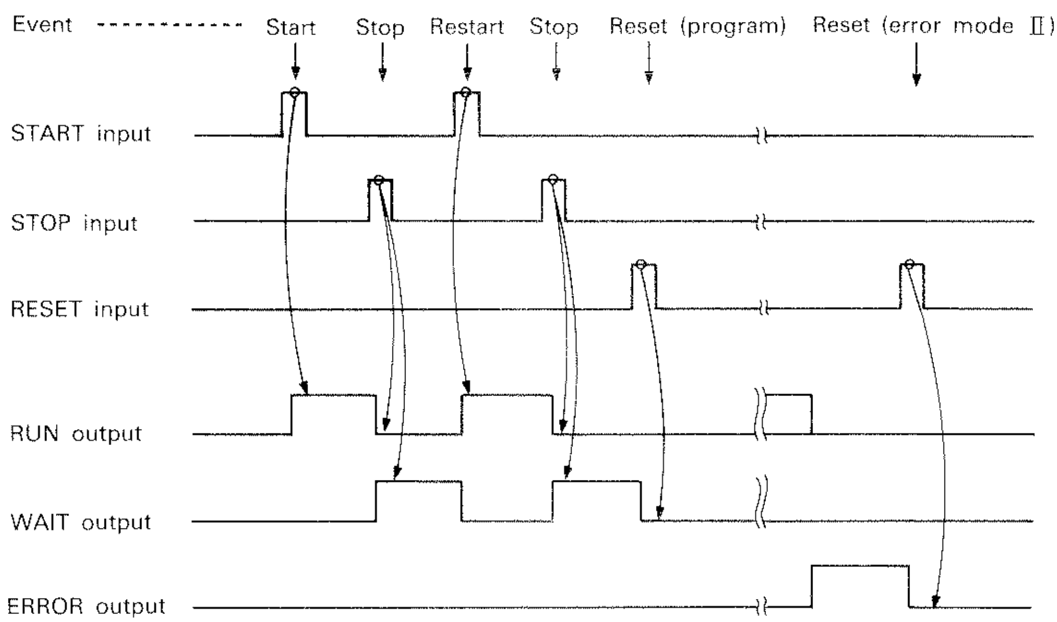


Fig. 5.3.5.B Dedicated I/O Timing

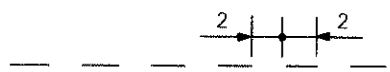
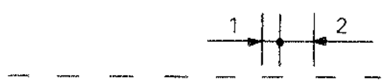
3.6 External I/O Cable

A dedicated cable is available as an option to connect the drive unit to external peripheral equipment. One end of this 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 by referring to color coding described in 3.1 External I/O Connector Pin Assignments.

Example:

Pin No.	Signal	Wire Color
21	Input bit 5	White/red A

- White: Jacket color
- red: Dot marking color
- A: Dot marking type (See below.)

Dot Marking Type	Dot Pattern
Type A	
Type B	
Type C	
Type D	

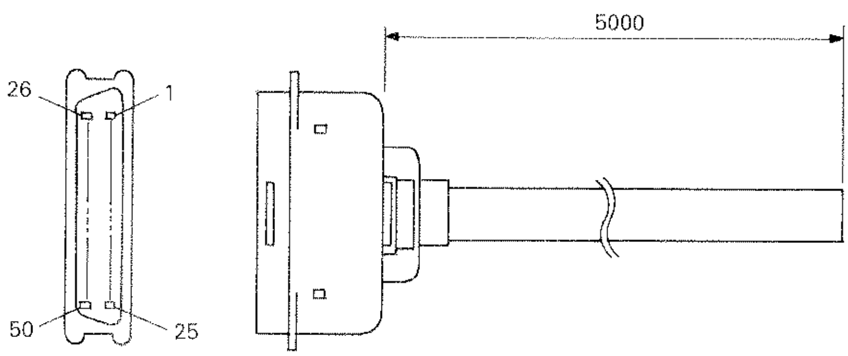


Fig. 5.3.6 External I/O Cable Marking

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) For the input signal, use the no-voltage contact signal or transistor open-collector signal.

(3) When energizing the 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 must be connected in series. See Fig. below.

(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 or ground (to protect the internal output transistor).

(8) A signal is given instantly (about 0.1 seconds) from the external output port when the drive unit is switched off. If this may misoperate the peripheral equipment, switch off the drive unit after the peripheral equipment or form a new idea to input the signal to the peripheral equipment (e.g. check the signal by entering it twice at intervals of 0.5 seconds).

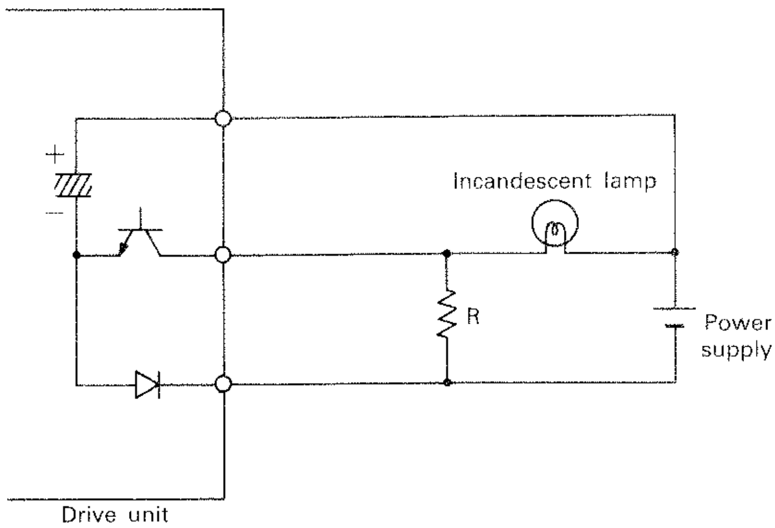


Fig. 5.3.7.A Typical Circuit Including an Incandescent Lamp (Type A I/O Card)

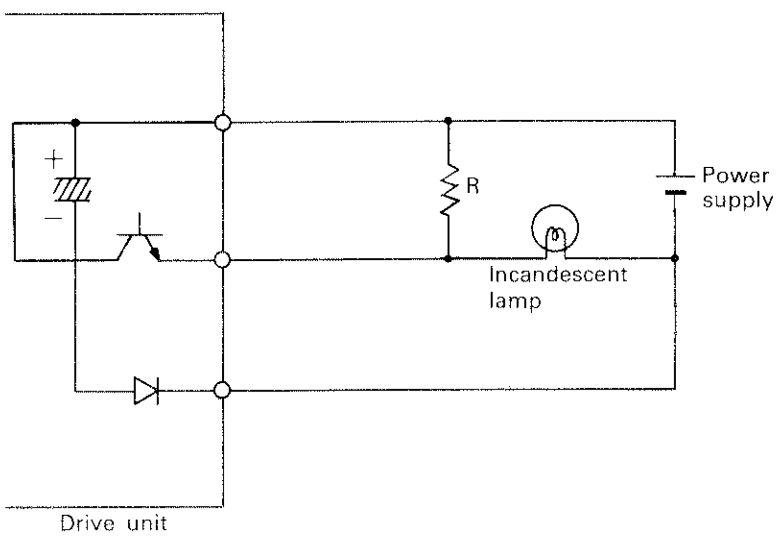


Fig. 5.3.7.B Typical Circuit Including an Incandescent Lamp (Type B I/O Card)

4. CARTESIAN COORDINATE SYSTEM REFERENCE POSITION SETTING

This chapter contains the procedure for setting the cartesian coordinate system reference position to effect as accurate axis motions as possible. It also describes the procedure for storing the reference position data in the EPROM.

4.1 Moving the Robot to Reference Position

(1) Secure the robot onto a reference plane such as a surface plate.

(2) Turn power OFF and remove the EPROM from the user socket SOC2 located inside the drive unit side door. At the same time, place ST2 and bit 4 of SW1 located in the side door in the upper position.

(3) Turn power ON and perform the return-to-origin operation.

(4) After the robot has returned to the origin, perform the following operation to move the robot to a temporary reference position which will be defined by the system ROM data. (This will cause the robot arm to be almost fully extended in the forward direction.)

Using the teaching box

Turn ON the teaching box and press [ORG] and [ENT] keys in that order.

Using intelligent commands through personal computer (personal computer mode)

Turn OFF the teaching box and execute the command "OG" in direct execution mode.

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

4.2 Setting the Reference Position

(1) Using the teaching box, fully extend the robot arm so that it runs in parallel with the reference plane as shown in Fig. 5.4.1. At this time, the line joining centers of rotation in all joints, i.e., the shoulder, elbow, and wrist (pitch and roll), should be a straight line and run in parallel with the reference plane. Perform this operation as accurately as possible by using jigs: remember, positioning accuracy in the cartesian coordinate system is determined by this setting.

NOTE: Wrist rool cannot be rotated.

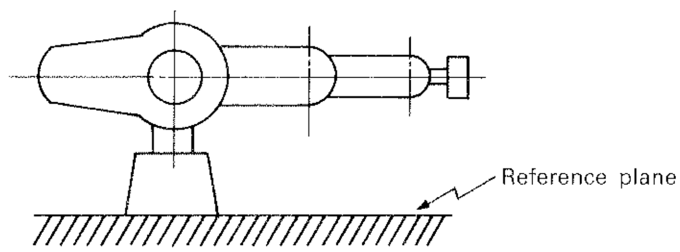


Fig. 5.4.1 Setting the Reference Position in the Cartesian Coordinate System

(2) Perform the following operation to define the current position as the reference position.

Using the teaching box

Turn ON the teaching box and press [P.S], [0], and [ENT] keys in that order.

Using intelligent commands through personal computer

Turn OFF the teaching box and execute the command "HO" in direct execution mode.

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

(3) Place bit 4 of SW1 located inside the drive unit side door in the lower position (OFF). This inhibits the setting of the reference position.

(4) This completes the reference position setting procedure.

4.3 Storing Reference Position Data in EPROM

The reference position data just defined is stored in the drive unit RAM as position number "0." Now, write the data into the EPROM by following the steps given below. These procedures may be performed after other position data have been defined and the program steps have been written.

(1) Ready an erased EPROM and insert it into SOC2 located inside the drive unit side door.

(2) Perform the following operation to write the reference position data into ROM.

Using the teaching box

Turn ON the teaching box and press [WRT] and [ENT] keys in that order.

Using intelligent commands through personal computer

Turn OFF the teaching box and execute the command "WR" in direct execution mode.

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

4.4 Loading the Reference Position Data

The reference position data written in the EPROM can be transferred to the drive unit RAM by performing the following operation. These procedures must be performed before teaching the robot through a series of points.

(1) Insert the EPROM, in which the reference position data has been written, into SOC2 located inside the drive unit side door.

(2) Place bit 4 of SW1 located inside the drive unit side door in the upper position (ON). Then, transfer the data in EPROM to the RAM by performing the following procedure. Note that, at this time, the program steps and other position data are also transferred.

Transferring data upon power-up

Place ST2 located inside the drive unit side door in the upper position (ON) and then turn power ON.

Using the teaching box

Turn ON the teaching box and press [TRN] and [ENT] keys in that order.

Using intelligent command through personal computer

Turn OFF the teaching box and execute the command "TR" in direct execution mode.

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

(3) Place bit 4 of SW1 located inside the drive unit side door in the lower position (OFF). This inhibits the setting of the reference position.

WARNING

All position data obtained through teaching are calculated based on the angle with reference to the reference position in the cartesian coordinate system. If the reference position is changed, therefore, the robot moves through a different series of points than those taught.

5. PROGRAMMING SYSTEM USING PERSONAL COMPUTER

It is wise to create a programming system when a personal computer is used to operate the robot and generate programs. This chapter 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.

5.1 Writing and Executing the Program

```
1000 ' ******** RV-M1 PROGRAMMING SYSTEM ********
1010 ' ******** USING MULTI 16 BY MITSUBISHI ELECTRIC ********
1020 '
1030 OPEN "COM1 :9600, E, 7, 2" AS #1 ·········· Varies with different personal computer models.
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$="ER" THEN PRINT:GOTO 1910
1120 PRINT #1,A$: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 ' ******** RS232C 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 '
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)":PRINT
1390 FOR I=S TO E
```

When using NEC's PC9801, read the following lines of the program as indicated below:

```
1030 OPEN "COM1:E73" AS #1
  ⁝
1060 1F C$="DR" THEN 1170
1070 1F C$="LR" THEN 1240
1080 1F C$="PR" THEN 1360
1090 1F C$="WH" THEN 1600
1100 1F C$="CR" THEN 1780
1110 1F C$="ER" THEN 1910
1120 PRINT #1, A$
```

(Note)
If the LPRINT statement is used instead of the PRINT statement, the contents of the program can be output to the printer.

```
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
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)":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
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$="0" THEN 1860
1850    PRINT "SC";:PRINT USING"##" ;I;:PRINT USING "######" ;VAL (A$)
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
```

〈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 the RS232C read instructions containing parameter, enter the first two letters only and define the parameter using the BASIC input statements (commands LR, PR, and CR).

Program generation:

Directly key in the intelligent commands following line numbers (1 to 2048). 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 with ※ 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. RN5 ↲ ..... Executes the program starting with line number 5.

Error mode II is caused 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 control panel.

<Applications>

```
RUN ↲                              ········· Direct execution example
NT ↲
MO 1 ↲
GO ↲
MO 2 ↲
DW 10, 20, 30 ↲
  ⁀
10 MO 1,C ↲                        ········· Program generation example
20 MO 2,O ↲
30 GC ↲
40 DW 10, −20, 30 ↲
50 ED ↲
  ⁀
LR ↲                               ········· Program read example ※1
START LINE=?10 ↲                             (Reading the program on specified
END LINE=?50 ↲                               line 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)
CUR. POS.	10.0	380.0	50.0	−70.0	−40.0

```
PR ↲                               ········· Position read example ※2
START POSITION = 1 ↲                         (Reading the position on specified
END POSITION = 2 ↲                           line numbers through RS232C)
```

POS. No.	X (mm)	Y (mm)	Z (mm)	P (deg)	R (deg)
PD 1,	0.0,	380.0,	300.0,	−70.0,	−40.0
PD 2,	−10.0,	350.0,	280.0,	−70.0,	−30.0

※ 1, 2: To correct any program and position read by using the cursor and required keys, the return key (↲) must be hit at the end (on the right) of the line and position corrected.

※ 3 : Hexadecimal data in the program read is displayed as decimal data (e.g. OD&FF → OD255).

5.2 Saving and Loading Program and Position Data

The following BASIC program allows the required program and position data to be stored from the drive unit to the floppy disk in the personal computer via RS-232C, and to be transferred from the floppy disk to the drive unit. Note that some modifications of the program are necessary depending on the model of your personal computer.

```
1000 ' ******** RV-M1 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 '
```

When using NEC's PC9801, read the following lines of the program as indicated below:

```
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:"

```
1390 ' ******** PROGRAM SAVE ********
1400 OPEN "COM1:9600, E, 7, 2" AS #1
1410 INPUT "FILE NAME (*. POS)=";F$
1420 OPEN "A:"+F$ FOR OUTPUT 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 OPEN "A:"+F$ FOR INPUT AS #2
1580 LINE INPUT #2, A$
1590 PRINT #1, A$:PRINT A$
1600 IF EOF (2)=-1 THEN 1610 ELSE 1580
1610 CLOSE
1370 GOTO 1030
```

〈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 press the return key. Operation examples are indicated on the next page.

Example 1: Program save

FILE NAME (*.PRG) = ? TEST. PRG ..... Enter any file name.
START LINE=? 10 ↲ ..... Enter the head line number to be saved.
END LINE=? 100 ↲ ..... Enter the final line number 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 ↲ ..... Enter 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 ↲ ..... Enter any file name.
START POSITION=? 100 ↲ ..... Enter the head position number to be saved.
END POSITION=? 200 ↲ ..... Enter the final position number 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 ↲ ..... Enter 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.

6. SAMPLE PROGRAMS

This chapter gives several sample programs in which intelligent commands are used. Note that the starting line number of the sample programs is that of the Movemaster program, and not that of the BASIC.

Example 1: Pick-and-place work

This program causes the robot to shift the workpiece from one place to another. The robot is taught through only positions 1 and 2 and the aerial distances of travel from the respective positions are to be predefined by the command "PD."

⟨Positions used⟩

Position 1: At which the workpiece is grasped.	} Teaching	
Position 2: Onto which the workpiece is placed.		
Position 10: Aerial distance of travel from position 1.	} Numeric values are predefined.	
Position 20: Aerial distance of travel from position 2.		

⟨Sample program⟩

PD 10, 0, 0, 20, 0, 0 (X, Y, Z, P, R)	; Defines aerial distance of travel from position 1 (Z = 20mm) in direct mode and identifies the aerial position as position 10.
PD 20, 0, 0, 30, 0, 0 (X, Y, Z, P, R)	; Defines aerial distance of travel from position 2 (Z = 30mm) in direct mode and identifies the aerial position as position 20.
30 SP 7	; Sets initial speed.
40 MA 1, 10, O	; Moves robot to a location above workpiece (20mm above position 1) with hand opened.
50 MO 1, O	; Moves robot to workpiece (to position 1).
60 GC	; Closes hand to grasp workpiece.
70 MA 1, 10, C	; Moves robot above position 1 (distance being 20mm) with workpiece grasped.
80 MA 2, 20, C	; Moves robot to a location 30mm above position 2.
90 MO 2, C	; Moves robot to position 2.
100 GO	; Opens hand to release workpiece.
110 MA 2, 20, O	; Moves robot above position 2 (distance being 30mm) with hand opened.
120 GT 40	; Returns robot to position 10 allowing it to repeat the sequence (jump to line number 40).

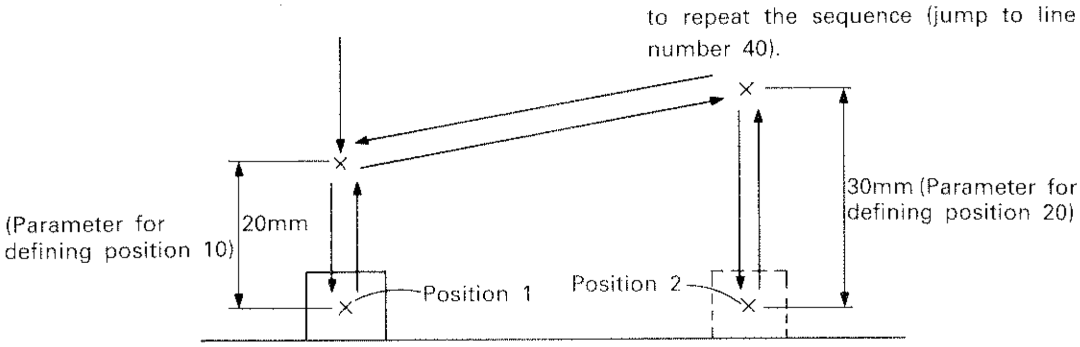


Fig. 5.6.1 Pick-and-Place Work

Example 2: Interrupt

This 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 drive unit input terminal.

〈Position used〉

Position 1: A position above the workpiece (teaching)

〈Input signal used〉

Bit 1: Workpiece detecting signal.

〈Sample program〉

```
 90 SP 5          ; Selects speed 5.
100 EA+1, 140     ; Enables the interrupt by bit 1.
110 MO 1, O       ; Moves robot to a position above the workpiece.
120 DW 0, 0, -50  ; Moves robot 50mm in -Z direction.
130 GT 110        ; Jumps to line number 110 causing robot to return
                    to position 1 as no workpiece has been detected.
140 DA 1          ; Disables interrupt by bit 1.
150 GC            ; Closes hand to grasp workpiece.
160 MO 1, C       ; Moves robot to position 1 with workpiece
                    grasped.
```

* In the above sample program, line number 120 causes the robot to move 50mm in −Z tool direction and, if there is a workpiece, the limit switch signal is input and the robot is stopped. Then, the program jumps to line number 140 to disable interrupt and to allow the robot to grasp 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.

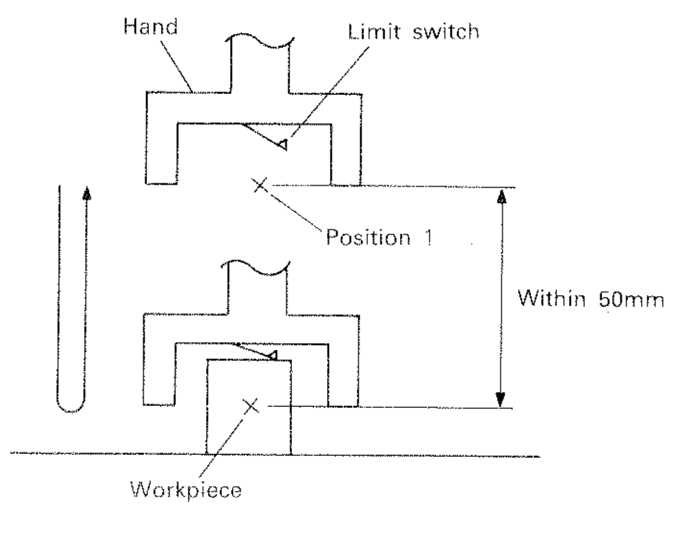


Fig. 5.6.2 Interrupt

Example 3: Palletizing

This 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 workpieces that have undergone the test. The program assumes that the shapes of the two pallets are different.

〈Position used〉

Position 1: Pallet 1 set position } Defined by command PT
Position 2: Pallet 2 set position }
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 } Teaching
Position 21: Pallet 2 column terminating position }
Position 22: Pallet 2 row terminating position }
Position 23: Pallet 2 corner position opposite to reference }
Position 30: Test equipment set position }
Position 50: Aerial distance of travel from pallets······ Numeric values are predefined.

〈Counter 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

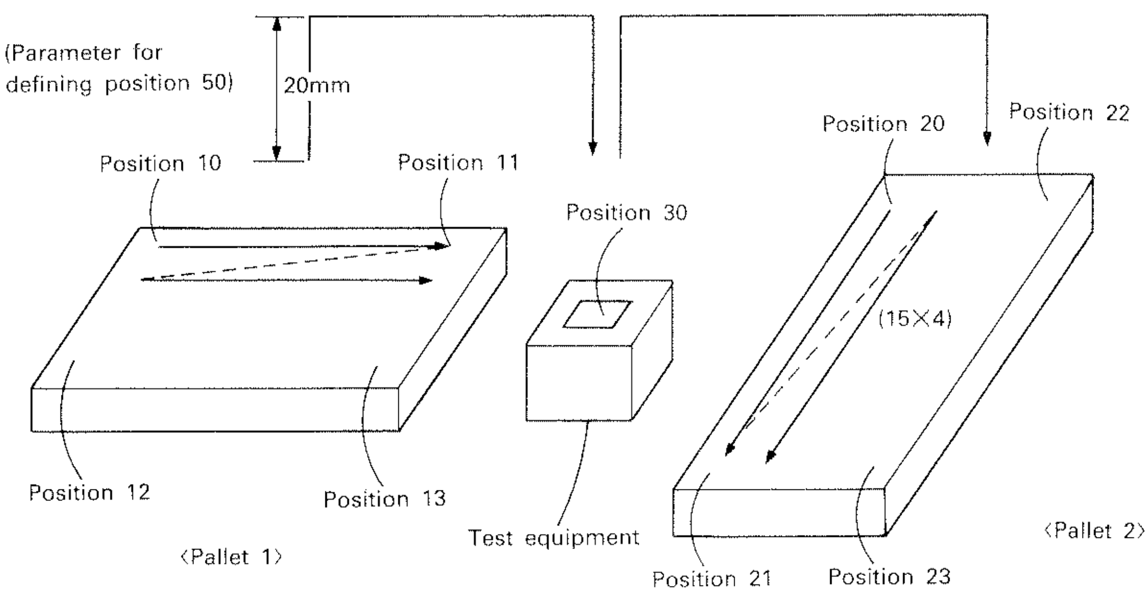


Fig. 5.6.3 Palletizing

〈Sample program〉

```
PD 50, 0, 0, 20, 0, 0       ; Defines aerial distance of travel (Z =
      (X, Y, Z, P, R)         20mm) in direct mode and identifies the
                              aerial position as position 50.

 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              ; Defiens the number of grid points in the column
                              and row directions for pallet 1 (vert. 10 X horiz. 6).
30 PA 2, 15, 4              ; Defiens the number of grid points in the column
                              and row directions for pallet 2 (vert. 15 X horiz. 4).
35 SC 11,1                  ; Loads initial value in pallet 1 column counter.
40 SC 12,1                  ; Loads initial value in pallet 1 row counter.
45 SC 21,1                  ; Loads initial value in pallet 2 column counter.
50 SC 22,1                  ; Loads initial value in pallet 2 row counter.
```

(Main program)

```
100 RC 60                   ; Sets the number of repeat cycles of a loop up to
                              line number 140.
110 GS 200                  ; Causes robot to pick up a workpiece from pallet 1.
120 GS 300                  ; Causes robot to set up the workpiece on test
                              equipment.
130 GS 400                  ; Causes robot to place workpiece in pallet 2.
140 NX                      ; Returns to line number 100.
150 ED                      ; End.
```

* The followings are subroutines used in the main program.

(Subroutine: Picking up workpiece for test)

```
200 SP 7                    ; Sets speed.
202 PT 1                    ; Identifies the coordinates of calculated grid point
                              on pallet 1 as position 1.
204 MA 1,50,O               ; Moves robot to a location above position 1
                              (distance being 20mm in Z direction).
206 SP 2                    ; Sets speed.
208 MO 1,O                  ; Moves robot to position 1.
210 GC                      ; Closes hand to grasp workpiece.
212 MA 1,50,C               ; Moves robot to a location above position 1 with
                              workpiece grasped (distance being 20mm in Z
                              direction).
214 IC 11                   ; Increments pallet 1 column counter by 1.
216 CP 11                   ; Loads value in counter 11 into internal compari-
                              son register.
218 EQ 11,230               ; Jumps to line number 230 on completing column
                              line.
220 RT                      ; Ends subroutine otherwise.
230 SC 11,1                 ; Initializes counter 11.
```

232 IC 12	; Increments pallet 1 row counter by 1.
234 RT	; Ends subroutine.

(Subroutine: Setting up workpiece on test equipment)

300 SP 7	; Sets speed.
302 MT 30,−50,C	; Moves robot to a location 50mm ahead of test equipment.
304 SP 2	; Sets speed.
306 MO 30, C	; Causes robot to set up workpiece on test equipment.
308 ID	; Fetches input data.
310 TB −7,308	; Causes robot to wait for test to complete.
312 MT 30,−50,C	; Moves robot to a location 50mm ahead test equipment.
314 RT	; Ends subroutine.

(Subroutine: Placing tested workpiece in pallet 2)

400 SP 7	; Sets speed.
402 PT 2	; Identifies the coordinates of calculated grid point on pallet 2 as position 2.
404 MA 2,50,C	; Moves robot to a location above position 2 (distance being 20mm in Z direction).
406 SP 2	; Sets speed.
408 MO 2,C	; Moves robot to position 2.
410 GO	; Opens hand to release workpiece.
412 MA 2,50,O	; Moves robot to a location above position 2 (distance being 20mm in Z direction).
414 IC 21	; Increments pallet 2 column counter by 1.
416 CP 21	; Loads value in counter 21 into internal comparison register.
418 EQ 16,430	; Jumps to line number 430 on completing row line.
420 RT	; Ends subroutine otherwise.
430 SC 21,1	; Initializes counter 21.
432 IC 22	; Increments pallet 2 row counter by 1.
434 RT	; Ends 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 following column line.
  (See line numbers 214 to 232, 414 to 432.)

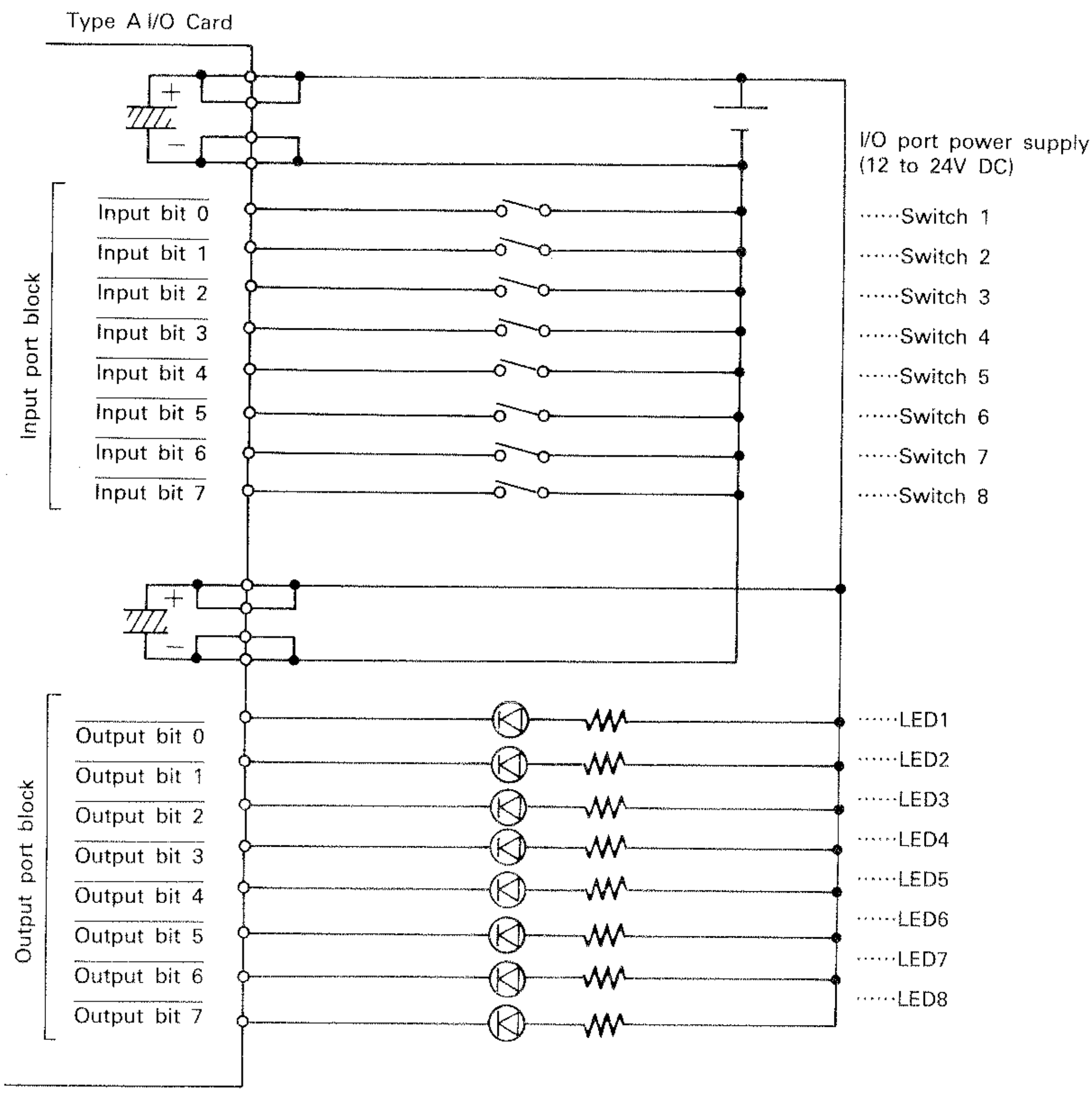
* Robot waits until a test complete signal is input. (See line number 310.)

* 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

This program causes the robot to select any of 8 works through 8 switches connected to the inputs for use as external I/O equipment and display the work currently being executed by any of the 8 LEDs connected to the outputs.

〈Connection〉



※ See Fig. 5.3.2.B for Type B I/O card.

Fig. 5.6.4 Connection with External I/O Equipment

〈Flow chart〉

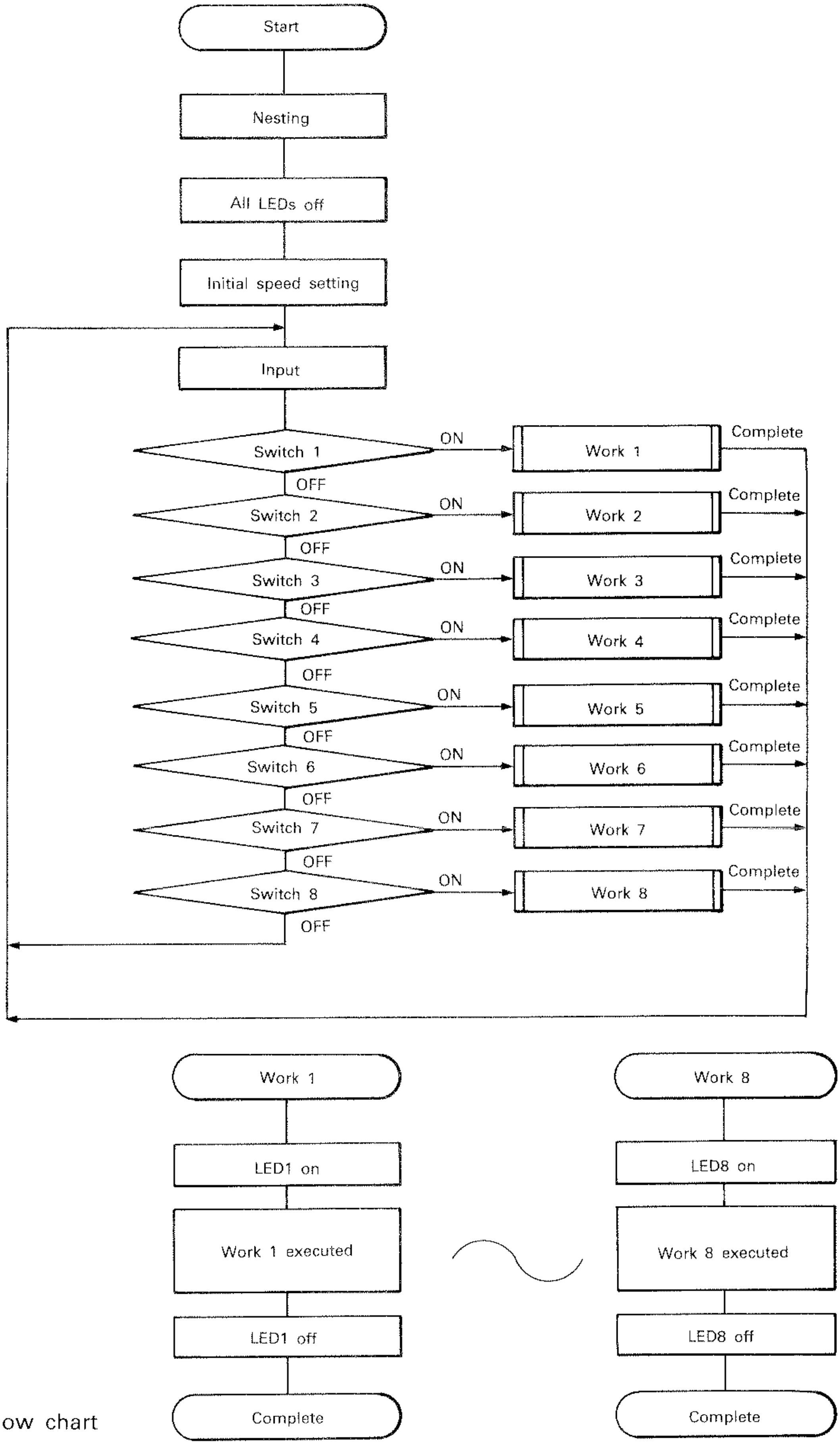


Fig. 5.6.5 Flow chart

〈Sample program〉

(Main routine)

```
10 NT              ; Nesting
15 OD 0            ; Switches off all LEDs.
20 SP 5            ; Sets speed.
25 ID              ; Input
30 TB +0, 100      ; Jumps to line number 100 when switch 1 is turned
                     on. (Work 1)
31 TB +1, 200      ; Jumps to line number 200 when switch 2 is turned
                     on. (Work 2)
32 TB +2, 300      ; Jumps to line number 300 when switch 3 is turned
                     on. (Work 3)
33 TB +3, 400      ; Jumps to line number 400 when switch 4 is turned
                     on. (Work 4)
34 TB +4, 500      ; Jumps to line number 500 when switch 5 is turned
                     on. (Work 5)
35 TB +5, 600      ; Jumps to line number 600 when switch 6 is turned
                     on. (Work 6)
36 TB +6, 700      ; Jumps to line number 700 when switch 7 is turned
                     on. (Work 7)
37 TB +7, 800      ; Jumps to line number 800 when switch 8 is turned
                     on. (Work 8)
38 GT 25           ; Returns to line number 25 (when all switches are
                     off.)
```

(Subroutines)

```
100 OB +0          ; Switches on LED1 (work started).        ┐
105 MO 10          ; Executes work 1.                        │
     ⌇                                                       │ Work 1
198 OB -0          ; Switches off LED1 (work complete).      │
199 GT 25          ; Returns to line number 25.              ┘
     ⌇
800 OB +7          ; Switches on LED8 (work started).        ┐
805 MO 80          ; Executes work 8.                        │
     ⌇                                                       │ Work 8
898 OB -7          ; Switches off LED8 (work complete).      │
899 GT 25          ; Returns to line number 25.              ┘
```

7. COMMAND LIST

A Position/Motion Control Instructions

Program yes ······ Possible
no ······ Not possible

	Name	Input Format	Function	Program	Remarks
1	Decrement Position	DP	Moves robot to a predefined position with a position number smaller than the current one.	yes	
2	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	
3	Here	HE a	Defines the coordinates of the current position by assigning position number (a) to it.	yes	$1 \leqq a \leqq 629$
4	Home	HO	Establishes the reference position in the cartesian coordinate system.	yes	
5	Increment Position	IP	Moves robot to a predefined position with a position number greater than the current one.	yes	
6	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	$1 \leqq a_1, a_2 \leqq 629$ O: Hand opened; C: Hand closed
7	Move Continuous	MC a_1, a_2	Moves robot continuously through predefined intermediate points between position numbers (a_1) and (a_2).	yes	$1 \leqq a_1, a_2 \leqq 629$
8	Move Joint	MJ w, s, e, p, r	Turns each joint the specified angle from the current position.	no	
9	Move	MO a [, O/C]	Moves hand end to position (a).	yes	$1 \leqq a \leqq 629$ O: Hand opened; C: Hand closed
10	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	
11	Move Straight	MS a, n [, O/C]	Moves robot to position (a) through n intermediate points on a straight line.	yes	$1 \leqq a \leqq 629$ $1 \leqq n \leqq 99$ O: Hand opened; C: Hand closed
12	Move Tool	MT a, b [, O/C]	Moves hand end from the current position to a position away from a specified position (a) in incremental distance b in the tool direction.	yes	$1 \leqq a \leqq 629$ O: Hand opened; C: Hand closed
13	Nest	NT	Returns robot to mechanical origin.	yes	
14	Origin	OG	Moves robot to the reference position in the cartesian coordinate system.	yes	
15	Pallet Assign	PA i, j, k	Defines the number of grid points (j, k) in the column and row directions for pallet (i).	yes	$1 \leqq i \leqq 9$ $1 \leqq j, k \leqq 255$
16	Position Clear	PC a_1, [, a_2]	Clears all position data from position a_1 to a_2.	no	$a_1 \leqq a_2$ $1 \leqq a_1, a_2 \leqq 629$ (or $a_1 = 0$)
17	Position Define	PD a, x, y, z, p, r	Defines the coordinates (x, y, z, p, r) of position (a).	no	$1 \leqq a \leqq 629$

	Name	Input Format	Function	Program	Remarks
18	Position Load	PL a_1, a_2	Assigns the coordinates of position (a_2) to position (a_1).	yes	$1 \leqq a_1, a_2 \leqq 629$
19	Pallet	PT a	Calculates the coordinates of a grid point on pallet (a) and identifies the coordinates as position (a).	yes	$1 \leqq a \leqq 9$
20	Position Exchange	PX a_1, a_2	Exchanges the coordinates of position (a_1) for those of position (a_2).	yes	$1 \leqq a_1, a_2 \leqq 629$
21	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	$1 \leqq a_1, a_2 \leqq 629$
22	Speed	SP a [, H/L]	Sets the operating velocity and acceleration/deceleration time for robot. 0: Minimum speed; 9: Maximum speed	yes	$0 \leqq a \leqq 9$ H: High acceleration/deceleration time; L: Low acceleration/deceleration time
23	Timer	TI a	Halts motion for time (a). (Unit: 0.1 second)	yes	$0 \leqq a \leqq 32767$
24	Tool	TL a	Establishes the distance between hand mounting surface and hand end.	yes	$0 \leqq a \leqq +300.0$ Unit: mm

B Program Control Instructions

	Name	Input Format	Function	Program	Remarks
25	Compare Counter	CP a	Loads value in counter (a) into the internal register.	yes	$1 \leqq a \leqq 99$
26	Disable Act	DA a	Disables interrupt by a signal through bit (a) of external input terminal.	yes	$0 \leqq a \leqq 7$ (15)
27	Decrement Counter	DC a	Decrements counter (a) by 1.	yes	$1 \leqq a \leqq 99$
28	Delete Line	DL a_1 [, a_2]	Deletes contents of line numbers from a_1 to a_2.	no	$a_1 \leqq a_2$ $1 \leqq a_1, a_2 \leqq 2048$
29	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	(−15) (+15) $-7 \leqq a_1 \leqq +7$ +: ON; −: OFF $1 \leqq a_2 \leqq 2048$
30	End	ED	Ends the program.	yes	
31	If Equal	EQ a_1 (or &b), a_2	Causes a jump to occur to line number (a_2) if external input data or counter data equals a_1 (or &b).	yes	(−32767) (32767) $0 \leqq a_1 \leqq 255$ (decimal) $0 \leqq b \leqq$ &FF (hex.) (&8001) (&7FFF) $1 \leqq a_2 \leqq 2048$
32	Go Sub	GS a	Permits the instruction sequence to jump to sub-routine which starts with line number (a)	yes	$1 \leqq a \leqq 2048$
33	Go To	GT a	Permits the program sequence to jump to line number (a) unconditionally.	yes	$1 \leqq a \leqq 2048$
34	Increment Counter	IC a	Increments counter (a) by 1.	yes	$1 \leqq a \leqq 99$

	Name	Input Format	Function	Program	Remarks
35	If Larger	LG a_1 (or &b), a_2	Causes a jump to occur to line number (a_2) if external input data or counter data is greater than a_1 (or &b).	yes	(−32767) (32767) $0 \leqq a_1 \leqq 255$ (decimal) $0 \leqq b \leqq$ &FF (hex.) (&8001) (&7FFF) $1 \leqq a_2 \leqq 2048$
36	If Not Equal	NE a_1 (or &b), a_2	Causes a jump to occur to line number (a_2) if external input data or counter data does not equal a_1 (or &b).	yes	(−32767) (32767) $0 \leqq a_1 \leqq 255$ (decimal) $0 \leqq b \leqq$ &FF (hex.) (&8001) (&7FFF) $1 \leqq a_2 \leqq 2048$
37	New	NW	Deletes all program and position data in RAM.	no	
38	Next	NX	Specifies the range of a loop in a program executed by command RC.	yes	
39	Repeat Cycle	RC a	Repeats the loop specified by command NX (a) times.	yes	$1 \leqq a \leqq 32767$
40	Run	RN a_1 [, a_2]	Executes line numbers from (a_1) to (a_2), (a_2) not included.	no	$1 \leqq a_1, a_2 \leqq 2048$
52	Return	RT	Completes subroutine activated by command GS and returns to main program.	yes	
42	Set Counter	SC a_1, [a_2]	Loads (a_2) into counter (a_1).	yes	$1 \leqq a_1 \leqq 99$ $-32767 \leqq a_2 \leqq 32767$
43	If Smaller	SM a_1 (or &b), a_2	Causes a jump to occur to line number (a_2) if external input data or counter data is smaller than a_1 (or &b).	yes	(−32767) (32767) $0 \leqq a_1 \leqq 255$ (decimal) $0 \leqq b \leqq$ &FF (hex.) (&8001) (&7FFF) $1 \leqq a_2 \leqq 2048$

C Hand Control Instructions

	Name	Input Format	Function	Program	Remarks
44	Grip Close	GC	Closes hand grip.	yes	
45	Grip Flag	GF a	Defines the open/close state of hand grip, used in conjunction with command PD.	yes	a = 0 (open), 1 (closed)
46	Grip Open	GO	Opens hand grip.	yes	
47	Grip Pressure	GP a_1, a_2, a_3	Defines gripping force and gripping force retention time.	yes	$0 \leqq a_1, a_2 \leqq 15$ $0 \leqq a_3 \leqq 99$ (Unit: 0.1 second)

D I/O Control Instructions

	Name	Input Format	Function	Program	Remarks
48	Input Direct	ID	Fetches external signal unconditionally from input port.	yes	
49	Input	IN	Fetches external signal synchronously from input port.	yes	

	Name	Input Format	Function	Program	Remarks
50	Output Bit	OB a	Sets the output state of bit (a) of external output terminal.	yes	$-7 \leqq a \leqq +7$ (−15) (+15) +: ON; −: OFF
51	Output Direct	OD a (or &b)	Outputs data a (or &b) unconditionally through output port.	yes	(−32767) (32767) $0 \leqq a \leqq 255$ (decimal) $00 \leqq b \leqq$ &FF (hex.) (&8001) (&7FFF)
52	Output	OT a (or &b)	Outputs data a (or &b) synchronously through output port.	yes	(−32767) (32767) $0 \leqq a \leqq 255$ (decimal) $00 \leqq b \leqq$ &FF (hex.) (&2701) (&7FFF)
53	Test Bit	TB a_1, a_2	Causes a jump to occur to line number a_2 by means of bit (a_1) in external input terminal.	yes	$-7 \leqq a_1 \leqq +7$ (−15) (+15) +: ON; −: OFF $1 \leqq a_2 \leqq 2048$

E RS232C Read Instructions

	Name	Input Format	Function	Program	Remarks
54	Counter Read	CR a	Reads contents of counter (a).	yes	$1 \leqq a \leqq 99$
55	Data Read	DR	Reads data in external input terminal, used in conjunction with commands ID and IN.	yes	
56	Error Read	ER	Reads status of error (no error:0; error mode Ⅰ: 1; error mode Ⅱ: 2).	no	
57	Line Read	LR a	Reads contents of line number (a).	no	$1 \leqq a \leqq 2048$
58	Position Read	PR a	Reads coordinates of position (a).	yes	$1 \leqq a \leqq 629$
59	Where	WH	Reads coordinates of current position.	yes	

F Miscellaneous

	Name	Input Format	Function	Program	Remarks
60	Reset	RS	Resets error mode Ⅱ.	no	
61	Transfer	TR	Transfers contents of EPROM to RAM.	no	
62	Write	WR	Writes contents of RAM into EPROM.	no	
63	Comment		Allows programmer to write a comment following '.	yes	

8. TECHNICAL INFORMATION

8.1 Relation between Speed Parameters and Velocities

Table 5.8.1 and Fig. 5.8.1 indicate the relationship between the speed parameters of the SP (speed) command and the velocity rates (% with respect to the maximum rotation velocity of each axis).

When two or more axes are to be operated, the specified speed parameter is valid for the axis of rotation which has the maximum number of moving pulses. The specified velocity may not be reached if the moving distance is shorter than the given acceleration/deceleration distance.

SP	0	1	2	3	4	5	6	7	8	9
%	1.2	2.4	4.7	9.4	15.3	24.7	38.3	56.0	76.5	100

Table 5.8.1 Relation between Speed Parameters and Velocities

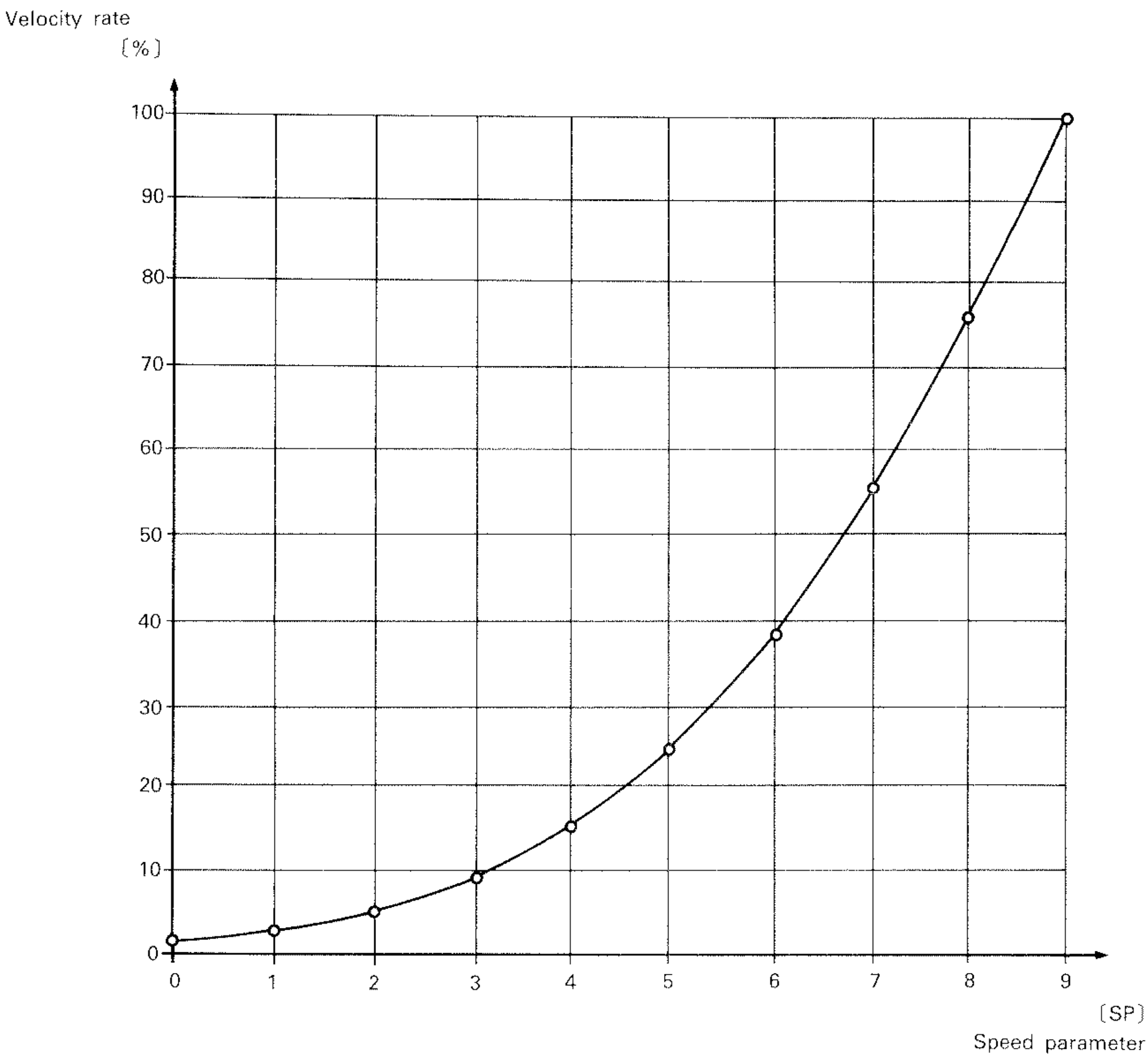


Fig. 5.8.1 Relation between Speed Parameters and Velocities

8.2 Reduction Ratios and Encoder Resolutions

Item	Unit	J_1	J_2	J_3	J_4	J_5
Reduction ratio	——	100	165.3	110	180	110
Encoder resolution	Pulses/rotation	200	200	200	96	96

Table 5.8.2 Reduction Ratios and Encoder Resolutions

8.3 Service Life of the DC Servo Motor

The service life of the 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.

(The bearing grease cannot be changed. Hence, the motor should be replaced when the grease life is expired. The J_4 and J_5 motor brushes cannot be changed. Hence, these motors should be replaced when the life of their brushes is expired.)

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	8000 hours	8000 hours	8000 hours	——	——

Table 5.8.3 Motor Brush and Bearing Grease Lives

9. OPTIONAL FEATURES

9.1 Optional Line Voltage

The optional voltages in Table 5.9.1 are available for the drive unit:

Primary Line Voltage	AC Solenoid Line Voltage	Main Destinations
100V AC	100V, 0.5A	Japan
120V AC	120V, 0.5A	U.S.A.
200V AC	200V, 0.25A	Japan, Hong Kong
220V AC	220V, 0.25A	West Germany, France
230V AC	230V, 0.2A	Singapore
240V AC	240V, 0.2A	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 set" in SPECIFICATIONS Part), only that of the J3 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. OPERATIONAL SPACE DIAGRAM

Use the following diagram for examining layout of peripheral equipment and pallet sizes.

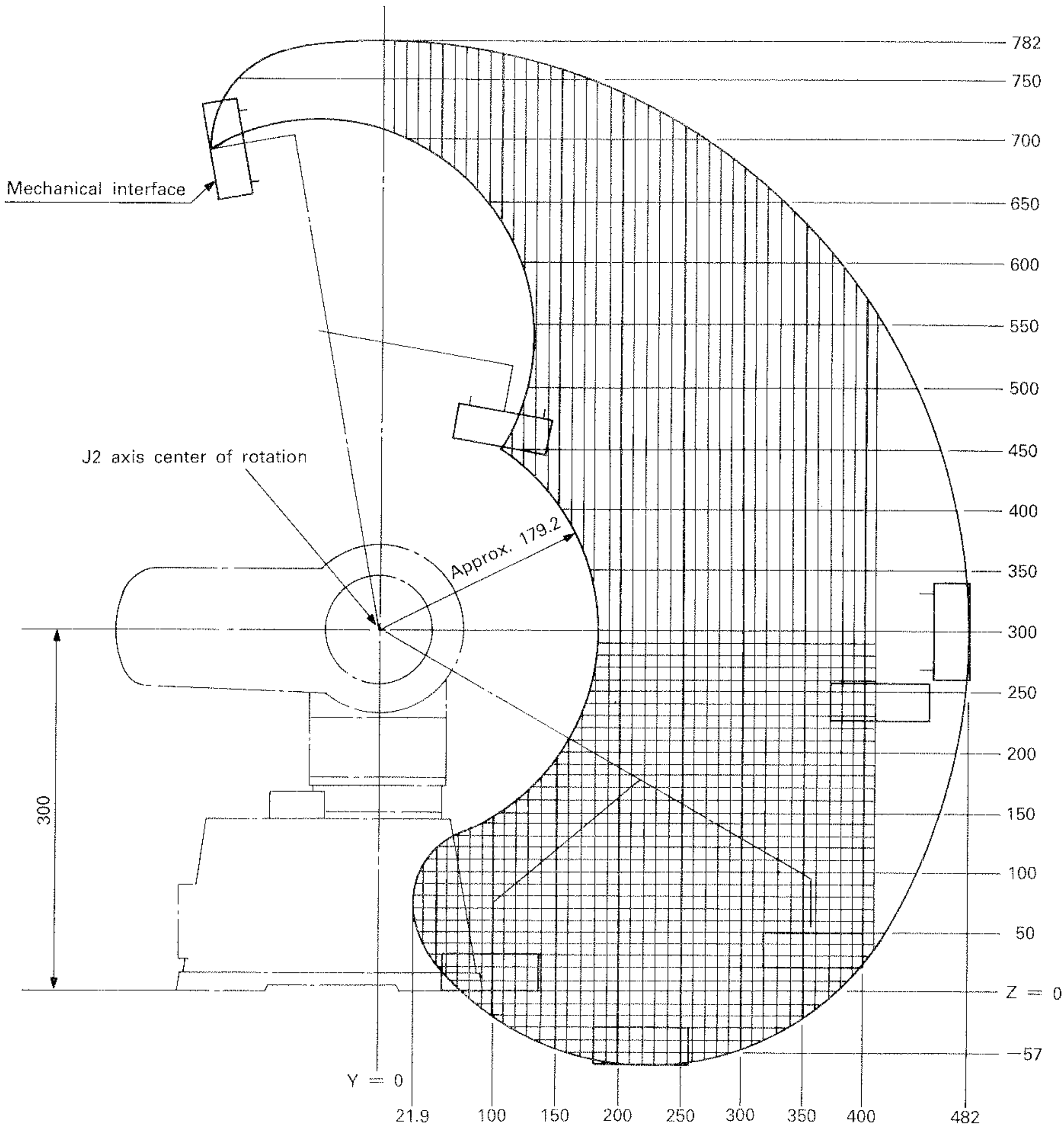
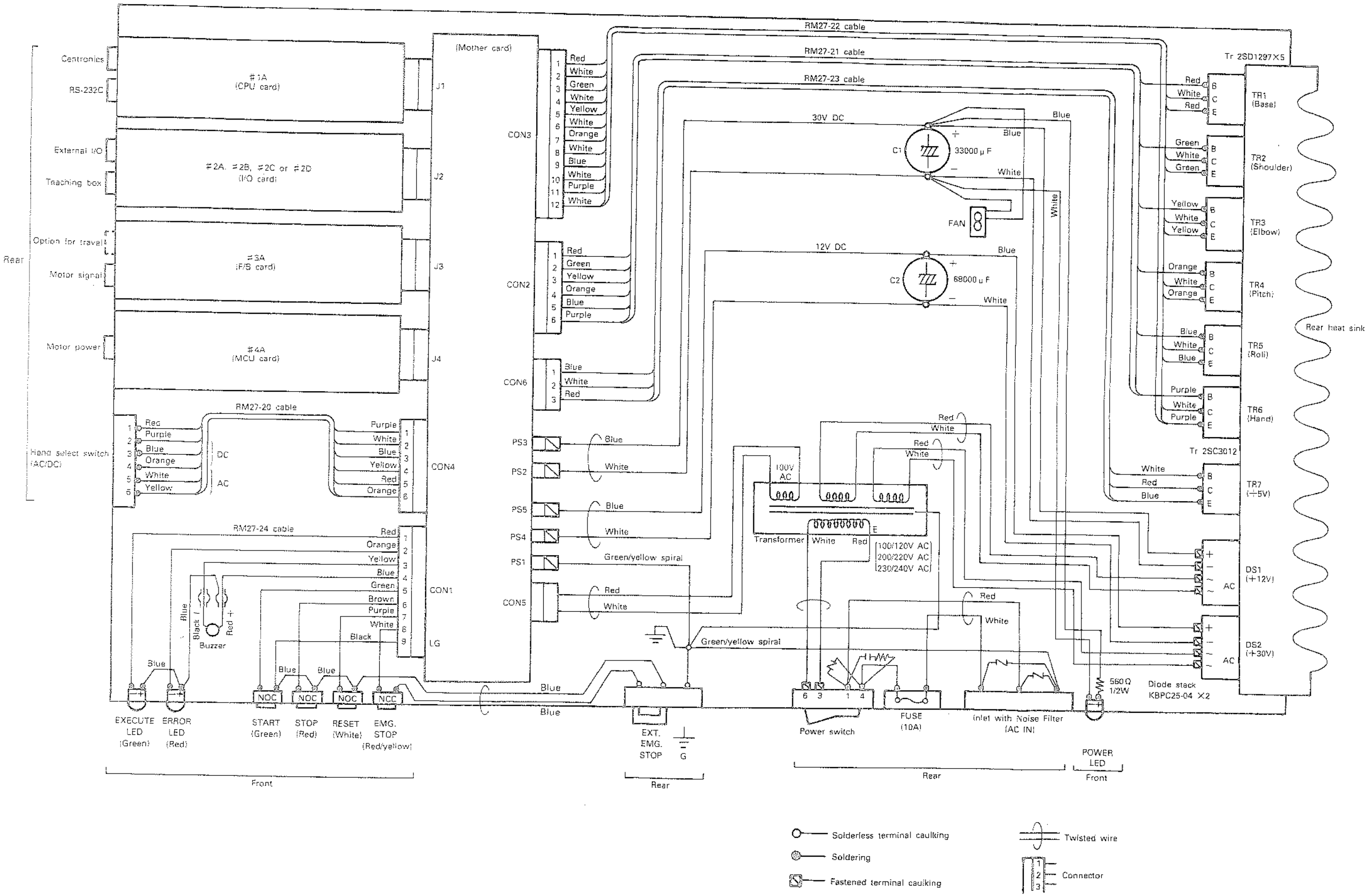
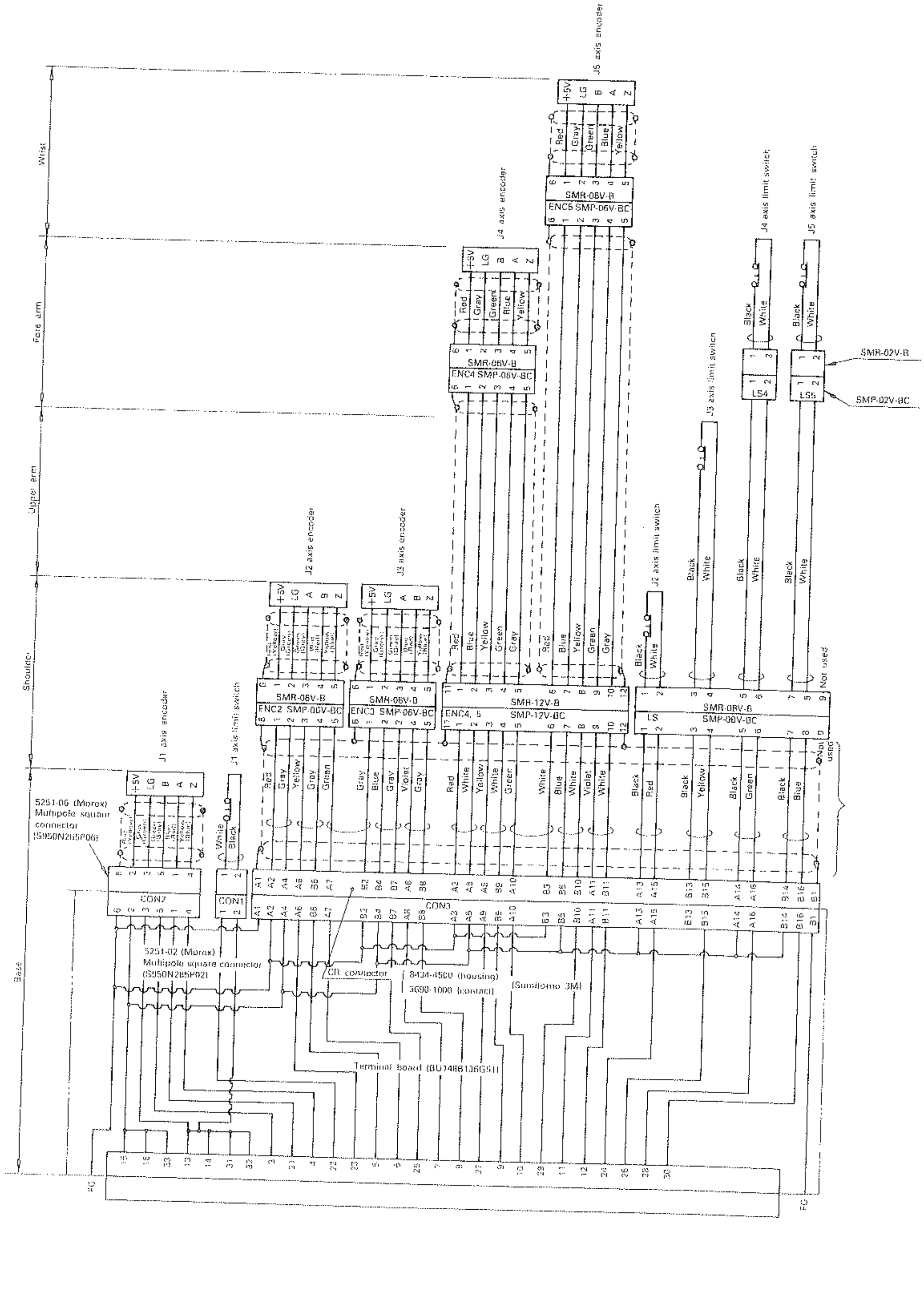


Fig. 5.10.1 Operational Space Diagram

11. DRIVE UNIT WIRING DIAGRAM

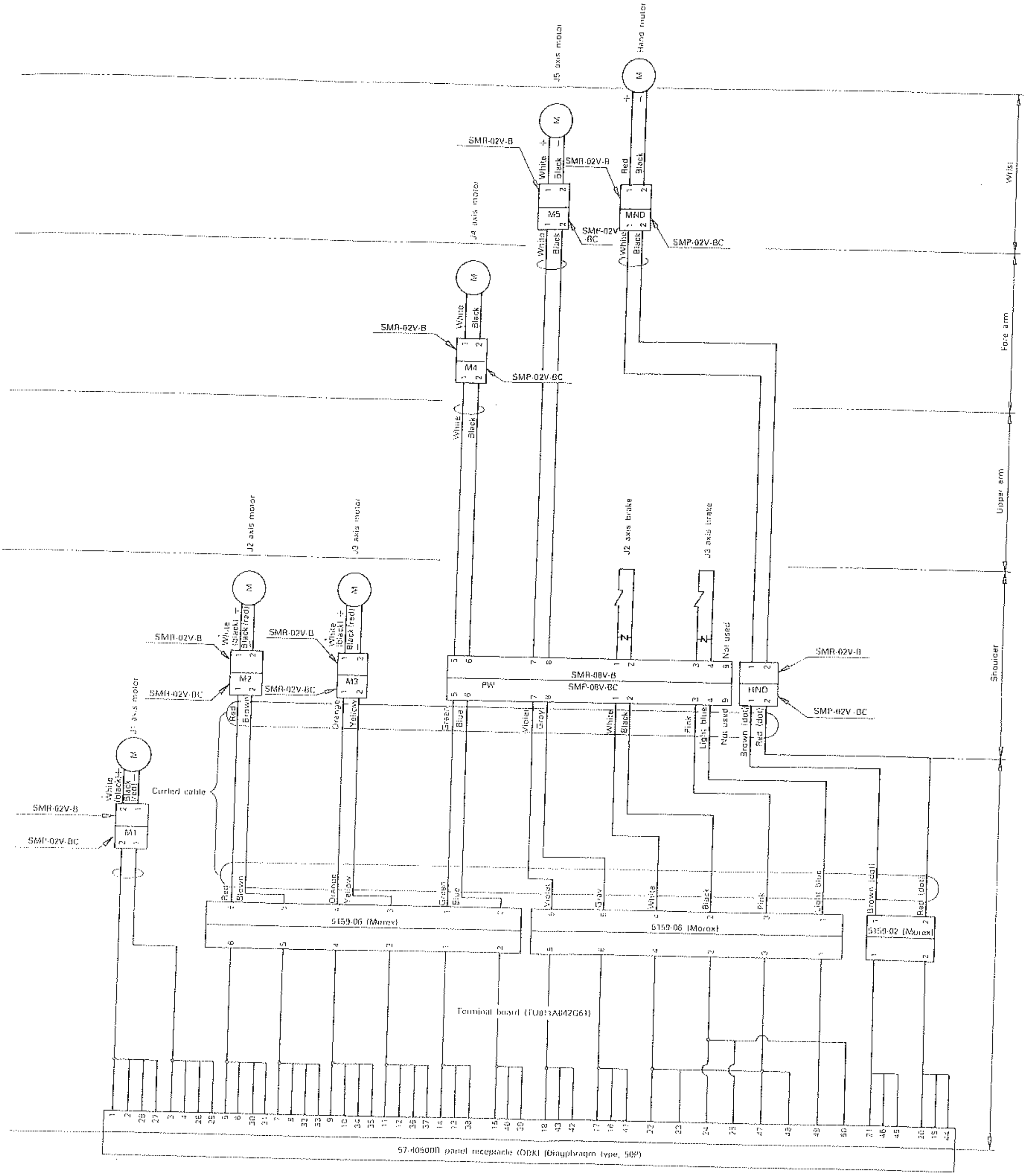


12. ROBOT ARM WIRING DIAGRAM



* The RV-MI's J1 to J3 axis motor builders depend on lots as follows:

* Sanyo Electric Co., Ltd. motor (frame in black) wire color are indicated outside parentheses.
Daito Seisakusho Co., Ltd. motor (frame in gold) wire colors are indicated in parentheses.


Hand motor
J5 axis motor
J4 axis motor
J2 axis motor
J3 axis motor
J1 axis motor
J2 axis brake
J3 axis brake
SMR-02V-B
SMP-02V-BC
SMR-08V-B
SMP-08V-BC
M1
M2
M3
M4
M5
MND
HND
PW
Curled cable
5159-06 (Morex)
5159-02 (Morex)
Terminal board (TU011A842G61)
57-40500D panel receptacle (DDK) (Diaphragm type, 50P)
Wrist
Fore arm
Upper arm
Shoulder
Not used

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

(9106)R0D Printed in Japan

Specifications subject to change without notice.