YAMAHA

YAMAHA MOTOR CO., LTD.

YAMAHA 2-AXIS ROBOT CONTROLLER

DRCX

User's Manual

ENGLISH E

YAMAHA MOTOR CO., LTD.

IM Operations

882 Soude, Naka-ku, Hamamatsu, Shizuoka 435-0054.Japan

URL http://www.yamaha-motor.jp/robot/index.html

E67-Ver. 6.08

General Contents

MEMO

Chapter 1 OVERVIEW

Thank you for purchasing the YAMAHA dual-axis robot controller DRCX series (hereafter called "DRCX controller" or simply "DRCX" or "this controller"). This manual describes DRCX controller features and operating procedures.

When used with a YAMAHA single-axis FLIP-X series or XY robot, the DRCX controller performs positioning and pick-and-place tasks of various mechanical parts and devices.

This first chapter explains basic information you should know before using the DRCX controller such as names and functions of the various parts, steps necessary to prepare the robot for operation, and the architecture of the system itself. Please read this chapter carefully for a basic overview of the DRCX controller.

1-1 Features of the DRCX Series Controller

The DRCX series is a high-performance dual-axis robot controller using a 32-bit RISC chip CPU. The DRCX controller is capable of controlling two YAMAHA single-axis FLIP-X series robots simultaneously. In addition to positioning tasks, the DRCX controller performs I/O control of solenoid valves and sensors, and controls communication with a PC (personal computer).
Using only one DRCX controller allows configuring a complete system for simple applications such as pick-and-place tasks.

The DRCX series has the following features:

- The most suitable controller can be selected according to the robot motor capacity.
- A high-performance 32-bit RISC chip CPU is used for high-speed, high-precision software servo control.
- Absolute method is used as a standard feature. This eliminates return-to-origin operation which has been necessary each time the power is turned on, allowing you to begin actual robot tasks immediately after power-on.
- The program assets that were created with the previous DRC, DRCA, and DRCH series can be used without any modifications. The robot language and I/O control operations are the same as using the ERCX, SRCX and TRCX series. Options such as the TPB programming box, POPCOM support software, memory card and I/O checker can also be directly used as is.
- Ideal acceleration and deceleration speeds can be obtained by simply entering the number of the robot to control and the payload parameter. No troublesome servo adjustments are required.
- The I/O interface provides 16 input and 13 output points for general-purpose user wiring as a standard feature, as well as a 24V power supply for I/O functions. No additional power supply is required.
- The TPB programming box (option) allows interactive user operation by simple menus that permit immediate use. The robot can also be operated from a personal computer (PC) just the same as TPB when the POPCOM software (option) is installed in the PC.
- Programs for robot operation can be written with an easy-to-learn robot language that closely resembles BASIC. Even first-time users will find it easy to use.
- Users not accustomed to robot language can use a PLC (programmable logic controller) to directly move the robot by specifying the operation points.
- Users can create programs and control the robot on a personal computer (PC). Communication with the PC is performed with an easy-to-learn robot language similar to BASIC. Even first-time users will find it easy to use.
- A built-in multi-task function allows efficiently creating the programs.

NOTE

The DRCX controller can be operated from either a TPB (programming box) or a PC running with communication software such as POPCOM. This user's manual mainly describes operations using the TPB. For details on operation with POPCOM, refer to the POPCOM manual. If you want to use your own methods to operate the DRCX controller from a PC, refer to Chapter 11 "Communications with PC" for pertinent information.

1-2 Setting Up for Operation

The chart below illustrates the basic steps to follow from the time of purchase of this controller until it is ready for use. The chapters of this user's manual are organized according to the operation procedures, and allow first time users to proceed one step at a time.

Basic steps

Operation	Information to be familiar with	Refer to
Installation	• Installing the controller	2-1
↓		
Wiring and connection	• Connecting the power supply	2-2
	• Grounding	2-3
	• Connecting peripheral equipment	2-4 to 2-8
	• Understanding the I/O interface	Chapter 3
↓		
Setting parameters	• Understanding basic TPB operations	Chapter 4
	• Setting the various parameters	Chapter 5
↓		
Programming	• Inputting or editing programs	Chapter 6
	• Editing point data	Chapter 7
	• Robot language	Chapter 8
↓		
Running the robot	• Return-to-origin	Chapter 9
	• Various operation steps	
	• Emergency stop	

1-3 External View and Part Names

This section explains part names of the DRCX controller and TPB along with their functions. Note that the external view and specifications are subject to change without prior notice to the user.

1-3-1 DRCX controller

1. Status Display Lamp
   This lamp indicates the operating status of the robot and controller.
   Refer to "15-1-3 LED display" for information on controller status and the matching LED display.
2. Escape Switch (ESC switch)
   Hold down this switch when connecting or disconnecting the TPB from the DRCX controller. (See "4-1 Connecting and Disconnecting the TPB.")
3. TPB Connector
   This is used to connect the TPB or DPB programming box, or the RS-232C terminal of a PC (personal computer).
4. COM Connector
   This is used to connect a network system when the optional network card is installed. (This is covered when the option is not in use.)
5. Robot I/O Connector
   This is used for input and output from robot peripheral devices such as resolver, origin sensor and brake signals.
6. I/O Connector
   This is used to connect external equipment such as a PLC.
7. BAT Connector (1, 2)
   This is the connector for the absolute battery.
8. Motor Connector (X, Y)
   This is the power line connector for the servo motor.
9. Regenerative Unit Connector (RGEN connector)
   Some types of robots require connection to a regenerative unit. In such cases, use this to connect the regenerative unit (RGU-2).
10. Terminal Block

    ACIN1(PWR) (L, N, ⏚)

    This is the connector for supplying AC power to the DRCX controller. The ground terminal must be properly grounded to prevent electrical shock to the human body and to maintain equipment reliability.

    NC

    No connection. Do not use.

    T1, T2

    These are input power voltage switching terminals. When an input power voltage of AC100 to 115V is used, short the T1 and T2 terminals. When an input power voltage of AC200 to 230V is used, leave the T1 and T2 terminals open.

    The models DRCX-1020, DRCX-2010 and DRCX-2020 are specifically designed to operate on AC200 to 230V, so always leave the T1 and T2 terminals open.

Fig. 1-1 Exterior of the DRCX controller

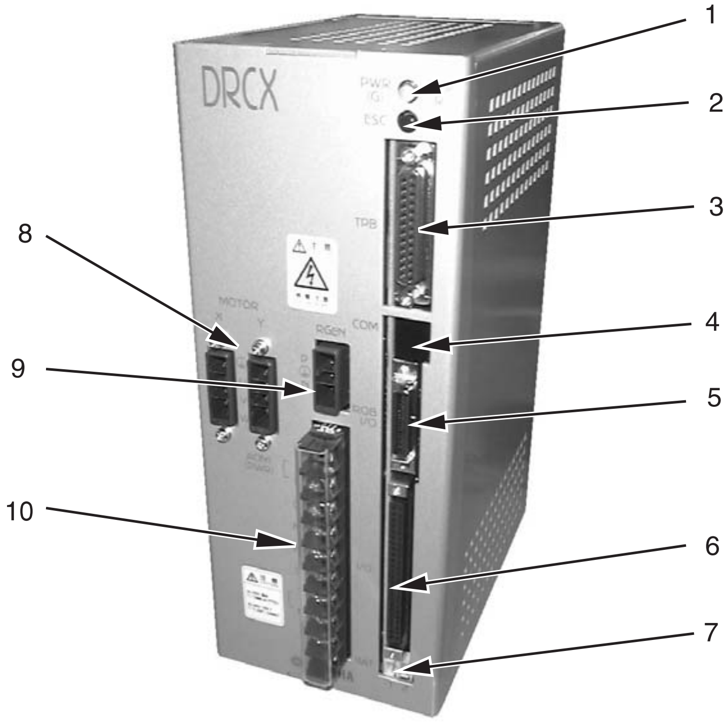
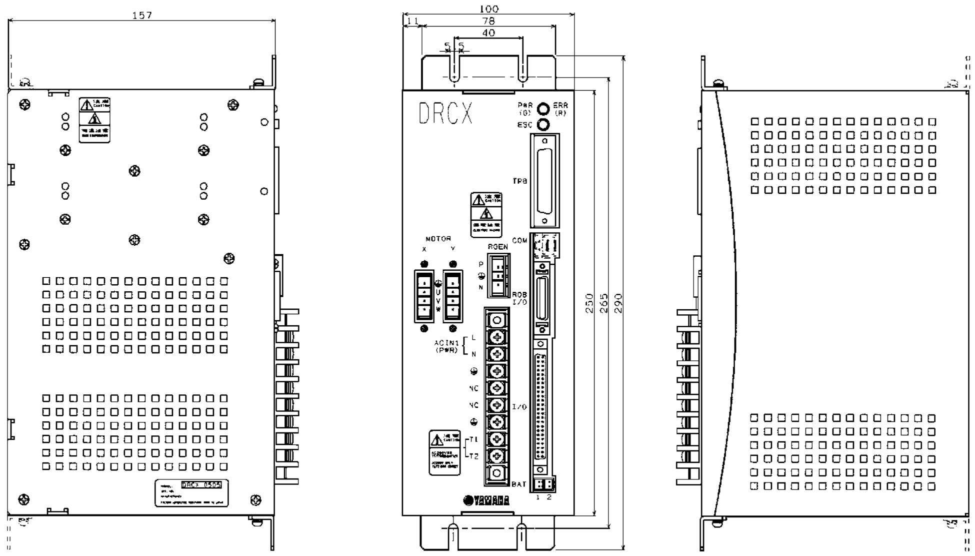


Fig. 1-2 Three-side view of the DRCX controller



1-3-2 TPB

1. Liquid Crystal Display (LCD) Screen
   This display has four lines of twenty characters each and is used as a program console.

2. Memory Card Slot
   An IC memory card can be inserted here. Be careful not to insert the card upside-down.

3. Control Keys
   The TPB can be operated in interactive data entry mode. Instructions are input through the control keys while reading the contents on the LCD screen.

4. Connection Cable
   This cable connects the TPB to the DRCX controller.

5. DC Power Input Terminal
   Not used.

6. Emergency Stop Button
   This is the emergency stop button. When pressed, it locks in the depressed position. To release this button, turn it clockwise.
   To cancel emergency stop, first release this button and then use the servo recovery command via the I/O interface or the servo recovery operation from the TPB.

Fig. 1-3 Exterior of the TPB

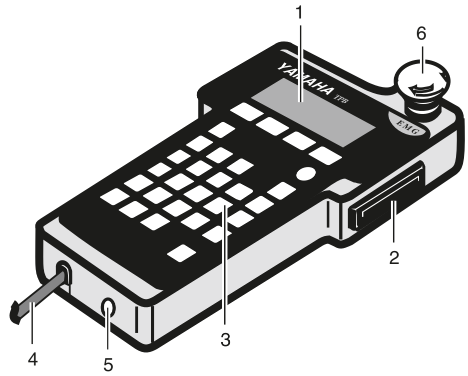
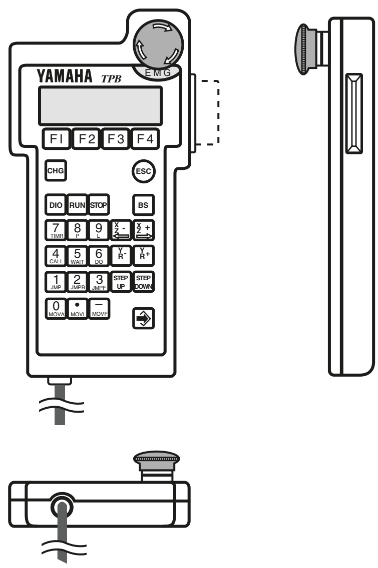


Fig. 1-4 Three-side view of the TPB

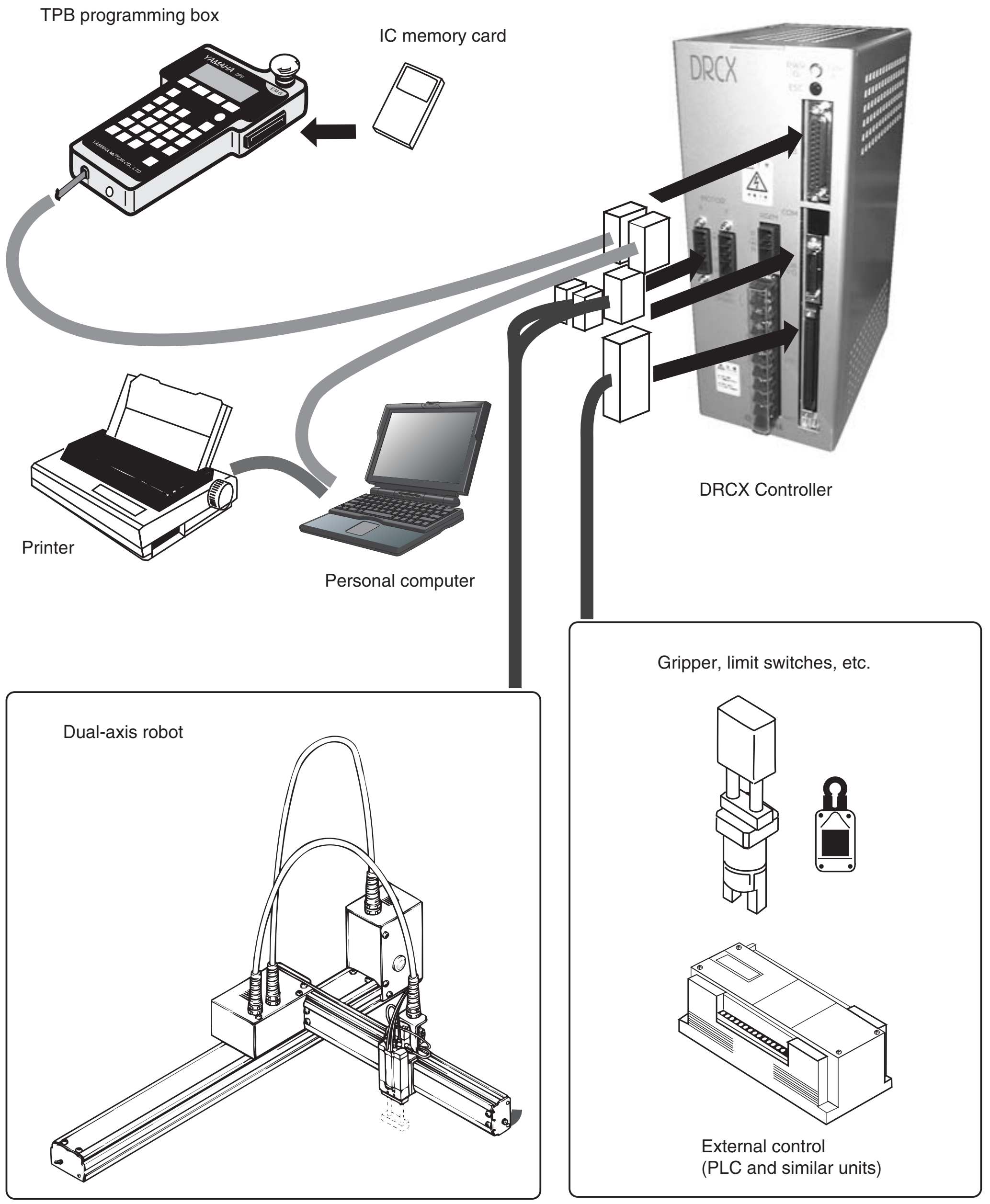


1-4 System Configuration

1-4-1 System configuration

The DRCX series dual-axis controller can be combined with various peripheral units and optional products to configure a robot system as shown below.

Fig.1-5 System configuration diagram



1-5 Accessories and Options

1-5-1 Accessories

The DRCX controller comes with the following accessories. After unpacking, check that all items are included.

1. I/O connector

Connector	: FCN-361P048-AU	made by Fujitsu	1 piece
Connector cover	: FCN-360C048-E	made by Fujitsu	1 piece

2. RS-232C dust cover

XM2T-2501		made by OMRON	1 piece

3. Absolute battery unit (B1, B2)

Ni-Cd battery	: (Either of the following types is supplied according to the user's order.)		
	B1 type (3.6V/700mAh)	made by Sanyo Electric	2 pieces
	B2 type (3.6V/2000mAh)	made by Sanyo Electric	2 pieces
Cable tie	: T30R	made by Tyton	4 pieces
Binding strap	: A TMS-30	made by Kitagawa Industries	4 pieces

1-5-2 Peripheral options

The following options are available for the DRCX controller:

1. TPB
   This is a hand-held programming box that connects to the DRCX controller for teaching point data, editing robot programs and operating the robot. The TPB allows interactive user operation by simple menus so that even first-time users can easily operate the robot with the TPB.

2. IC memory card
   An IC memory card can be used with the TPB to back up programs, point data and parameter data.

3. POPCOM
   The POPCOM is support software that runs on a PC (personal computer) connected to the DRCX controller. The POPCOM software allows easy editing of robot programs and operation of a robot just the same as with a TPB.

4. I/O checker
   The I/O checker connects to the I/O connector and can be used as an I/O status monitor (with LED indicators) or as a simulated input device by toggle switches.

Chapter 2 INSTALLATION AND CONNECTION

This chapter contains precautions that should be observed when installing the controller, as well as procedures and precautions for wiring the controller to the robot and to external equipment.

2-1 Installing the DRCX Controller

2-1-1 Installation method

Using the L-shaped brackets attached to the top and bottom of the controller, install the controller from the front or rear position. (See Fig.1-2 Three-side view of the DRCX controller.)

2-1-2 Installation location

- Install the controller in locations where the ambient temperature is between 0 to 40°C and the humidity is between 35 to 85% without condensation.
- Do not install the controller upside down or at an angle.
- Install the controller in locations with sufficient space (at least 20mm away from the wall or other object) for good ventilation and air flow.
- Do not install the controller in locations where corrosive gases such as sulfuric acid or hydrochloric acid gas are present, or in atmosphere containing flammable gases and liquids.
- Install the controller in locations with a minimal amount of dust.
- Avoid installing the controller in locations subject to cutting chips, oil or water from other machines.
- Avoid installing the controller in locations where electromagnetic noise or electrostatic noise is generated.
- Avoid installing the controller in locations subject to shock or large vibration.

2-2 Connecting the Power Supply

2-2-1 Power supply

Type and Item	Power supply voltage	No. of phases	Frequency	Max. power consumption
DRCX-0505	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	500VA or less
DRCX-0510	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	700VA or less
DRCX-0520	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	1100VA or less
DRCX-1005	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	700VA or less
DRCX-1010	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	900VA or less
DRCX-1020	AC200 to 230V ±10%	Single-phase	50/60Hz	1300VA or less
DRCX-2005	AC100 to 115/200 to 230V ±10%	Single-phase	50/60Hz	1100VA or less
DRCX-2010	AC200 to 230V ±10%	Single-phase	50/60Hz	1300VA or less
DRCX-2020	AC200 to 230V ±10%	Single-phase	50/60Hz	1600VA or less

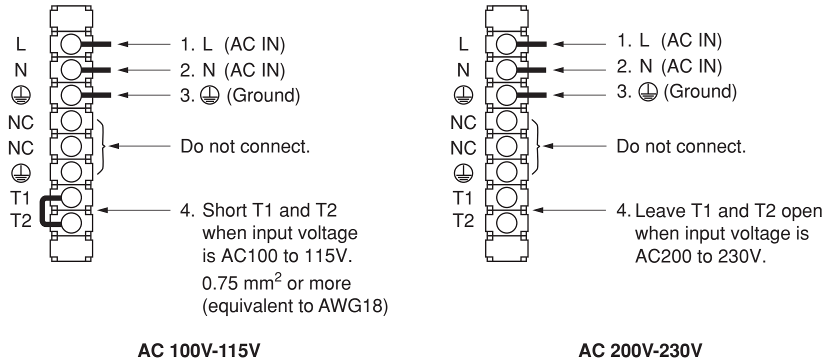
⚠ ***CAUTION***

If the power supply voltage drops below the above range during operation, the alarm circuit will work and return the controller to the initial state the same as just after power-on, or stop operation. To avoid this problem, use a regulated power supply with voltage fluctuations of less than ±10%.
Since the controller uses a capacitor input type power supply circuit, a large inrush current flows when the power is turned on. Do not use fast-blow circuit breakers and fuses. For the same reason, avoid turning the power off and on again repeatedly in intervals of less than 10 seconds. This could harm the main circuit elements in the controller.

2-2-2 Connecting the power supply

Connect the power supply to the power supply terminal block. Refer to the printed marks on the panel, and correctly connect to the connection terminals. Incorrect connections can lead to major hazards such as fire, etc. Treat the end of the wires so that the wires do not disconnect from the terminal block. The methods of connecting T1 and T2 differ according to the input power voltage as shown below. Please note that the model DRCX-1020, DRCX-2010 and DRCX-2020 are specifically designed to operate on 200 to 230V.

Fig. 2-1 Power supply connections



CAUTION

The DRCX series controller does not have a power switch. Be sure to provide a power supply breaker (insulation) of the correct specifications that will turn the power on or off to the entire system including the robot controller.

WARNING

Before beginning the wiring work, make sure that the power supply for the entire system is turned off. Doing the wiring work while power is still turned on may cause electrical shocks.

2-2-3 Installing an external leakage breaker

Since the robot controller drives the motors by PWM control, leakage current flows at high frequencies. This might cause the external leakage breaker to malfunction.
When installing an external leakage current breaker, it is important to choose the optimum sensitivity current rating (IΔn). (Check the leakage breaker manufacturer's data sheets to select the optimum product compatible with inverters.)

	Leakage current
DRCX	4mA (Max.)

CAUTION

1. ***Leak current was measured with a leak tester with a low-pass filter turned on (100Hz).***
   Leak tester: Hioki Electric 3283
2. ***When using two or more controllers, sum the leakage current of each controller.***
3. ***Make sure that the controller is securely grounded.***
4. ***Stray capacitance between the cable and FG may vary depending on the cable installation condition, causing the leakage current to fluctuate.***

2-2-4 Installing a circuit protector

An inrush current, which might be from several to nearly 20 times higher than the rated current, flows at the instant that the DRCX controller is turned on or the robot motors start to operate.
When installing an external circuit protector for the robot controller, select a circuit protector that provides optimum operating characteristics.
To ensure proper operation, we recommend using a medium to slow response circuit protector with an inertial delay function. (Refer to the circuit protector manufacturer's data sheets for making the selection.)

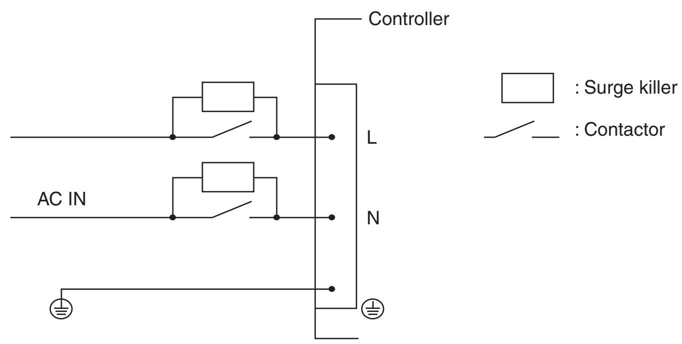
Example

	Rated current	Operating characteristics
DRCX	20A	Slow type with inertia delay (300% 2 sec. 1000% 0.01 sec.)

2-2-5 Installing current control switches

When controlling the power on/off of the robot controller from an external device such as a PLC, a current control switch (contactor, breaker, etc.) may be used. In this case, the current control switch usually creates a large on/off inrush current. To minimize this on/off inrush current, surge killers must be installed for surge absorption. Connect a surge killer in parallel with and close to each contact of the current control switch.

Recommended surge killer:
Okaya Electric XE1201, XE1202, RE1202
Example:



2-2-6 Insulation resistance and voltage breakdown tests

Never attempt insulation resistance tests or voltage breakdown tests on the DRCX controller. Since capacitive grounding is provided between the controller body and 0V, these tests may mistakenly detect excess leakage current or damage the internal circuitry. If these tests are required, please consult your YAMAHA sales office or representative.

2-3 Grounding

The DRCX controller must be grounded to prevent danger to personnel from electrical shocks in case of electrical leakage and prevent equipment malfunctions due to electrical noise.
We strongly recommend that Class D (grounding resistance of 100 ohms or less) or higher grounding be provided. For grounding the controller, use the ground terminal on the power supply terminal block.

* Class D grounding is the same as Class 3 grounding previously used.

2-4 Connecting the DRCX to the Control Unit

The DRCX controller can be operated either through the TPB programming box or through a PC (personal computer) equipped with an RS-232C terminal.
When using the TPB, plug the TPB cable connector into the TPB connector of the DRCX controller. (Refer to "4-1-1 Connecting the TPB to the DRCX controller".)
When using a PC, plug the RS-232C interface cable connector (25 pins) into the TPB connector of the DRCX controller. (Refer to "11-2 Communication Cable Specifications".)
To prevent equipment malfunction due to noise, we strongly recommend that Class D (grounding resistance of 100 ohms or less) or higher grounding be provided.

2-5 Connecting to the Robot

First make sure that the power to the DRCX controller is turned off, and then connect the robot cable to the robot I/O connector and motor connector on the front panel of the DRCX controller.
Be sure to fasten the robot I/O cable with the screws so that it will not come loose or be pulled off.

* When the robot cable is disconnected from the controller, an alarm (15: FEEDBACK ERROR 2) is issued. Since the controller is shipped with the robot cable disconnected, an alarm is always issued when the controller is first turned on. But this is not an equipment problem.

2-5-1 Robot I/O connector and signal table

Mating connector type No. : 10136-6000EL (3M)
Mating connector cover type No. : 10336-52A0-008
DRCX's connector type No. : 10236-52A2JL

Signal Table

Terminal No.	Signal name	Description	Terminal No.	Signal name	Description
1	PS1+	Resolver SIN input 1 (+)	19	PS2+	Resolver SIN input 2 (+)
2	PS1-	Resolver SIN input 1 (-)	20	PS2-	Resolver SIN input 2 (-)
3	PC1+	Resolver COS input 1 (+)	21	PC2+	Resolver COS input 2 (+)
4	PC1-	Resolver COS input 1 (-)	22	PC2-	Resolver COS input 2 (-)
5	R1+	Resolver excitation output 1 (+)	23	R2+	Resolver excitation output 2 (+)
6	R1-	Resolver excitation output 1 (-)	24	R2-	Resolver excitation output 2 (-)
7	DG	Digital ground	25	DG	Digital ground
8	DG	Digital ground	26	DG	Digital ground
9	+24V	Power supply (24V) for origin sensor	27	+24V	Power supply (24V) for origin sensor
10	NC	No connection	28	NC	No connection
11	PG	Power supply (0V) for origin sensor	29	PG	Power supply (0V) for origin sensor
12	ORG1	Origin sensor signal input 1	30	ORG2	Origin sensor signal input 2
13	PG	Power supply (0V) for origin sensor	31	PG	Power supply (0V) for origin sensor
14	BK1+	Brake signal 1 (+)	32	BK2+	Brake signal 2 (+)
15	BK1+	Brake signal 1 (+)	33	BK2+	Brake signal 2 (+)
16	BK1-	Brake signal 1 (-)	34	BK2-	Brake signal 2 (-)
17	BK1-	Brake signal 1 (-)	35	BK2-	Brake signal 2 (-)
18	FG	Frame ground	36	FG	Frame ground

2-5-2 Motor connector and signal table

Mating connector type No. : (X-axis) 1-178128-4, (Y-axis) 2-178128-4 (AMP)
Mating connector contact type No. : 1-175218-5
DRCX's connector type No. : (X-axis) 1-179553-4, (Y-axis) 2-179553-4

Signal table (X and Y axes)

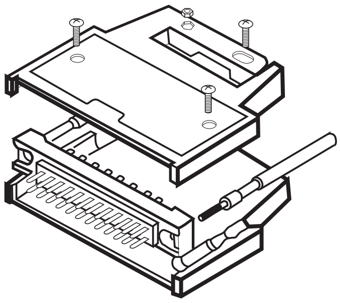
Terminal No.	Signal name	Description	Terminal No.	Signal name	Description
1	FG	Frame ground	3	MV	Motor V-phase output
2	MU	Motor U-phase output	4	MW	Motor W-phase output

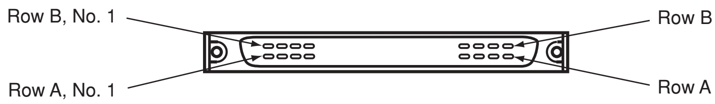
2-6 Connecting to the I/O Connector

The I/O connector is used for connecting the DRCX controller to external equipment such as a PLC. When using external equipment for I/O control, connect the writing to the I/O connector supplied as an accessory and then plug it into the I/O connector on the DRCX controller.
The signals assigned to each of the I/O connector terminals and their functions are described in detail in Chapter 3.

The I/O connector that is compatible with the DRCX controller is listed below.

Connector type No. : FCN-361P048-AU (Fujitsu)
Connector cover type No. : FCN-360C48-E





CAUTION

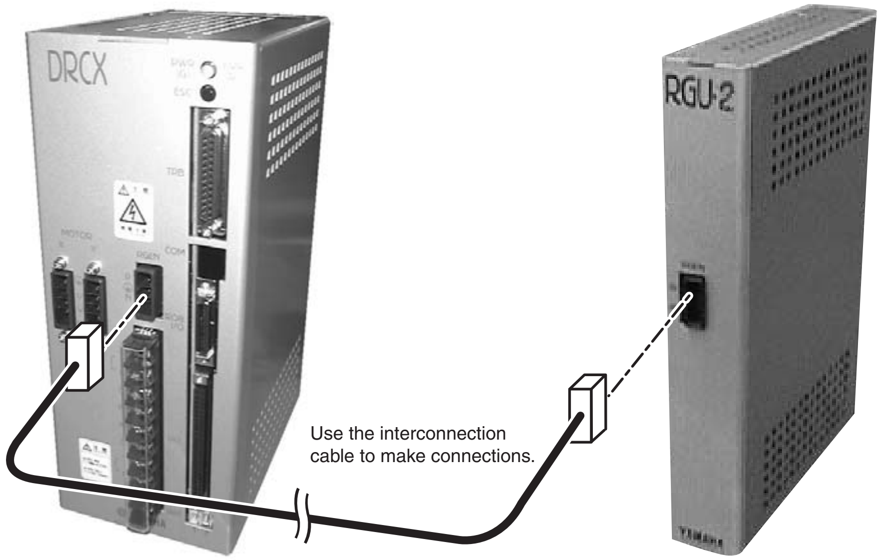
Even if not using I/O control, the I/O connector should be plugged in after completing the following wiring.
1. ***Short pin numbers A24 (EMG1) and B24 (EMG2).***
2. ***Short pin numbers B4 (LOCK) and A15 or B15(0V).***
3. ***Short pin numbers A13 or B13 (+IN COM) and A14 or B14 (+24V)***
   (An external 24V power supply can also be connected to A13 or B13 (+IN COM) instead of using the internal power supply.)

If step 1 is not completed, an emergency stop is triggered. If step 2 is not completed, an interlock occurs. In either case, the controller cannot be operated (see Chapter 3).
Note that 24V power will not be supplied to the I/O circuit unless shorted as in 3. An alarm is issued (06:24V POWER OFF) when power is not supplied and the operation disabled.

2-7 Connecting to the Regenerative Unit

Some types of robots must be connected to a regenerative unit. In such cases, use the interconnection cable to connect the DRCX controller to the regenerative unit.

Fig. 2-2 Connection of the DRCX controller to a regenerative unit



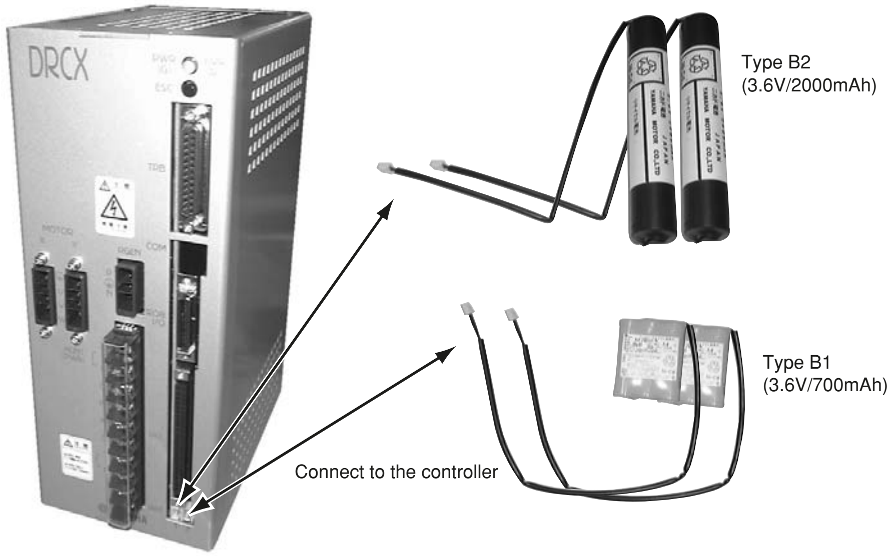
2

INSTALLATION AND CONNECTION

2-8 Connecting the Absolute Battery

Two absolute batteries are supplied with the controller. These two batteries are identical and should be connected to the controller. Data backup is possible with only one battery, but the data backup time will be one-half. After making connections, use cable ties and binding straps (supplied) to secure the batteries on the side of the controller or in the system equipment so that they will not budge. Either type B1 or B2 battery is supplied with the controller according to the customer order request.

Fig. 2-3 Absolute batteries connection to the controller



CAUTION
Do not modify the battery wire or extend it. Modification and extended wire may cause troubles or malfunction of the robot.

* When the absolute battery is disconnected from the controller, an alarm (24: ABS. DATA ERROR) is issued. Since the controller is shipped with the absolute battery disconnected, an alarm is always issued when the controller is first turned on. But this is not an equipment problem. (An alarm "23: ABS. BAT. L-VOLTAGE" might occur in some cases.)
* When the controller is first used or is kept turned off for a period in excess of the data backup time, the battery must be recharged. The battery is automatically charged while the controller is turned on. Keep the battery charged for longer than the time listed in the table below. Since the battery charging time does not affect robot operation, the controller can be used to perform teaching, program editing and robot operation while the battery is still being charged.

	Hours until full charge *1)	Backup time *2)
Type B1 (3.6V/700mAh)	15h	120h
Type B2 (3.6V/2000mAh)	48h	340h

*1) At ambient temperature of 20℃
*2) After power is off with absolute battery fully charged.

* If the absolute backup function is unnecessary, the controller can be used with the absolute battery left removed. (See "16-1 Operation When Not Using Absolute Function".)

Chapter 3 I/O INTERFACE

The DRCX series has an I/O interface consisting of emergency stop inputs, interlock, 7 dedicated command inputs, 3 dedicated outputs, 16 general-purpose inputs, 13 general-purpose outputs, a 24V power supply for I/O control, etc. This I/O interface allows exchanging commands and data between the DRCX series and external equipment. This I/O interface can also directly connect to and control actuators such as valves and sensors. To construct a system utilizing the features of the DRCX series, you must understand the signals assigned to each terminal on the I/O connector and how they work. This chapter covers this fundamental information.

This chapter also provides examples of I/O circuit connections and timing charts for expanding the system by using a PLC or similar devices. Refer to these diagrams and examples when creating sequence programs.

Terms "ON" and "OFF" used in this chapter mean "on" and "off" of switches connected to the input terminal when referring to input signals. They also mean "on" and "off" of output transistors when referring to output signals.

3-1 I/O Signals

The standard I/O connector of the DRCX controller has 48 pins, with an individual signal assigned to each pin. The following table shows the pin number as well as the name and description of each signal assigned to each pin. For a more detailed description of each signal, refer to "3-2 Input Signal Description" and onwards.

No.	Pin No.	Signal name	Description
1	A1	ABS-PT	Absolute point movement command
2	B1	INC-PT	Relative point movement command
3	A2	AUTO-R	Automatic operation start command
4	B2	STEP-R	Step operation start command
5	A3	ORG-S	Return-to-origin command
6	B3	RESET	Reset command
7	A4	SERVO	Servo recovery command
8	B4	LOCK	Interlock
9	A5	DI 0	General-purpose input 0
10	B5	DI 1	General-purpose input 1
11	A6	DI 2	General-purpose input 2
12	B6	DI 3	General-purpose input 3
13	A7	DI 4	General-purpose input 4
14	B7	DI 5	General-purpose input 5
15	A8	DI 6	General-purpose input 6
16	B8	DI 7	General-purpose input 7
17	A9	DI 8	General-purpose input 8
18	B9	DI 9	General-purpose input 9
19	A10	DI 10	General-purpose input 10
20	B10	DI 11	General-purpose input 11
21	A11	DI 12	General-purpose input 12
22	B11	DI 13	General-purpose input 13
23	A12	DI 14	General-purpose input 14
24	B12	DI 15/SVCE	General-purpose input 15/SERVICE mode input
25	A13	+IN COM	External +24V power supply input for controller
26	B13	+IN COM	External +24V power supply input for controller
27	A14	+24V	Internal +24V power supply output for controller
28	B14	+24V	Internal +24V power supply output for controller
29	A15	0V	Reference 0V for input/output
30	B15	0V	Reference 0V for input/output
31	A16	DO 0	General-purpose output 0
32	B16	DO 1	General-purpose output 1
33	A17	DO 2	General-purpose output 2
34	B17	DO 3	General-purpose output 3
35	A18	DO 4	General-purpose output 4
36	B18	END	End-of-run output
37	A19	BUSY	Command-in-progress output
38	B19	READY	Ready-to-operate output
39	A20	DO 5	General-purpose output 5
40	B20	DO 6	General-purpose output 6
41	A21	DO 7	General-purpose output 7
42	B21	DO 8	General-purpose output 8
43	A22	DO 9	General-purpose output 9
44	B22	DO 10	General-purpose output 10
45	A23	DO 11	General-purpose output 11
46	B23	DO 12	General-purpose output 12
47	A24	EMG 1	Emergency stop input 1 (used with EMG2)
48	B24	EMG 2	Emergency stop input 2 (used with EMG1)

CAUTION

Terminals A14 and B14 are used as the output terminals of the internal 24V power supply. Do not connect these terminals to the external 24V power supply. Otherwise the controller might malfunction.

NOTE

Terminal B12 functions as the SERVICE mode input terminal only when the SERVICE mode function is enabled.

3-2 Input Signal Description

Input signals consist of 7 dedicated command inputs, 16 general-purpose inputs, interlock signals and an emergency stop input.

* DI15 functions as the SERVICE mode input when the SERVICE mode function is enabled. In this case, 15 general-purpose inputs are available.

All input circuits other than the emergency stop input use photocoupler-isolated input circuit specs. Only the emergency stop input circuit uses contact point input circuit specs. This contact point is directly connected to the relay coil that turns the internal motor power supply on and off.

3-2-1 Dedicated command input

The dedicated command input is used to control the DRCX controller from a PLC or other external equipment. To accept this input, the READY, BUSY and LOCK signals must be set as follows.

■ READY signal : ON
■ BUSY signal : OFF
■ LOCK signal : ON

If the above conditions are not satisfied, then dedicated command inputs cannot be accepted even if they are input from external equipment. For example, when the BUSY signal is on, this means that the controller is already executing a dedicated command, so other dedicated commands are ignored even if they are input. When the LOCK signal is off, no other commands can be accepted since an interlock is active. (One exception is the reset and servo recovery commands that can be executed even when the LOCK signal is off as long as the READY and BUSY signals meet the above conditions.)

A dedicated command input is accepted when the dedicated command input is switched from "off" to "on" (at the instant the contact point closes). Whether the controller accepts the command or not can be checked by monitoring the BUSY signal.
Note that dedicated command inputs cannot be used as data in a program.

⚠ ***CAUTION***
The dedicated command inputs explained below must always be pulse inputs. In other words, they must be turned off (contact open) after the BUSY signal turns on.
If a dedicated command input is not turned off, then the BUSY signal will remain on even when the command has ended normally. So the next command will not be accepted.

⚠ ***CAUTION***
When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function" for more details.)
- ***No dedicated commands can be executed in "SERVICE mode state" when command input from other than the TPB is prohibited.***

■ Absolute point movement command (ABS-PT)

This command moves the robot to an absolute position of a point number specified by DI0 to DI9 along an axis coordinate whose origin is defined as 0, at a speed selected by DI10 or DI11. (See "3-2-2 General-purpose input (DI0 to DI15)"). The robot axis can be specified with DI13 and DI14 by enabling PRM10. (Refer to "PRM10: Control axis selection with I/O command".)

CAUTION

The DI0 to DI11 status must be confirmed before ABS-PT is executed. DI13 and DI14 must also be determined when selecting an axis. (See "3-6-6 When executing a point movement command".)

■ Relative point movement command (INC-PT)

This command moves the robot a distance defined by a point number specified by DI0 to DI9 from the current position at a speed selected by DI10 or DI11. The robot axis can be specified with DI13 and DI14 by enabling PRM10. (Refer to "PRM10: Control axis selection with I/O command".)

NOTE

Current position does not always indicate the actual robot position. More accurately, it is the current position data stored in the controller. Each time a movement command is executed correctly, the current position data in the controller is replaced with the target position data of the movement command.
Therefore, if the robot is stopped by an interlock while executing a relative movement command, re-executing the same relative movement command moves the robot to the target position. (The robot does not move a relative distance from the stopped position by the interlock.)
Similarly, after a robot movement command is executed, the controller still retains the target position data of that movement command as the current position data even if you move the robot to another position by manual operation.
When a relative movement command is executed under this condition, the robot moves the specified distance from the target position of the movement command that was previously executed, rather than the actual robot position, so use caution.
Current position data differs from the actual robot position when:

- *Emergency stop or interlock (LOCK) was activated while the robot was moving.*
- *A communication command ^C (movement interruption) was transmitted while the robot was moving.*
- *The SERVICE mode input was changed while the robot was moving.*
- *The robot was moved by manual operation.*
- *The robot was moved by hand during servo-off (including emergency stop).*

CAUTION

The DI0 to DI11 status must be specified before INC-PT is executed. DI13 and DI14 must also be determined when selecting an axis. (See "3-6-6 When executing a point movement command".)

■ Automatic operation start command (AUTO-R)

This command executes the robot program continuously, starting from the current step.

All tasks are executed if the robot program is a multi-task program.

■ Step operation start command (STEP-R)

This command executes the robot program one step at a time, starting from the current step.

Only the selected task is executed even if the robot program is a multi-task program.

■ Return-to-origin command (ORG-S)

This command returns the robot to its origin position when the search method is selected as the origin detection method. When the mark method is selected, this command checks the return-to-origin status. The return-to-origin axis can be specified with DI13 and DI14 by enabling PRM10. (Refer to "PRM10: Control axis selection with I/O command".)

NOTE

When both the mark method and search method are used for the same robot, then return-to-origin on the axis using the mark method must first be completed before performing return-to-origin on the axis using the search method. Use the TPB to perform return-to-origin on the axis using the mark method.

NOTE

Once return-to-origin is performed after the robot cable and absolute battery are connected, there is no need to repeat it even when the controller is turned off. (As an exception, return-to-origin becomes incomplete if the absolute backup function is disabled or a parameter relating to the origin is changed. Return-to-origin must be reperformed in that case.)

CAUTION

When performing return-to-origin by the stroke-end detection method, do not interrupt return-to-origin operation while the origin position is being detected (robot is making contact with its mechanical limit). Otherwise, the operation will stop due to a controller overload alarm and the power will need to be turned off and back on again.

CAUTION

If return-to-origin must be repeated by the stroke-end detection method, wait at least 5 seconds before repeating it.

■ Servo recovery command (SERVO)

After emergency stop, releasing the emergency stop button and turning this input on (closing the contact) turns the servo power on, so the robot is ready for restart.
(As with other dedicated command inputs, the servo recovery command should be a pulse input, so it must be turned off (contact open) when the BUSY signal turns on.)

The axis to turn the servo power on can be specified with DI13 and DI14 by enabling PRM10. (Refer to "PRM10: Control axis selection with I/O command".)

■ Reset command (RESET)

This command returns the program step to the first step of the lead program and turns off DO0 to DO12 and the memory I/O. It also clears the point variable "P" to 0. (Counter variables "C" and "D" are not cleared.)

* When PRM2 ("Operation at return-to-origin complete" parameter) is set to 1 or 3, DO5 does not turn off even if the reset command is executed. Likewise, when PRM21 ("Servo status output" parameter) is set to 1, DO7 does not turn off even if the reset command is executed.

NOTE

The lead program is the program that has been selected as the execution program by the TPB or POPCOM. (See "9-4 Switching the Execution Program".)
The lead program can also be selected by executing a communication command "@SWI". It may also be selected when the program data is loaded into the DRCX controller from the memory card.

3-2-2 General-purpose input (DI0 to DI15)

These general-purpose inputs are available to users for handling data input in a program.
These inputs are usually connected to sensors or switches. These inputs can also be directly connected to a PLC output circuit.

As a special function during execution of an ABS-PT or INC-PT point movement command, DI0 to DI9 can be used to specify the point numbers and DI10 and DI11 to specify the movement speed.
As the table below shows, the point numbers should be input with DI0 to DI9 in binary code, to specify P0 to P999. The movement speed is specified as 100% when DI10 and DI11 are off. In other cases, it is set to the speed specified by the parameter. (See "5-2 Parameter Description".)
In addition, DI13 and DI14 can be used to select the robot axis by parameter settings. (See "5-2 Parameter Description".)

Example of point number setting

DI No. / Point No.	DI9 (2^9)	DI8 (2^8)	DI7 (2^7)	DI6 (2^6)	DI5 (2^5)	DI4 (2^4)	DI3 (2^3)	DI2 (2^2)	DI1 (2^1)	DI0 (2^0)
P0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
P1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
P3	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
P7	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
P15	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON
P31	OFF	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON
P63	OFF	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON
P127	OFF	OFF	OFF	ON	ON	ON	ON	ON	ON	ON
P254	OFF	OFF	ON	ON	ON	ON	ON	ON	ON	OFF
P511	OFF	ON	ON	ON	ON	ON	ON	ON	ON	ON
P999	ON	ON	ON	ON	ON	OFF	OFF	ON	ON	ON

Example of point movement speed setting

DI11	DI10	Movement speed
OFF	OFF	100%
OFF	ON	PRM5
ON	OFF	PRM6
ON	ON	PRM7

Example of point movement axis specification

DI14	DI13	Movement axis
OFF	OFF	All axes
OFF	ON	X-axis
ON	OFF	Y-axis
ON	ON	All axes

* DI15 functions as the SERVICE mode input when the SERVICE mode function is enabled. In this case, DI0 to DI14 can be used as the general-purpose inputs.

3-2-3 SERVICE mode input (SVCE)

When the SERVICE mode function is enabled, DI15 functions as the SERVICE mode input (SVCE). The SERVICE mode input is used to notify the DRCX controller whether the current state is a "SERVICE mode state". This input should be turned off (contact open) in "SERVICE mode state".
Refer to "10-4 SERVICE mode function" for details on the SERVICE mode function.

NOTE
Operation stops immediately if the SERVICE mode input status is changed during robot operation while the SERVICE mode function is enabled.

NOTE
Even with the SERVICE mode function enabled, the SERVICE mode input status can be checked in the program as DI15.

3-2-4 Interlock (LOCK)

This input is used to temporarily stop robot movement.
The robot immediately stops when this input is turned off (contact open) during execution of a dedicated I/O command or during program operation or return-to-origin operation from the TPB (or PC). (This also interrupts the robot program operation.)
As long as this input is off (contact open), no dedicated I/O commands can be executed, and also no programs and return-to-origin operation can be performed from the TPB (or PC). The only exceptions to this are the reset command and servo recovery command that can be executed regardless of whether the LOCK signal is on or off. Leave this LOCK signal turned on (contact closed) during normal operation.
Once this LOCK signal is turned off (contact open), the robot remains stopped even after this input is turned back on (contact closed), until another command (AUTO-R, ORG-S, etc.) is input.

3-2-5 Emergency stop inputs 1, 2 (EMG1, EMG2)

Use these inputs to trigger robot emergency stop from an external safety device (for example, safety enclosure, manual safety switch, etc.). Servo power turns off at the same time when the contact between EMG1 and EMG2 is open (turned off). Use a relay contact with a current capacity of at least 50mA.
To resume operation, close (turn on) the contact between EMG1 and EMG2, check that the READY signal is turned on, and then input the servo recovery command (SERVO). The servo will turn on to enable robot operation.
The TPB or PC can also be used to reset emergency stop when the DRCX controller is connected to the TPB or PC.

3-3 Output Signal Description

The output signals consist of 3 dedicated outputs (READY, BUSY and END) and 13 general-purpose outputs. In this section, terms "ON" and "OFF" mean the output transistors are "on and off".

3-3-1 Dedicated output

The dedicated outputs are used for exchanging signals between the DRCX controller and an external device such as a PLC.

■ Ready-to-operate output (READY)

This output is on as long as the DRCX controller system is in normal operation. If an emergency stop or alarm occurs, then this output turns off to let the motor idle.

- When emergency stop was triggered:
  The READY signal turns on again when emergency stop is cancelled.

  Operation can be restarted by input of the servo recovery command (SERVO) after canceling emergency stop.
- When an alarm was issued:
  If the READY signal is off while the robot is not in emergency stop, this means that an alarm was issued. If this happens, correct the problem while referring to Chapter 13, "Troubleshooting".

  In this case, the DRCX controller should be turned off before attempting to restart operation.

■ Command-in-progress output (BUSY)

The BUSY signal is on during execution of a dedicated command input or a command from the TPB or PC. The BUSY signal turns on when the DRCX controller accepts a dedicated command input. The dedicated command input should be turned off (contact open) when the BUSY signal turns on. The BUSY signal turns off when command execution is complete. (At this point, all the dedicated command inputs must be turned off (contact open).)

CAUTION

The dedicated command input must be a pulse input so that it is off when the BUSY signal turns on. If the command input is left on, the BUSY signal cannot turn off even after the command execution is complete.
As long as the BUSY signal is on, the DRCX controller will not accept other dedicated command inputs or commands from the TPB or PC. Avoid operating the TPB while the DRCX controller is being operated through the I/O interface. (Doing so might cause malfunctions during data exchange with a PLC or cause communication errors on the TPB side.)

■ End-of-run output (END)

The END signal turns off when a dedicated command input is received and turns on when command execution is complete. The END signal remains off if an error occurs or an interlock or emergency stop is triggered during command execution.

CAUTION

When a reset command or a movement command specifying a very small amount of movement is used, the command execution time will be very short. In other words, the period that the END signal is off will be very short (1ms or less in some cases).
The END signal does not change by operation from the TPB or PC.

NOTE

The PRM20 (system mode selection parameter) setting can be changed so that the execution result END signal output for the completed dedicated command occurs only after the dedicated command input turns off. (See section 5-2 "Parameter Description".)

3-3-2 General-purpose output (DO0 to DO12)

These general-purpose outputs are available to users for freely controlling on/off operation in a program.
These outputs are used in combination with the controller's internal 24V power supply and an external 24V power supply, to drive loads such as solenoid valves and LED lamps. These outputs, of course, can be directly connected to a PLC input circuit.
All general-purpose outputs are reset (turned off) when the DRCX controller is turned on or the program is reset.

* When PRM2 ("Operation at return-to-origin complete" parameter) is set to 1 or 3, DO5 does not turn off even if the program is reset. Similarly, when PRM21 ("Servo status output" parameter) is set to 1, DO7 does not turn off even if the program is reset.

3-4 I/O Circuits

This section provides the DRCX controller I/O circuit specifications and examples of how the I/O circuits should be connected. Refer to these specifications and diagrams when connecting to external equipment such as a PLC.

3-4-1 I/O circuit specifications

■ **Input Circuit**

Excluding emergency stop input circuit

Insulation method:	Photocoupler insulation
Input terminal:	Relay contact or NPN open collector transistor connected between input terminal and 0V terminal.
Input response:	30ms max.
Input current:	5mA/DC24V
Input sensitivity:	Current on: 3mA min.
	Current off: 1mA max.

Emergency stop input circuit

Input terminal:	Relay contact connected between emergency stop inputs 1 and 2 (between EMG1 and EMG2).
Input response:	5ms max.
Input current:	33.3mA/DC24V

■ **Output Circuit**

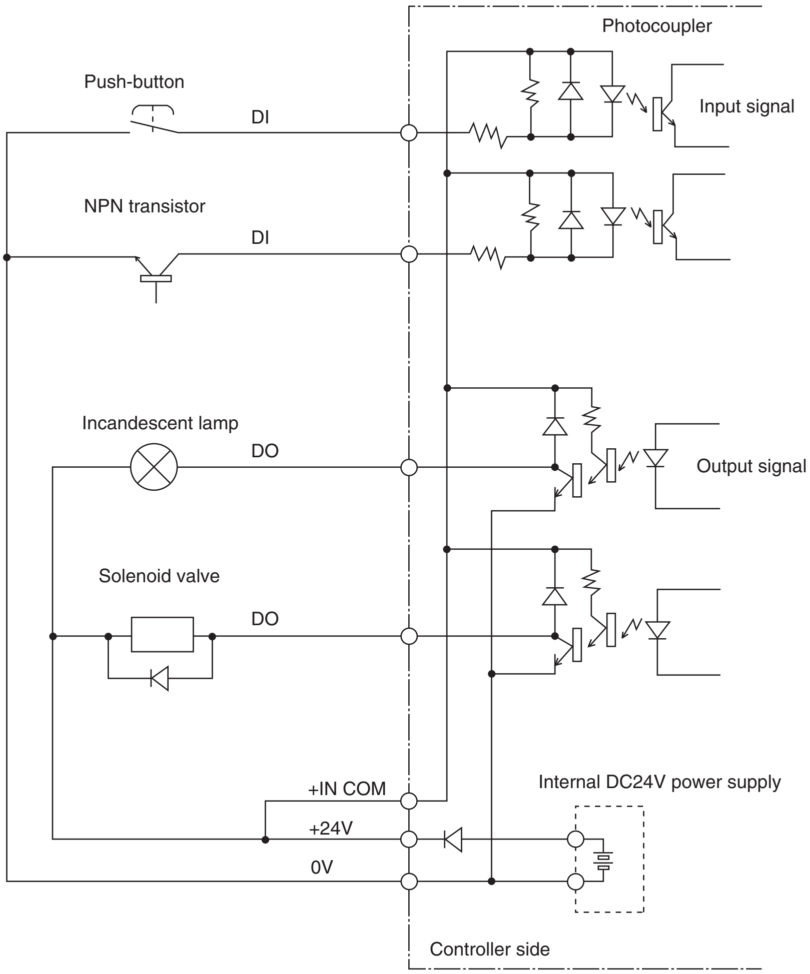
Insulation method:	Photocoupler insulation between internal circuit and output transistor
Output terminal:	NPN open collector output of all collective output common terminals (0V side)
Output response:	1ms max.
Max. output current:	0.5A/DC24V per output
Residual ON voltage:	1.5V max.

■ **Internal Power Supply**

Output voltage:	24V±10%	
Maximum supply current:	Two axes are horizontal type robots	900mA/DC24V
	One axis is horizontal type robot	600mA/DC24V
	Two axes are vertical type robots	300mA/DC24V

3-4-2 I/O circuit and connection example

When internal 24V power supply is used.



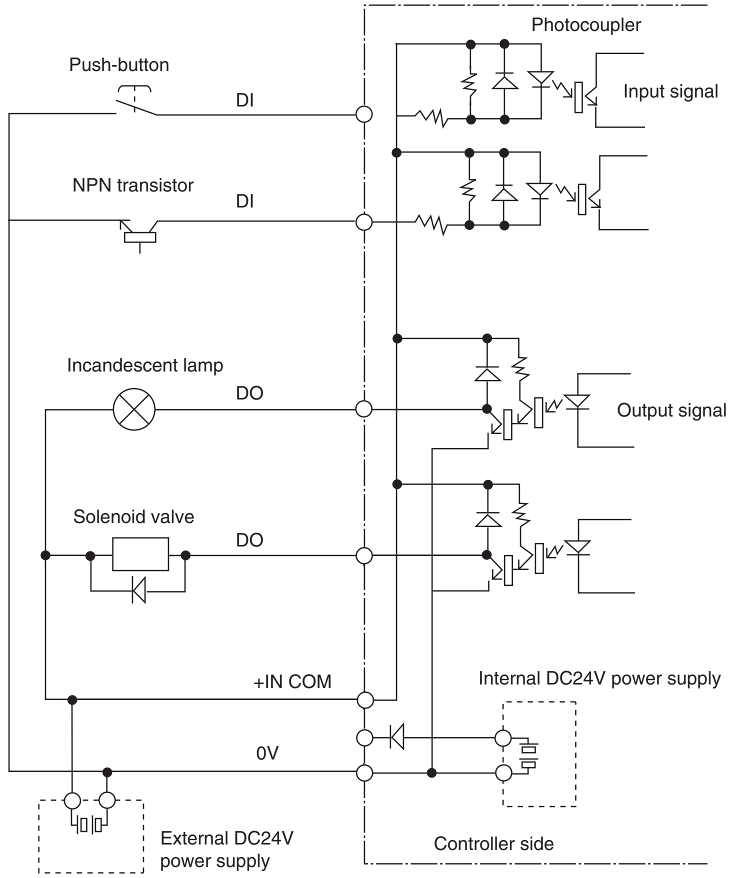
CAUTION

Do not short the output terminal to the DC24V terminal. This may cause equipment breakdown.
When using an inductive load (such as a solenoid valve) as the output load, connect a high-speed diode as a surge killer in parallel and near to the load to reduce noise.
When using a 2-wire type proximity sensor as an input signal, the residual voltage during on/off might exceed the input range for the DRCX controller depending on the sensor type. Using such a sensor will cause erroneous operation. Always check that the sensor meets the input signal specifications.
A large inrush current may flow depending on the output load. If it exceeds the maximum output current of the internal power supply, the protective circuit will work to turn off the output.
Keep the wires separated from the power lines of other machines, or shield the wires to prevent noise.
The supply current capacity of the internal DC24V power supply will differ according to the robot specifications.

Two axes are horizontal type robots	***900mA/DC24V***
One axis is horizontal type robot	***600mA/DC24V***
Two axes are vertical type robots	***300mA/DC24V***

Vertical type robots are robots having a sliding section that moves vertically and has a holding brake.

When external 24V power supply is used.

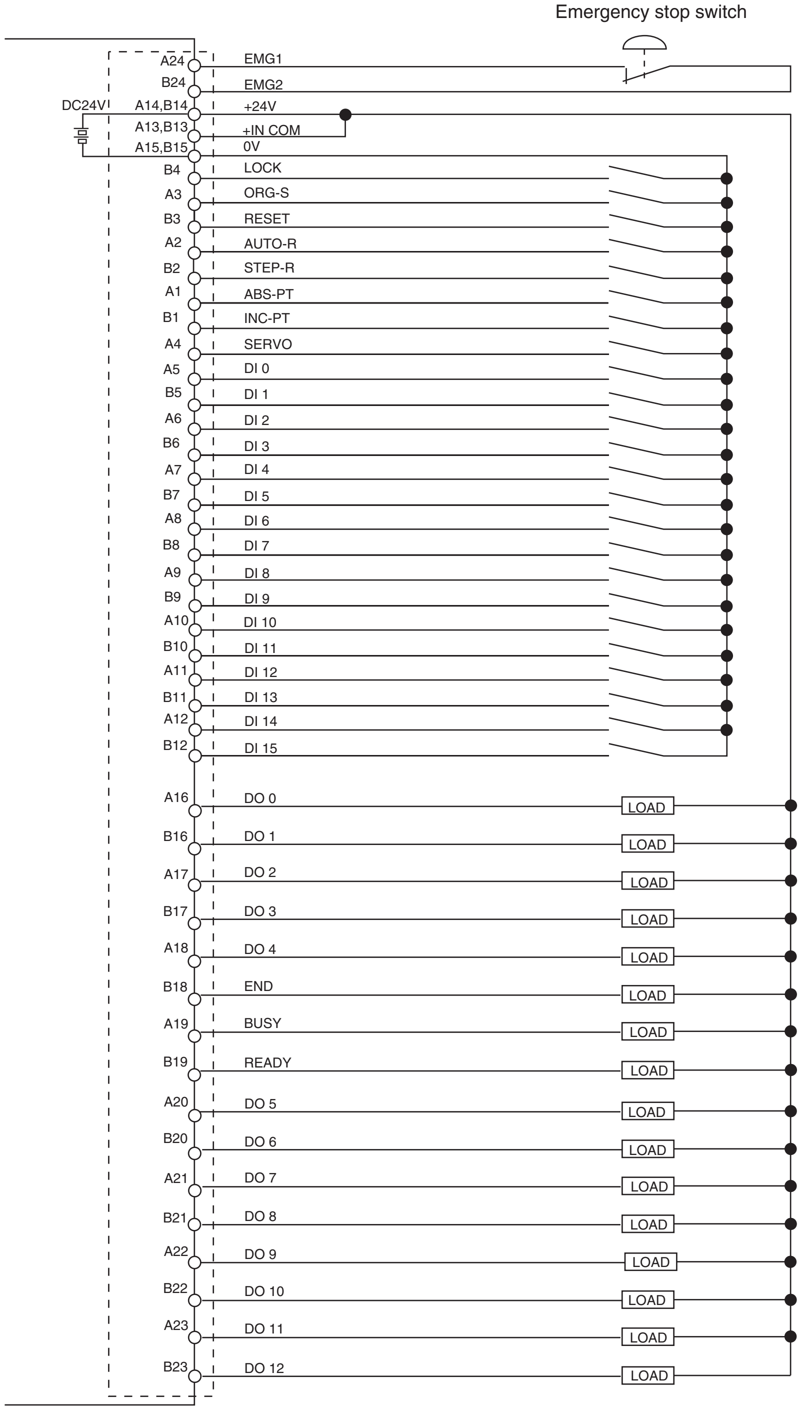


CAUTION
To prevent possible accidents when using an external 24V power supply, do not use it simultaneously with the internal power supply.

3-5 I/O Connection Diagram

3-5-1 General connections for internal 24V power supply

General connections for internal 24V power supply

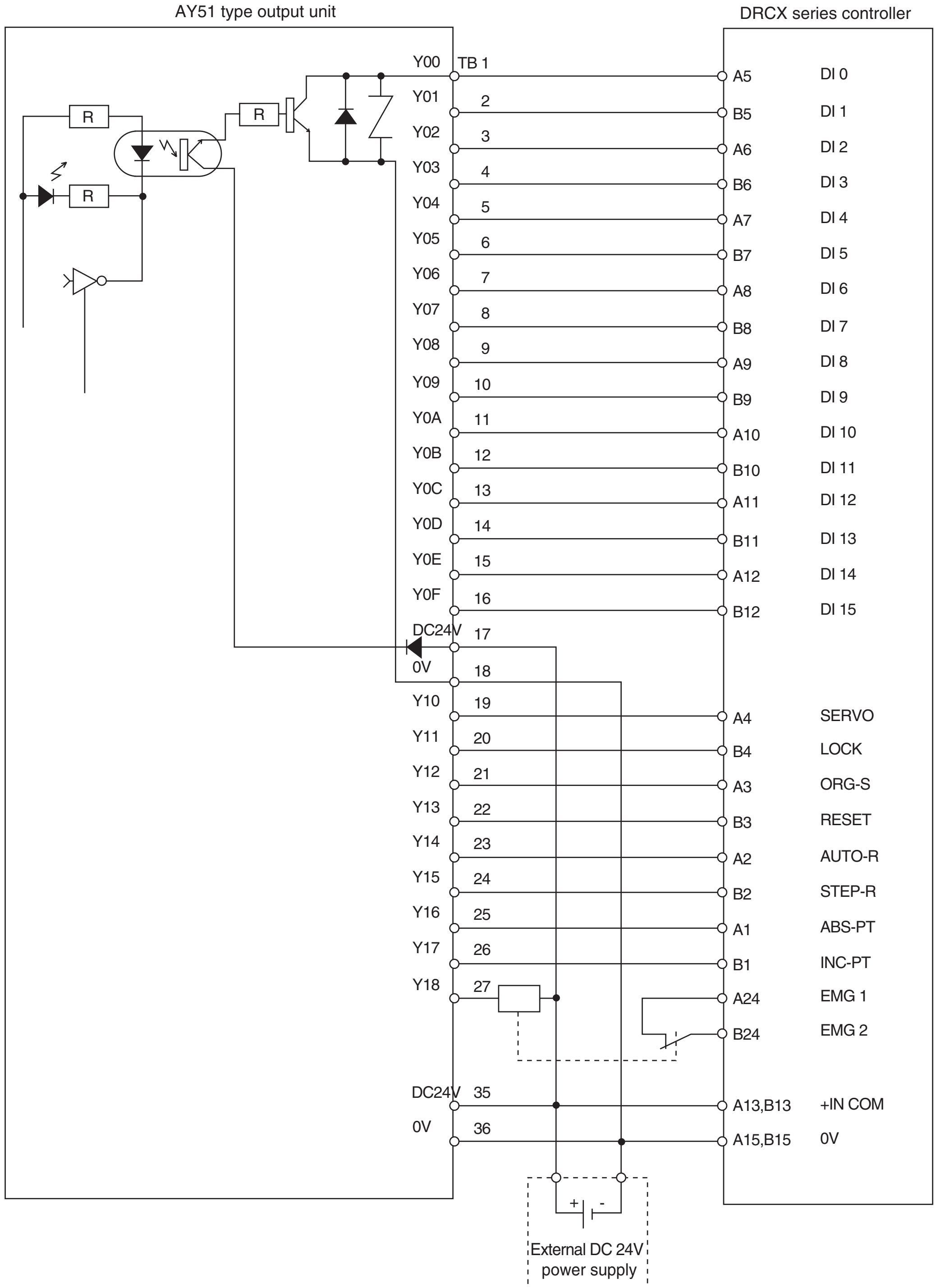


3

I/O INTERFACE

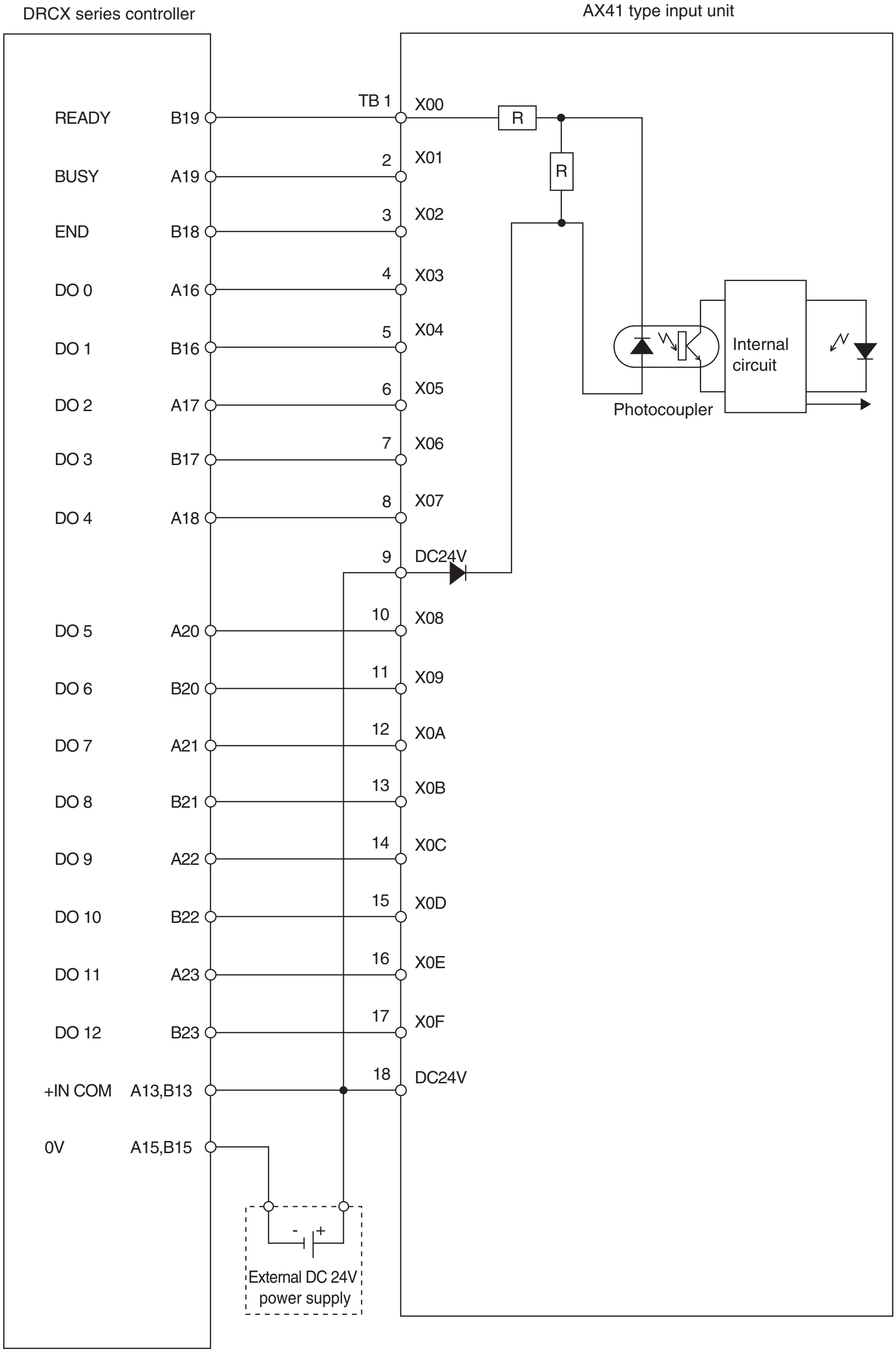
3-5-2 Connection to PLC output unit using external 24V power supply

Connection to the Mitsubishi© PLC AY51 output unit



3-5-3 Connection to PLC input unit using external 24V power supply

Connection to the Mitsubishi© PLC AX41 input unit

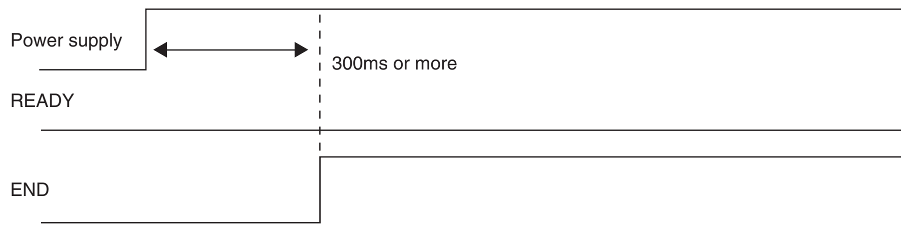


3-6 I/O Control Timing Charts

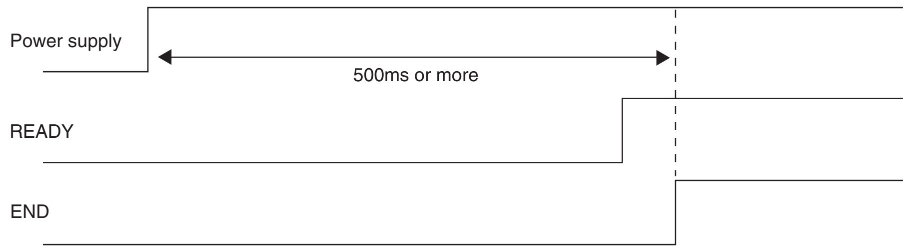
The following shows typical timing charts for I/O control. Refer to these diagrams when creating a sequence program.

3-6-1 When turning the power on

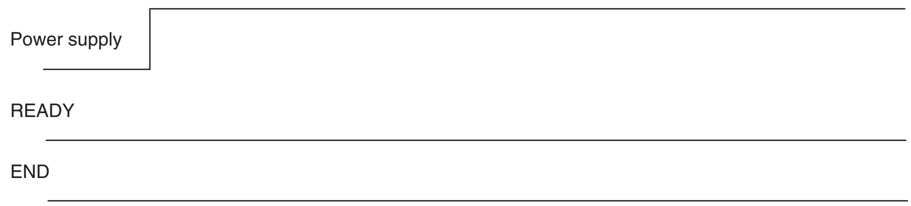
When emergency stop is triggered:



When emergency stop is canceled:



When an alarm is issued:



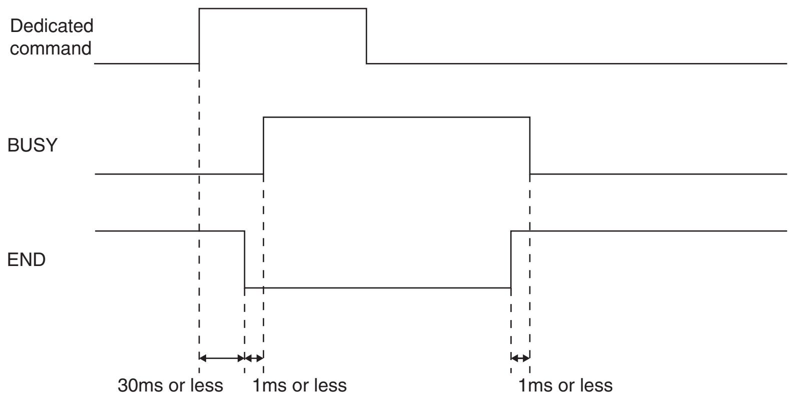
- The DRCX initial state depends on whether emergency stop is triggered when the power is turned on.
  When the power is turned on while emergency stop is cancelled, the DRCX controller starts with the READY signal and also the servo turned on. (Robot is ready to operate in this state.)
  In contrast, when the power is on while emergency stop is triggered, the DRCX controller starts with the READY signal turned off under emergency stop conditions. (Robot operation is prohibited in this state.)
  To enable robot operation, cancel the emergency stop to turn on the READY signal, and then input a servo recovery command (SERVO).
- After turning the power on, make sure that the END signal is on before inputting a dedicated command.
- If the READY and END signals are still off for more than the specified time after turning the power on, this means that an alarm has occurred. If that happens, correct the problem while referring to "13-2 Alarm and Countermeasures".

3-6-2 When executing a dedicated input command

■ The BUSY output signal turns on when a dedicated command is received. Whether the received command has ended normally can be checked with the END output signal status at the point that the BUSY signal turns off. When the END output signal is on, this means that the command has ended normally. If it is off, the command has not ended normally.

■ The dedicated command input must be a pulse input. If the dedicated command input stays on, the BUSY signal does not turn off even after the command has been executed.

(1) When a command with a long execution time runs and ends normally:
(Command execution is still in progress and the END signal is off when turning off (contact open) the dedicated command input.)



(1) At the rising edge of the dedicated command input, the END signal turns off and the BUSY signal turns on.

(2) Turn off (contact open) the dedicated command input after checking that the BUSY signal turns on.

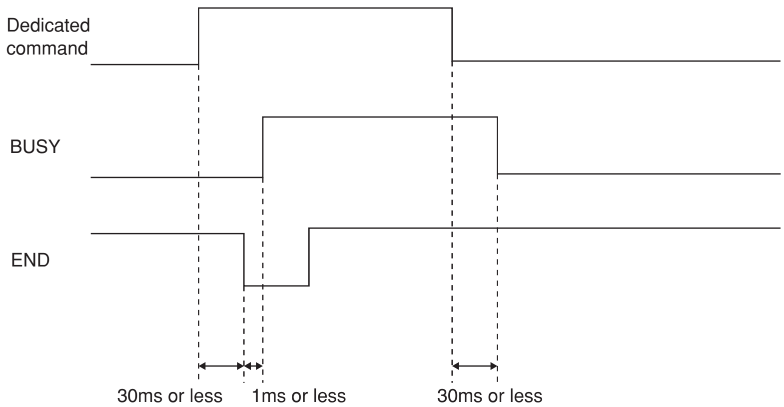
(3) Wait until the BUSY signal turns off.

(4) The END signal should be on when the BUSY signal turns off, indicating that the command has ended normally.

CAUTION
In the case of the automatic operation start command (AUTO-R), the END signal turns on and the BUSY signal turns off when the program ends or a STOP statement is executed. If an endless program (one that automatically returns to the top of the program from the last step) is executed, the BUSY signal will not turn off until an interlock or emergency stop is triggered.

(2) When a command with a short execution time runs and ends normally:
(Command execution has already ended and the END signal is on before turning off (contact open) the dedicated command input, as in the examples listed below.)

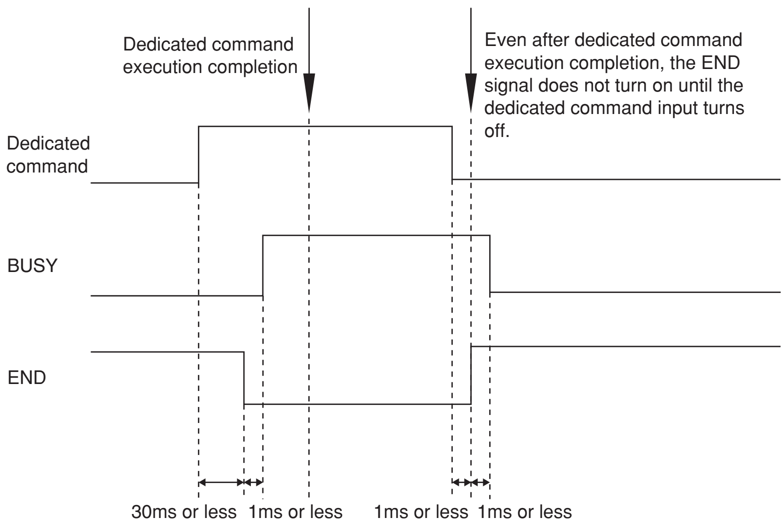
- A movement command (ABS-PT, INC-PT) for a very short distance was executed.
- A reset command (RESET) was executed.
- A step run was executed using a command with a very short execution time such as the L and DO statements.



(1) At the rising edge of the dedicated command input, the END signal turns off and the BUSY signal turns on.
(2) Turn off (contact open) the dedicated command input after checking that the BUSY signal turns on.
(3) Wait until the BUSY signal turns off. (The BUSY signal immediately turns off since the command execution is already complete.)
(4) The END signal should be on when the BUSY signal turns off, indicating that the command has ended normally.

However, the PRM20 (system mode selection parameter) "bit 7 END output sequence setting at command execution completion" setting can be changed so that the END signal turns on when the dedicated command input turns off.

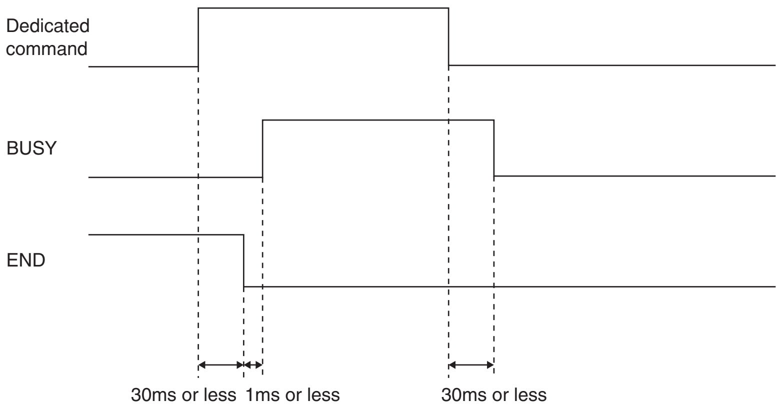
NOTE
The PRM20 (system mode selection parameter) "bit 7 END output sequence setting at command execution completion" setting is supported only in Ver. 18.74 and later versions.



(3) When a command cannot be executed from the beginning:

(Command execution is impossible from the beginning and the END signal does not turn on, as in the examples listed below.)

- A movement command (ABS-PT, INC-PT) was executed without return-to-origin being completed.
- An operation start command (AUTO-R, STEP-R) was executed while return-to-origin is incomplete (except for cases where PRM9 (Pre-operation action selection parameter) is set to 1 or 3).
- A movement command (ABS-PT, INC-PT) was executed by specifying a point number whose point data is unregistered.
- A dedicated command was executed during interlock or emergency stop (except for the reset (RESET) and servo recovery (SERVO) commands).
- When a dedicated command input (ABS-PT, INC-PT, AUTO-R, STEP-R, ORG-S, SERVO, RESET) was executed in “SERVICE mode state”.



(1) At the rising edge of the dedicated command input, the END signal turns off and the BUSY signal turns on.

(2) Turn off (contact open) the dedicated command input after checking that the BUSY signal turns on.

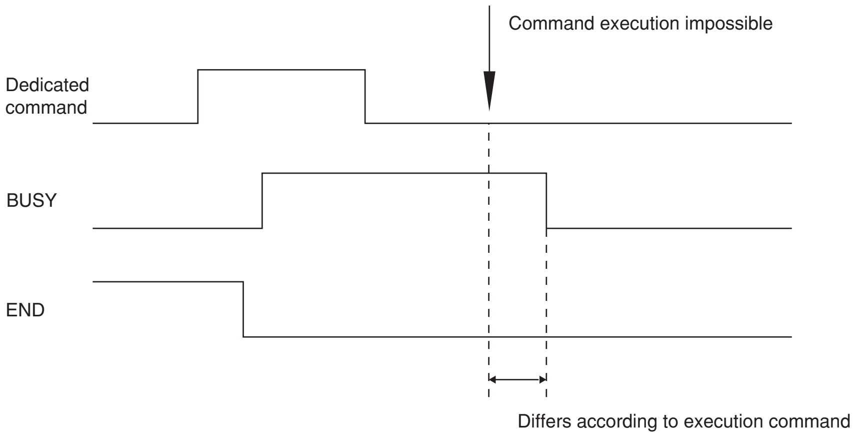
(3) Wait until the BUSY signal turns off. (The BUSY signal immediately turns off since the command cannot be executed from the beginning.)

(4) The END signal remains off when the BUSY signal turns off, indicating that the command could not end normally.

(4) When command execution cannot be completed:

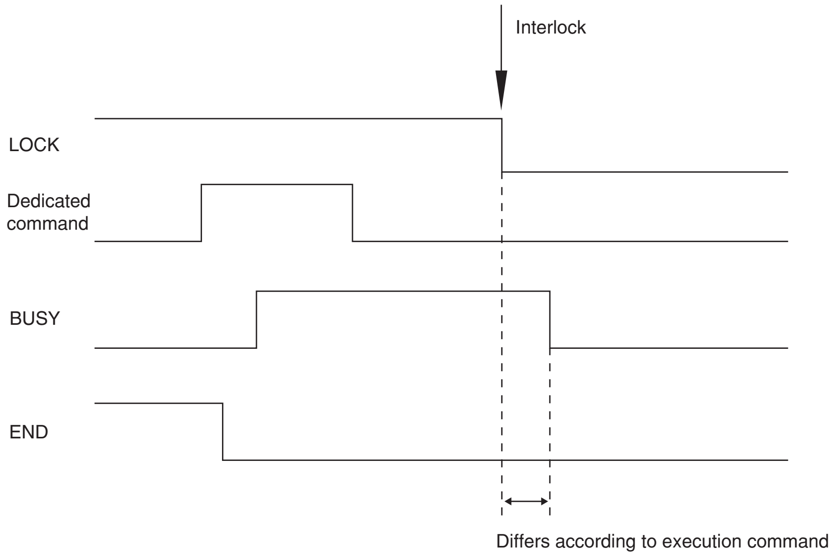
(Command execution stops before completion and the END signal does not turn on, as in the examples listed below.)

- An interlock or emergency stop was triggered during execution of a dedicated command.
- The SERVICE mode input was changed during execution of a dedicated command.
- An error was caused due to a jump to an unregistered program or point during automatic operation.



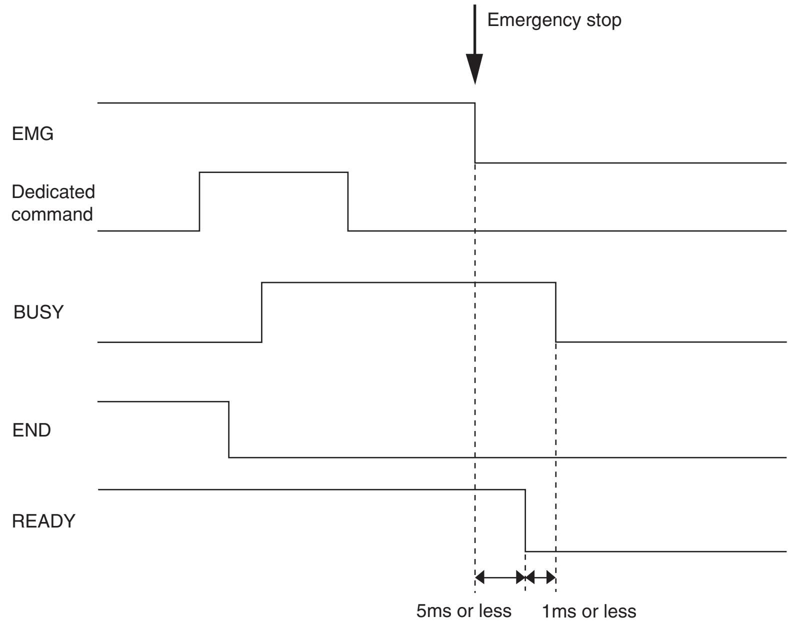
(1) At the rising edge of the dedicated command input, the END signal turns off and the BUSY signal turns on.
(2) Turn off (contact open) the dedicated command input after checking that the BUSY signal turns on.
(3) Wait until the BUSY signal turns off.
(4) The BUSY signal turns off since the command execution stops before completion.
(5) The END signal remains off when the BUSY signal turns off, indicating that the command could not end normally.

3-6-3 When interlock signal is input



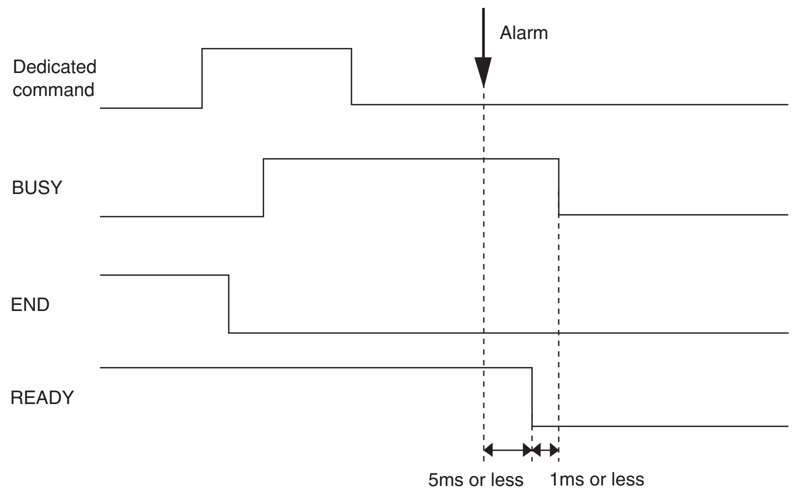
■ When a interlock signal is input while a dedicated command is being executed, the BUSY signal turns off. The READY and END signals remain unchanged.

3-6-4 When emergency stop is input



- The READY signal turns off. The BUSY signal also turns off while a dedicated command is being executed. The END signal remains unchanged.
- To enable robot operation, cancel emergency stop to turn on the READY signal, then input the servo recovery command (SERVO).

3-6-5 When alarm is issued



- The READY, BUSY and END signals all turn off.
- Correct the problem while referring to "13-2 Alarm and Countermeasures".

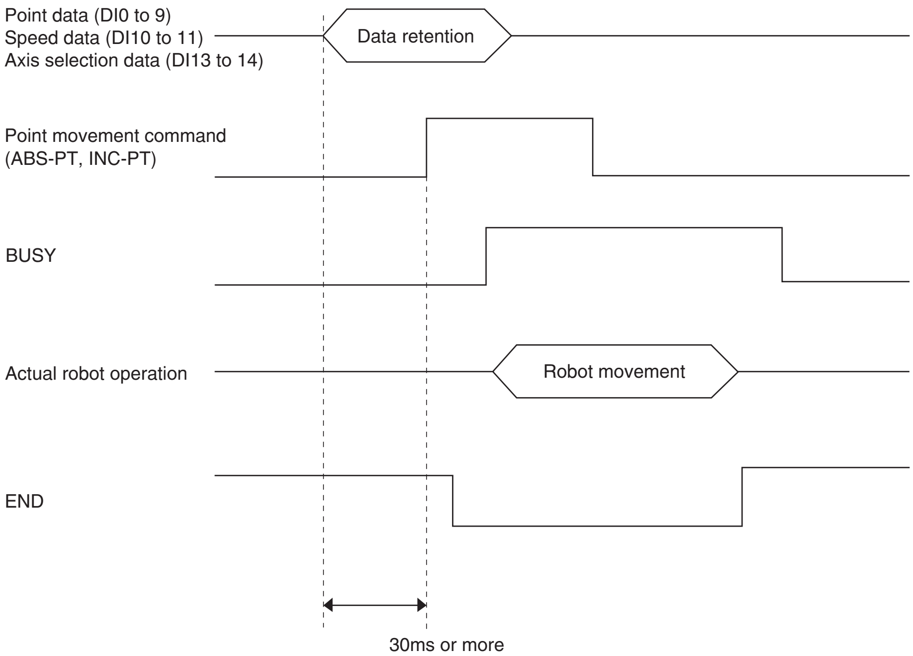
3

I/O INTERFACE

3-6-6 When executing a point movement command

■ When executing a point movement command (ABS-PT, INC-PT), the point data and speed data must first be input before inputting the command. When specifying the robot axis, the axis selection data must be input.

The point data and speed data can be specified with DI0 to DI11. Refer to "3-2-2 General-purpose input (DI0 to DI15)". The axis selection data input is designated with DI13 to DI14. (Refer to "3-2-2 General-purpose input (DI0 to DI15)".)



(1) Specify the point data and speed data, using the general-purpose input DI0 to DI11. When specifying the robot axis, the axis selection data should be input to DI13 and DI14. These input conditions should be kept unchanged until the BUSY signal turns on. (If these conditions are changed before the BUSY signal turns on, then the data might be misrecognized.)

(2) When a minimum of 30ms has elapsed, input the point movement command (ABS-PT, INC-PT).

(3) At the rising edge of the dedicated command input, the END signal turns off and the BUSY signal turns on.

(4) Turn off (contact open) the dedicated command input after checking that the BUSY signal turns on.

Then, the point data, speed data (DI0 to DI11) and axis selection data (DI13 and DI14) can be changed for the next movement.

(5) Wait until the BUSY signal turns off.

(6) The END signal should be on when the BUSY signal turns off, indicating that the command has ended normally.

3-7 I/O Assignment Change Function

3-7-1 Changing the I/O assignment

The function assigned to each I/O signal can be changed with PRM26 (I/O assignment selection parameter) setting. Refer to the table on next page when you want to change the I/O assignment.
After changing the I/O assignment, the DRCX controller must be restarted to enable the changes.

NOTE
The I/O assignment change function is available on controllers whose version is 18.57 or later.

I/O assignment list

Type		Type 0 (Conventional type)	Type 1	Type 2 (Point number output type)		Type 3 (Point teaching type)			
						Point trace mode	Teaching mode	Point trace mode	Teaching mode
PRM26 Setting / Function		0 (Standard)	–	xx20 *1	xx21 *1	xx30 *1		xx31 *1	
No. of points *3		1000	–	64	16	64		16	
No. of speed switching points *4		4	–	None	4	None		4	
Program operation by I/O		Yes	–	No	No	No		No	
	Pin No.								
Input	A1	ABS-PT	Cannot be used.	ABS-PT	ABS-PT	ABS-PT	JOG+	ABS-PT	JOG+
	B1	INC-PT		INC-PT	INC-PT	INC-PT	JOG-	INC-PT	JOG-
	A2	AUTO-R		-	-	-	PSET	-	PSET
	B2	STEP-R		-	-	CHG		CHG	
	A3	ORG-S		ORG-S	ORG-S	ORG-S		ORG-S	
	B3	RESET		RESET	RESET	RESET		RESET	
	A4	SERVO		SERVO	SERVO	SERVO		SERVO	
	B4	LOCK		LOCK	LOCK	LOCK		LOCK	
	A5	DI0		PI0	PI0	PI0		PI0	
	B5	DI1		PI1	PI1	PI1		PI1	
	A6	DI2		PI2	PI2	PI2		PI2	
	B6	DI3		PI3	PI3	PI3		PI3	
	A7	DI4		PI4	SPD1	PI4		SPD1	
	B7	DI5		PI5	SPD2	PI5		SPD2	
	A8	DI6		-	-	-		-	
	B8	DI7		-	-	-		-	
	A9	DI8		-	-	-		-	
	B9	DI9		-	-	-		-	
	A10	DI10		-	-	-		-	
	B10	DI11		-	-	-		-	
	A11	DI12		-	-	-		-	
	B11	DI13 *2		*2	*2	*2		*2	
	A12	DI14 *2		*2	*2	*2		*2	
	B12	DI15/SVCE		(SVCE)	(SVCE)	(SVCE)		(SVCE)	
Output	A16	DO0		PO0	PO0	PO0		PO0	
	B16	DO1		PO1	PO1	PO1		PO1	
	A17	DO2		PO2	PO2	PO2		PO2	
	B17	DO3		PO3	PO3	PO3		PO3	
	A18	DO4		PO4	ORG-O/ZONE0	PO4		ORG-O/ZONE0	
	A20	DO5		PO5	SRV-O/ZONE1	PO5		SRV-O/ZONE1	
	B20	DO6		-	-	-		-	
	A21	DO7		-	-	-		-	
	B21	DO8		-	-	-		-	
	A22	DO9		-	-	-		-	
	B22	DO10		-	-	-		-	
	A23	DO11		-	-	-		-	
	B23	DO12		-	-	-		-	
	B18	END		END	END	END		END	
	A19	BUSY		BUSY	BUSY	BUSY		BUSY	
	B19	READY		READY	READY	READY		READY	

*1: The PO output format varies depending on whether the PRM26 setting is specified in "hundreds" or "thousands" units. (See section 5-2 "Parameter Description")
*2: A desired axis can be specified when executing I/O dedicated commands (ABS-PT, INC-PT, ORG-S, SERVO), by specifying DI13 and DI14 (pin No. B11 and A12). To do this, the PRM10 (control axis selection with I/O command) must first be set to "Enable". The same applies when specifying a control axis for JOG motion commands (JOG+, JOG-).
*3: Specifies the permissible number of movement points for a point movement command (ABS-PT, INC-PT).
*4: Specifies the permissible number of speed switching points for a point movement command (ABS-PT, INC-PT).

3-7-2 I/O signal descripion

The meaning of each signal is explained below.

■ Point number designation inputs 0 to 5 (PI0 to PI5)

These inputs designate the point number of the target position where the robot moves with a point movement command (ABS-PT, INC-PT). (For details on the ABS-PT and INC-PT commands, see 3.2.1, "Dedicated command input" in this chapter.)
These inputs are also used to designate the point number of the target position where point data is written with a point data write command (PSET).
The point number of the target position must be specified before running a point movement command or point write command. The point number is specified by a binary code. See the table below to specify each point number.

Point number designation example

PI No. / Point No.	PI5 (2^5)	PI4 (2^4)	PI3 (2^3)	PI2 (2^2)	PI1 (2^1)	PI0 (2^0)
P0	OFF	OFF	OFF	OFF	OFF	OFF
P1	OFF	OFF	OFF	OFF	OFF	ON
P7	OFF	OFF	OFF	ON	ON	ON
P15	OFF	OFF	ON	ON	ON	ON
P31	OFF	ON	ON	ON	ON	ON
P63	ON	ON	ON	ON	ON	ON

■ Movement speed setting (SPD1, SPD2)

Designates the speed at which the robot moves with a point movement command (ABS-PT, INC-PT) or jog movement command (JOG+, JOG-). (For details on the ABS-PT and INC-PT commands, see 3.2.1, "Dedicated command input" in this chapter.)
The movement speed must be specified before running a point movement command or jog movement command. See the table below to specify the movement speed.

Movement speed setting example

SPD2	SPD1	Movement speed
OFF	OFF	100%
OFF	ON	PRM5
ON	OFF	PRM6
ON	ON	PRM7

■ Jog movement (+ direction) command (JOG+)

Moves the robot in jog mode along the + (plus) direction.
The robot moves in jog mode along the + (plus) direction as long as this signal is on. The movement speed is 100mm/sec.
This speed can be changed by using SPD1 and SPD2. In this case, the movement speed is given by the following equation.

Movement speed [mm/sec] = 100 × (Movement speed [%] specified with SPD1 and SPD2) / 100

In jog mode the robot usually moves along the X-axis. However, the Y-axis can be specified with DI13 and DI14 (pin No. B11 and A12) by enabling PRM10 (Control axis selection with I/O command). (Refer to "PRM10: Control axis selection with I/O command".)

CAUTION

- ***If the CHG (mode switch input) signal is switched during jog movement, the robot comes to an error stop.***
- ***When specifying the axis, the DI13 and DI14 (pin No. B11 and A12) status must be checked beforehand.(Refer to the Jog movement (JOG+, JOG-) timing chart in "3-7-3 Timing chart".)***

■ Jog movement (- direction) command (JOG-)

Moves the robot in jog mode along the - (minus) direction.
The robot moves in jog mode along the - (minus) direction as long as this signal is on. The movement speed is 100mm/sec.
This speed can be changed by using SPD1 and SPD2. In this case, the movement speed is given by the following equation.

Movement speed [mm/sec] = 100 × (Movement speed [%] specified with SPD1 and SPD2) / 100

In jog mode the robot usually moves along the X-axis. However, the Y-axis can be specified with DI13 and DI14 (pin No. B11 and A12) by enabling PRM10 (Control axis selection with I/O command). (Refer to "PRM10: Control axis selection with I/O command".)

CAUTION

- ***If the CHG (mode switch input) signal is switched during jog movement, the robot comes to an error stop.***
- ***When specifying the axis, the DI13 and DI14 (pin No. B11 and A12) status must be checked beforehand.(Refer to the Jog movement (JOG+, JOG-) timing chart in "3-7-3 Timing chart".)***

■ Point data write command (PSET)

Writes the current position data in the specified point number.
To use this command, the point number for writing the current position data must first be specified using a PI (point number designation input) input.
The PSET is enabled only when return-to-origin has been completed.

■ Mode switch input (CHG)

Switches the Type 3 (Point teaching type) mode. Selectable modes are as follows.

(1) Point trace mode
(2) Teaching mode

The Type 3 (Point teaching type) mode is switched to "Point trace mode" when the CHG signal is off, and is switched to "teaching mode" when the CHG signal is on.

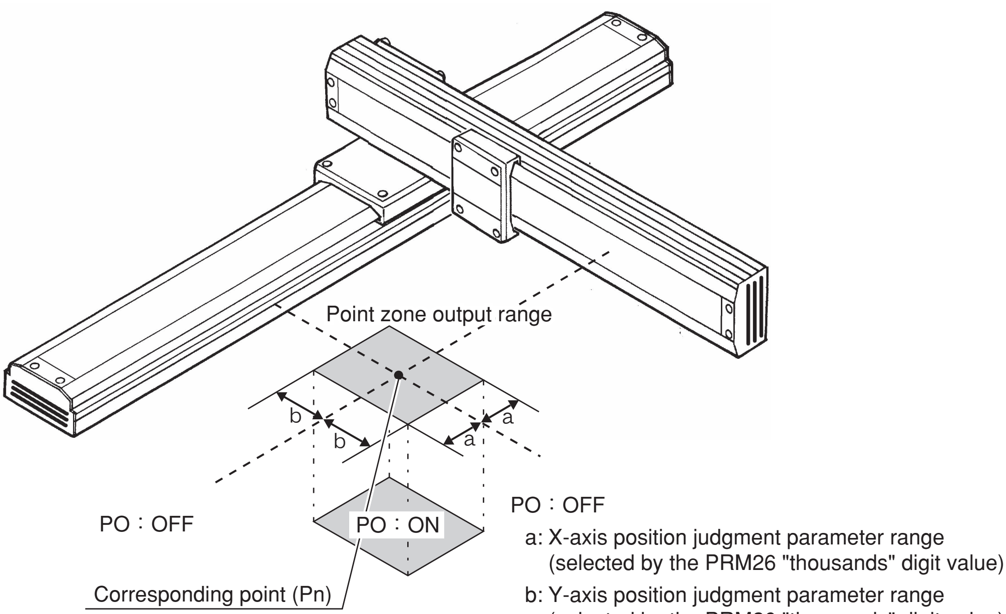
CAUTION

If the CHG signal is switched during execution of a point movement command (ABS-PT, INC-PT) or jog movement command (JOG+, JOG-), the robot comes to an error stop.

■ Target position's point number outputs 0 to 5 (PO0 to PO5)

These are the output signals for the point movement command (ABS-PT, INC-PT) target position point numbers, and for the point numbers corresponding to the point zone output and movement point zone output functions. (For details on ABS-PT and INC-PT commands, see 3.2.1, "Dedicated command input" in this chapter.)
The "point zone output function" outputs the corresponding point number to the PO when the robot enters the point zone output range (corresponding point ± position judgment parameter range). The corresponding point of this point zone output range is the point data registered at the controller.
Moreover, the point zone output range's corresponding point can be further narrowed to correspond to point movement commands (ABS-PT, INC-PT), with the point number being output to the PO. This is referred to as the movement point zone output function.

Point zone output function



Target position point numbers for point movement commands (ABS-PT, INC-PT) are output as binary values. The same applies to point numbers which correspond to the point zone output function and the movement point zone output function.
The PO output format is determined by the PRM26 (I/O assignment selection parameter) setting's "hundreds" digit value.

0: PO output occurs at normal movement completion.
1: PO output occurs when movement command is received.
2: Point zone output
(PO output occurs when the current position enters the point data (registered at the controller) ± position judgment parameter range.)
3: Movement point zone output (supported only by Ver. 18.64 and later versions)
(PO output occurs when the current position enters the point data registered at the controller, and the point movement command's (ABS-PT, INC-PT) movement point data ± position judgment parameter range.)

Output example

PO No. / Point No.	PO5 (2^5)	PO4 (2^4)	PO3 (2^3)	PO2 (2^2)	PO1 (2^1)	PO0 (2^0)
P0	OFF	OFF	OFF	OFF	OFF	OFF
P1	OFF	OFF	OFF	OFF	OFF	ON
P7	OFF	OFF	OFF	ON	ON	ON
P31	OFF	ON	ON	ON	ON	ON
P63	ON	ON	ON	ON	ON	ON

CAUTION

When using PO as an output signal that indicates the target position's point number for point movement commands (ABS-PT, INC-PT):

- ***If moving the robot to point 0 with at the first point movement command which is executed after turning the controller on, all the PO0 to PO5 signals still remain off (because P0 = 000000 (binary)) even after the robot has moved to point 0. This means that the PO0 to PO5 signal statuses do not change even after the robot has moved to P0, so no information is available to indicate whether the robot motion to P0 is complete (or whether the movement command was received). This should be kept in mind when moving the robot to point 0.***

When using PO as an output signal that indicates the corresponding point number at the point zone output function or the movement point zone output function:

- ***If outputting point 0 (P0) as the corresponding point for the point zone output function or the movement point zone output function, all the PO0 to PO5 signals remain off (because P0 = 000000 (binary)). This means that the PO0 to PO5 signal statuses do not change even after the robot has entered the zone specified by P0. This should be kept in mind when monitoring P0.***

NOTE

When using PO as an output signal that indicates the target position's point number for point movement commands (ABS-PT, INC-PT):

- *When a point movement is received through a parallel I/O, the target position's point number is output to the corresponding parallel I/O (PO0 to PO5). When received through a serial I/O such as a CC-Link, the target position's point number is output to the corresponding serial I/O (PO200 to PO205).*
- *All PO outputs are reset (OFF) when a program reset is performed.*

When using PO as an output signal that indicates the corresponding point number at the point zone output function:

- *The corresponding point number for the point zone output function is output to both the corresponding parallel I/O (PO0 to PO5) and the serial I/O (PO200 to PO205).*
- *All PO outputs are reset (OFF) when a program reset is performed.*

When using PO as an output signal that indicates the corresponding point number at the movement point zone output function:

- *The corresponding point number for the movement point zone output function is output to both the corresponding parallel I/O (PO0 to PO5) and the serial I/O (PO200 to PO205).*
- *Movement points are reset immediately after a controller power on, and all PO outputs are therefore turned off at that time. Movement points are also reset if a program reset is performed, and the movement point zone PO outputs are reset (OFF) at that time as well.*

■ Return-to-origin complete output / Zone output 0 (ORG-O / ZONE0)

This output notifies that return-to-origin operation is complete.
The ORG-O output turns on when return-to-origin is complete. It remains off as long as return-to-origin is incomplete.
When Zone 0 output is enabled with PRM24 (Zone output selection parameter), the ORG-O output is used as the output port of Zone 0. For details on the zone output signal, refer to "5.2 Parameter Description".

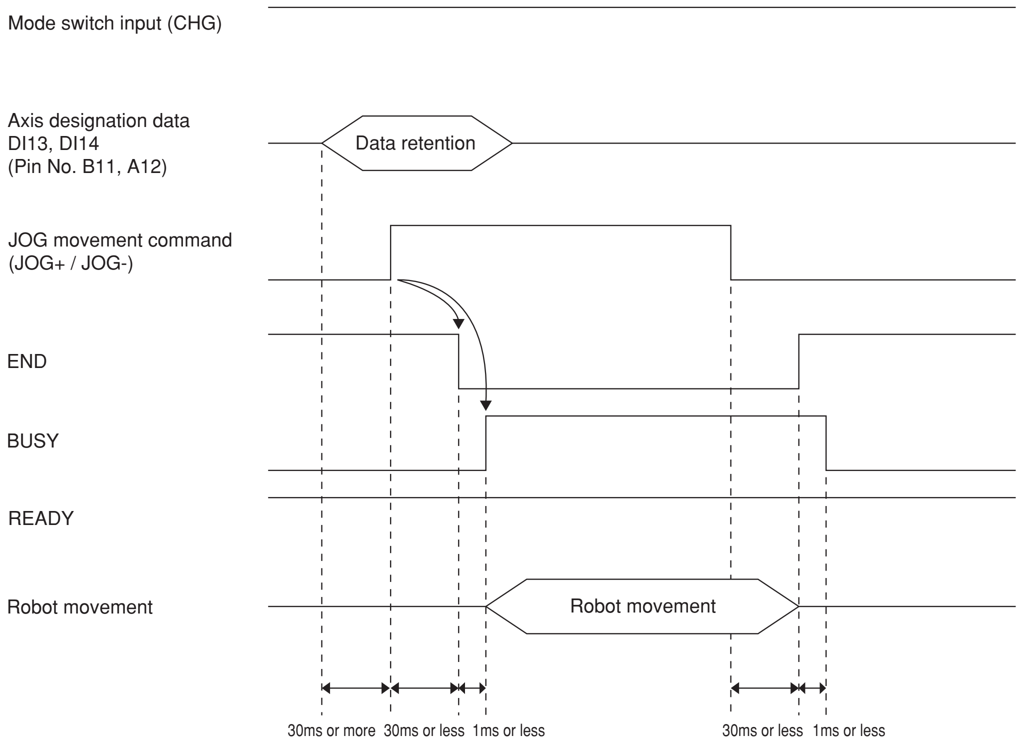
■ Servo status output / Zone output 1 (SRV-O / ZONE 1)

The SRV-O output turns on when the servo is on and turns off when the servo is off. When Zone 1 output is enabled with PRM24 (Zone output selection parameter), the SRV-O output is used as the output port of Zone 1. For details on the zone output signal, refer to "5.2 Parameter Description".

3-7-3 Timing chart

This section shows timing charts for the operations that are added by changing the I/O assignment.

■ Jog movement (JOG+, JOG-)



(1) Turn on the CHG signal.

(2) To specify the movement axis, input the axis designation data to DI13 and DI14 (pin No. B11 and A12).

- The input status specified here must be kept unchanged until step (4) is complete. If this input status is changed, the DRCX controller might misrecognize the data.

(3) Turn on the JOG+ (or JOG-) input signal while the CHG signal is on.

(4) The END signal turns off and the BUSY signal turns on, indicating that the DRCX received the jog movement command.

(5) The robot moves in jog mode as long as the JOG+ (or JOG-) input signal is on.

(6) Turn off the JOG+ (or JOG-) input signal.

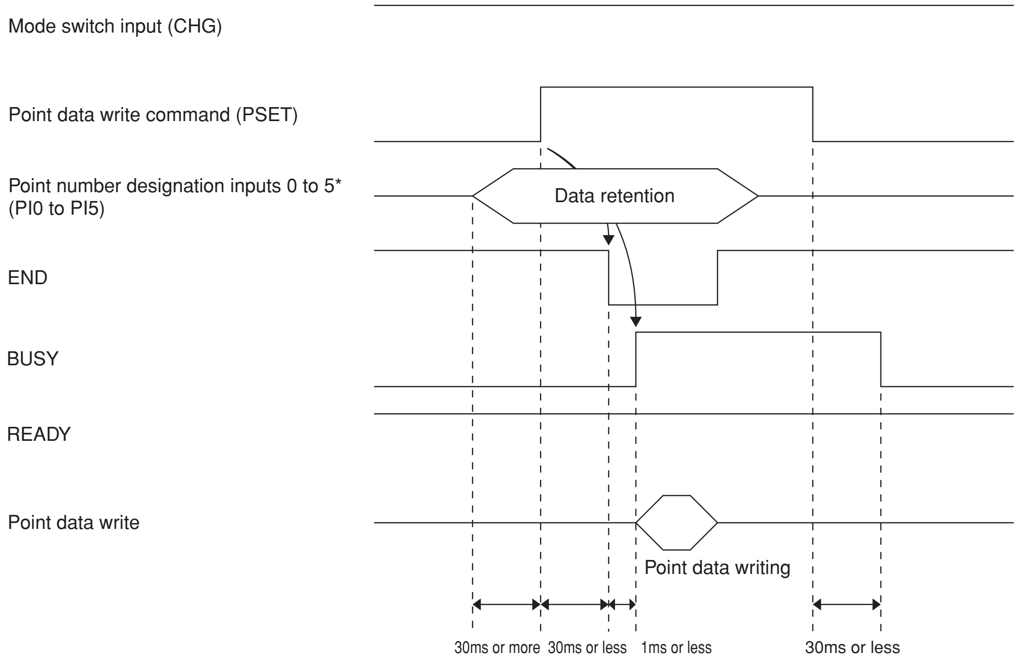
(7) Wait until the BUSY signal turns off.

(8) The BUSY signal turns off. The END signal should be on at this point, indicating that the jog movement is normally complete.

CAUTION

- ***If the CHG signal is switched during execution of a jog movement command (JOG+, JOG-), the robot comes to an error stop and the END signal remains off.***
- ***When specifying the axis, the DI13 and DI14 (pin No. B11 and A12) status must be checked beforehand. (Refer to "PRM10: Control axis selection with I/O command".)***

■ Point data write (PSET)



* Point numbers that can be used depend on the I/O assignment type.

Precondition: The CHG signal is on before and during point data writing (until the following procedure is complete).

(1) Designate the point number input (PI0 to PI5) to write the point data.
- The point numbers that can be used depend on the I/O assignment type. Refer to the I/O assignment list in "3-7-1 Changing the I/O assignment".
- The input status for designating the point number must be kept unchanged until step (3) is complete. If this input status is changed, the DRCX might misrecognize the data.

(2) After 30ms or more has elapsed, turn on the PSET.

(3) The END signal turns off and the BUSY signal turns on, indicating that the DRCX received the point data write command.

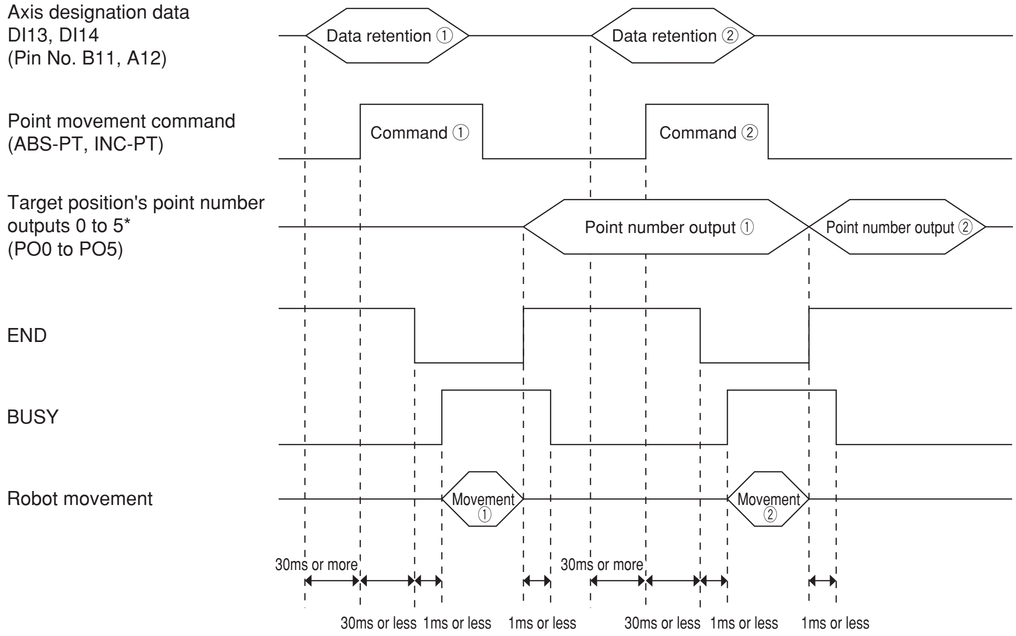
(4) Turn off the PSET.

(5) Wait until the BUSY signal turns off.

(6) The BUSY signal immediately turns off since point data writing is already finished. The END signal should be on at this point, indicating that the point data writing was completed normally.

■ Target position's point number output (PO)

(1) Outputting the point number at the timing that movement is normally completed



* The number of point number outputs that can be used depends on the I/O assignment type.

Precondition: 1) The following steps are explained assuming that PRM26=30.

When the PRM26 setting = 30	
I/O assignment type	Type 3 (point teaching type)
Permissible number of movement points	64 points
Point output selection	Point No. output to PO when movement ends normally

2) The point numbers of the target positions are designated before running a point movement command (ABS-PT, INC-PT).

[Point movement command execution ①]

(1) To specify the movement axis, input the axis designation data to DI13 and DI14 (pin No. B11 and A12).

- The input status specified here must be kept unchanged until step (3) is complete. If this input status is changed, the DRCX controller might misrecognize the data.

(2) Turn on the ABS-PT (or INC-PT).

(3) The END signal turns off and the BUSY signal turns on, indicating that the DRCX received the point movement command.

(4) Turn off the ABS-PT (or INC-PT).

(5) Wait until the BUSY signal turns off.

(6) The BUSY signal turns off. The END signal should be on at this point, indicating that the point movement is normally finished.

(7) When the END signal is on in step (6), the target position's point number is output from the specified point number (PO0 to PO5).

- The output status of the target position's point number is retained until execution of the next point movement command is complete.

↓

[Point movement command execution ②]

(8) Execute the next point movement command.

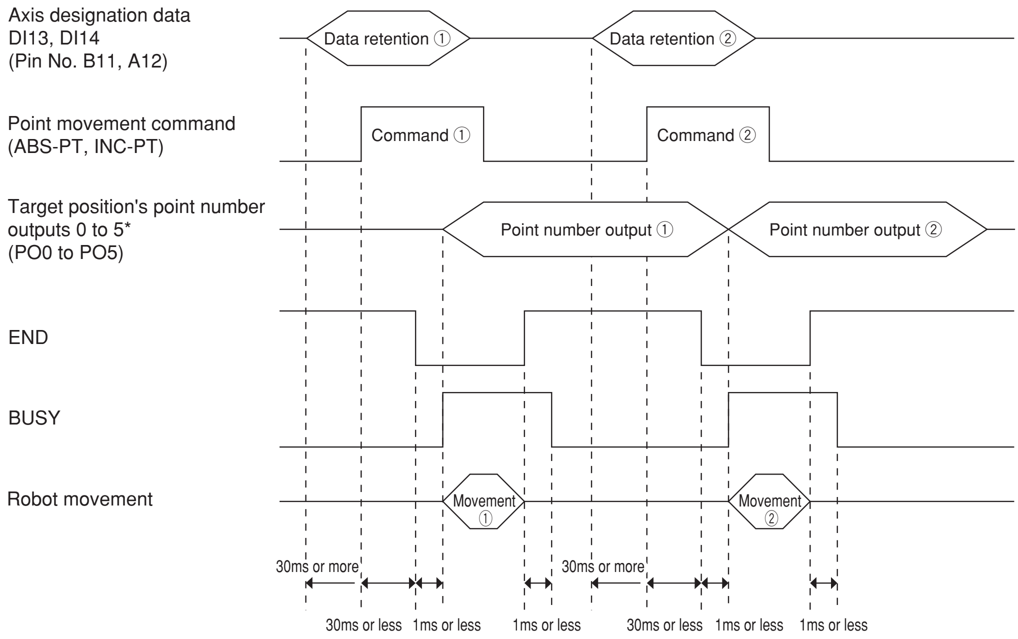
(9) Point movement ends.

(10) The END signal turns on. The previous target position's point number being output from the specified point number (PO0 to PO5) is cleared and the current target position's point number is then output.

CAUTION

- ***If moving the robot to point 0 with a point movement command that is first executed after turning on the controller, all of PO0 to PO5 still remain off (because P0 = 000000 (binary)) even after the robot has moved to point 0. This means that the PO0 to PO5 status does not change even after the robot has moved to P0, so no information is available to indicate whether the robot motion to P0 is complete (or whether the movement command was received). This should be kept in mind when moving the robot to point 0.***
- ***When specifying the axis, the DI13 and DI14 (pin No. B11 and A12) status must be checked beforehand. (Refer to "PRM10: Control axis selection with I/O command".)***

(2) Outputting the point number at the timing that a movement command is received



* The number of point number outputs that can be used depends on the I/O assignment type.

Precondition: 1) The following steps are explained assuming that PRM26=130.

When the PRM26 setting = 130	
I/O assignment type	Type 3 (point teaching type)
Permissible number of movement points	64 points
Point output selection	Point No. output to PO when movement command is received

2) The point numbers of the target positions are designated before running a point movement command (ABS-PT, INC-PT).

[Point movement command execution ①]

(1) To specify the movement axis, input the axis designation data to DI13 and DI14 (pin No. B11 and A12).

- The input status specified here must be kept unchanged until step (3) is complete. If this input status is changed, the DRCX controller might misrecognize the data.

(2) Turn on the ABS-PT (or INC-PT).

(3) The END signal turns off and the BUSY signal turns on, indicating that the DRCX received the point movement command.

(4) When the BUSY signal turns on in step (3), the target position's point number is output from the specified point number (PO0 to PO5).

- The output status of the target position's point number is retained until the next point movement command is received.

(5) Turn off the ABS-PT (or INC-PT).

(6) Wait until the BUSY signal turns off.

(7) The BUSY signal turns off. The END signal should be on at this point, indicating that the point movement finished normally.

↓

[Point movement command execution ②]

(8) Execute the next point movement command.

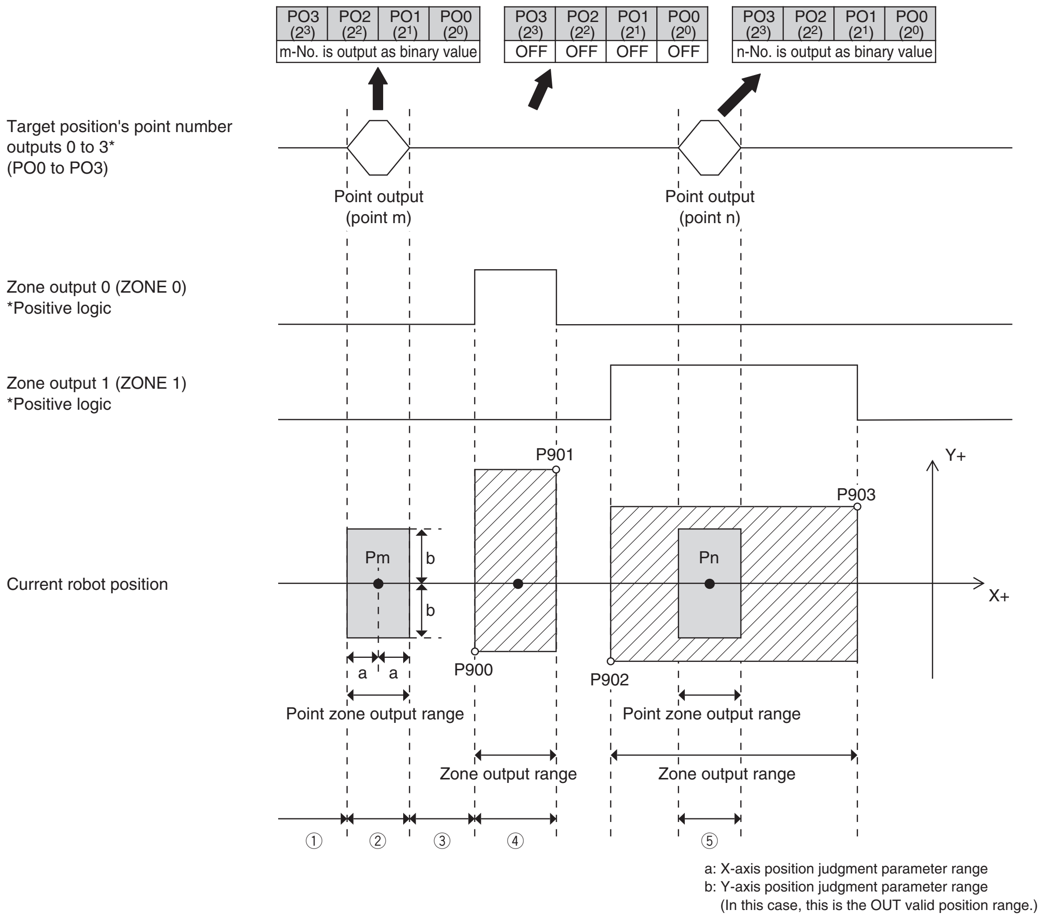
(9) When the DRCX received the point movement command and the BUSY signal turned on, the previous target position's point number being output from the specified point number (PO0 to PO5) is cleared and the current target position's point number is then output.

CAUTION

- *If moving the robot to point 0 by specifying it with a point movement command that is first executed after turning on the controller, all of PO0 to PO5 still remain off (because P0 = 000000 (binary)) even after the robot has moved to point 0. This means that the PO0 to PO5 status does not change even after specifying P0 as the target position, so no information is available to indicate whether the movement command to P0 was received. This should be kept in mind when moving the robot to point 0.*
- *When specifying the axis, the DI13 and DI14 (pin No. B11 and A12) status must be checked beforehand. (Refer to "PRM10: Control axis selection with I/O command".)*

(3) Outputting the corresponding point number by the point zone output function

Zone outputs (ZONE 0, ZONE 1) are also explained here.



* The number of target point number outputs that can be used depends on I/O assignment type.

Precondition: 1) The following steps are explained assuming that PRM26=221.

When the PRM26 setting = 221	
I/O assignment type	Type 2 (Point No. output type)
Permissible number of movement points	16 points
Point output selection	Point zone output
Point zone judgment method (position judgment parameter)	OUT valid position

2) The Zone 0 and Zone 1 output signals are enabled and set to positive output by the Zone output selection parameter (PRM24).

(1) Target position's point number outputs PO0 to PO3 are off since the current robot position is not within the point zone output range. ZONE 0 and ZONE 1 output signals are also off since the robot does not yet enter the zone output range.

(2) Outputs the corresponding point number through PO0 to PO3 since the current robot position is within the point zone output range (Pm ± OUT valid position range). ZONE 0 and ZONE 1 output signals are still off since the robot does not yet enter the zone output range.

(3) As with (1), all the target position's point number outputs PO0 to PO3, ZONE 0 signal and ZONE 1 output signal are off.

(4) ZONE 0 output signal turns on since the current robot position is within the zone output range (P900 to P901). (ZONE 1 signal remains off since the robot is not within the zone output range of P902 to P903). At this point, the target position's point number outputs PO0 to PO3 are still off since the robot is not within the point zone output range.

(5) Outputs the corresponding point number through PO0 to PO3 since the current robot position is within the zone output range (P902 to P903) and also within the point output range (Pn ± OUT valid position range). At this point, ZONE 1 output signal turns on. (ZONE 0 output signal turns off since the robot is not within the zone output range of P900 to P901).

NOTE

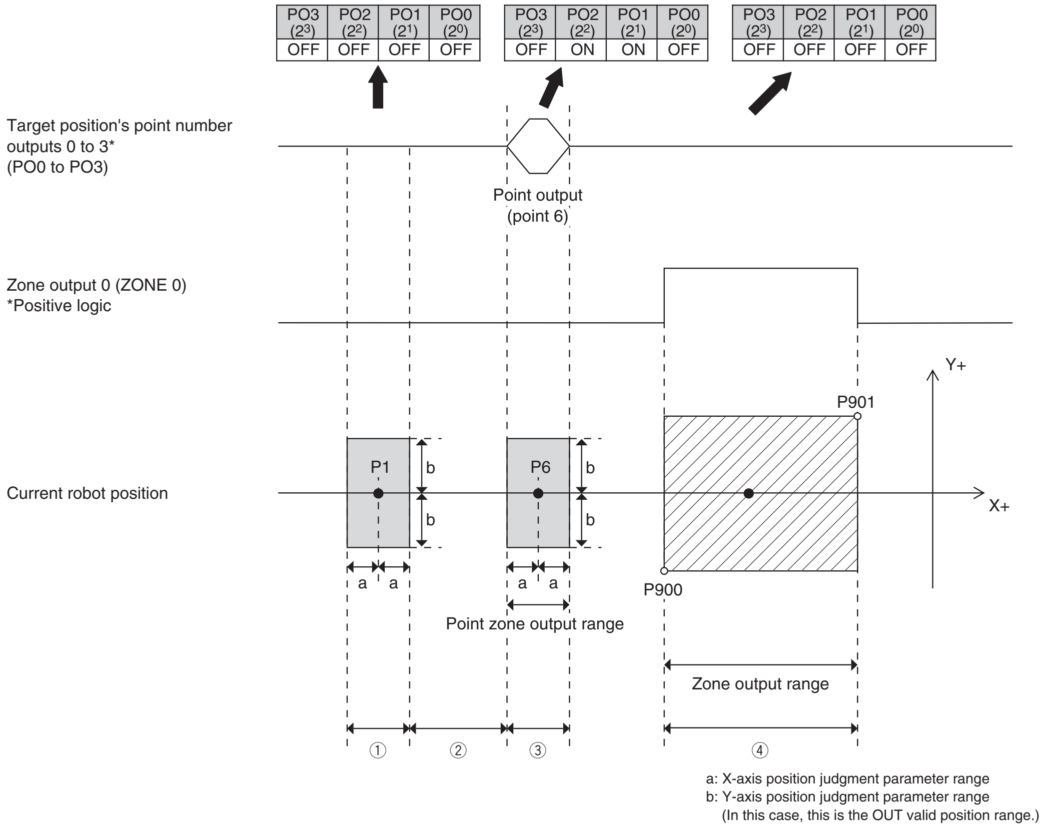
- *When using an optional unit such as a CC-Link, the corresponding point number for the point zone output function is output to both the corresponding parallel I/O (PO0 to PO5) and the serial I/O (PO200 to PO205).*
- *At controller Ver. 18.64 and later versions, the point zone judgment method can be selected (by the position judgment parameter) as either the "OUT valid position" or "Positioning-completed pulse" (this setting is specified by the PRM26 setting's "thousands" digit value).*
  In versions prior to Ver. 18.64, only the "OUT valid position" can be used as the point zone judgment method (specified by the position judgment parameter).
- *The "OUT valid position" can be changed by parameter setting (X-axis: PRM56, Y-axis: PRM96).*
- *The "Positioning-completed pulse" can be changed parameter setting (X-axis: PRM53, Y-axis: PRM93).*

CAUTION

- ***When the current robot position is within two or more point zone output ranges, the smaller or smallest point number is output.***
  Example: If the current robot position is within two point output ranges specified by P2 and P5, then P2 is output.
- ***If the current robot position is not within any point output range, all of PO0 to PO5 turn off.***
- ***A 10ms sampling time is needed for position monitoring, so the point zone output might not be detected when moving the robot at high speeds.***
- ***For point zone output and zone output, PO and ZONE0, ZONE1 are output only when the all axes of the robot are within the output range specified for each zone.***
- ***If outputting point 0 (P0) as the corresponding point for the point zone output function, all of PO0 to PO5 remain off (because P0 = 000000 (binary)). This means that the PO0 to PO5 status does not change even after the robot has entered the zone specified by P0. This should be kept in mind when monitoring P0.***

(4) Outputting the corresponding point number by the movement point zone output function

Zone outputs (ZONE 0) are also explained here.



* The number of target point number outputs that can be used depends on I/O assignment type.

Precondition: 1) The following steps are explained assuming that PRM26=321.

When the PRM26 setting = 321	
I/O assignment type	Type 2 (Point No. output type)
Permissible number of movement points	16 points
Point output selection	Movement point zone output
Point zone judgment method (position judgment parameter)	OUT valid position

2) The Zone 0 output signal is enabled and set to positive output by the Zone output selection parameter (PRM24).

3) Set the movement point as P6.

(1) Although the robot is within the P1 ± OUT valid position range (point zone output range), all the PO0 to PO3 target position point number outputs are off because P1 is not the movement point. Moreover, the ZONE 0 output is also off because the robot is not within the specified zone output range.

(2) All the PO0 to PO3 target position point number outputs are off because the robot is not within the point zone output range. Moreover, the ZONE 0 output is also off because the robot is not within the specified zone output range.

(3) The corresponding point number P6 is output to PO0 through PO3 (P1, P2 are on; P0, P3 are off) because the robot is within the P6 ± OUT valid position range (point zone output range), and because P6 is the movement point. ZONE 0 remains off at this time because the robot is not within the specified zone output range.

(4) The ZONE 0 output turns on because the robot is within the specified zone output range (P900 to P901). All the PO0 to PO3 target position point number outputs are off at this time because the robot is not within any point zone output range.

NOTE

- *The movement point zone output function is supported only in Ver. 18.64 and later versions.*
- *When using an optional unit such as CC-Link, the corresponding point number for the movement point zone output function is output to the corresponding parallel I/O (PO0 to PO5) and the serial I/O (PO200 to PO205).*
- *The movement point number specified just prior to movement START by point movement command (ABS-PT, INC-PT) is registered as the movement point.*
- *Because movement points are reset immediately after a controller power on, all PO outputs turn off. Movement points are also reset when the RESET command is executed, and movement point zone outputs by PO are cleared.*
- *The point zone judgment method can be selected as either the "OUT valid position" or "Positioning-completed pulse" (this setting is specified by the PRM26 setting's "thousands" digit value).*
- *The "OUT valid position" can be changed by parameter setting (X-axis: PRM56, Y-axis: PRM96).*
- *The "Positioning-completed pulse" can be changed parameter setting (X-axis: PRM53, Y-axis: PRM93).*

CAUTION

- ***All the PO0 to PO5 outputs are off when the robot is not within the point zone output range.***
- ***A 10ms sampling time is needed for position monitoring, so the point zone output may not be detected during high-speed robot motion.***
- ***For movement point zone output and zone output, PO and ZONE 0, ZONE 1 are output only when all robot axes are within the specified zone output ranges.***
- ***When outputting point 0 (P0) as the corresponding point for the movement point zone output function, all the PO0 to PO5 outputs remain off (because P0 = 000000 (binary)). Therefore, the PO0 to PO5 statuses do not change even after the robot has entered the zone specified by P0. This should be kept in mind when monitoring P0.***

Chapter 4 BASIC OPERATION OF THE TPB

The TPB is a hand-held, pendant-type programming box that connects to the DRCX controller to edit or run programs for robot operation.
The TPB allows interactive user operation on the display screen so that even first-time users can easily operate the robot with the TPB. This chapter describes the basic operation of the TPB.

4-1 Connecting and Disconnecting the TPB

4-1-1 Connecting the TPB to the DRCX controller

⚠ *CAUTION*

Do not modify the TPB cable or use any type of relay unit for connecting the TPB to the DRCX controller. Doing so might cause communication errors or malfunctions.

■ When the power supply to the controller is turned off

Connect the TPB connector to the connector labelled "TPB" on the front panel of the controller and supply power to the controller. A beep sounds for approximately 1 second and then the screen shown at the right appears. This screen is referred to as the "Initial screen" from this point onwards.

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

■ When the power supply to the controller is turned on

The TPB can also be connected to the DRCX controller if the power supply to the controller is on. In this case, hold down the ESC switch on the front panel of the controller as you plug in the TPB connector. If the TPB is connected to the controller without pressing the ESC switch, emergency stop might be triggered causing the robot servo to turn off. Also, if the TPB is connected while the controller is executing a program or an I/O dedicated command, then the execution will be interrupted regardless of whether or not the ESC switch is held down.

⚠ *CAUTION*

Any of the messages "08: PNT DATA DESTROY", "09: PRM DATA DESTROY" or "10: PGM DATA DESTROY" may appear on the TPB when the power to the controller is turned on. (See "13-2 Alarm and Countermeasures".) If one of these messages appears, turn off the power to the controller and then turn it back on again while the emergency stop button of the TPB is still depressed. In this state, the robot servo remains off, but the initial screen appears on the TPB to allow key operation, so initialize and restore the data.

If the message "05: BATT. LOW-VOLTAGE" appears on the TPB when the power is turned on, turn off the power to the controller and then turn it on again while the emergency stop button of the TPB is still depressed. In this state, the robot servo remains off, but the initial screen appears on the TPB to allow key operation, so make a backup of the data, and then replace the lithium battery in the controller (the lithium battery normally lasts five years). (See "14-2 Replacing the System Backup Battery".)

If the message "SIO error" is displayed on the TPB, check whether the I/O dedicated command input is on. If the dedicated command input is on, the TPB cannot be used, so the dedicated input must always be a pulse input (the dedicated command input must be off when the BUSY signal turns on.) (Refer to "3-2-1 Dedicated command input".)

4-1-2 Disconnecting the TPB from the DRCX controller

To disconnect the TPB from the controller while a program or an I/O dedicated command is being executed, pull out the TPB while holding down the ESC switch on the front panel of the controller. Failing to hold down the ESC switch will trigger emergency stop in the controller and turn off the servo.
When the TPB will be left disconnected from the controller for a long period of time, we recommend attaching the RS-232C connector dust cover (supplied) to the TPB connector on the controller.

4-2 Basic Key Operation

1) Selectable menu items are displayed on the 4th line (bottom line) of the TPB screen.

   Example A is the initial screen that allows you to select the following modes.

   1EDIT
   2OPRT
   3SYS
   4MON

   The number to the left of each mode corresponds to the function keys from F1 to F4.

2) On the initial screen shown in A, pressing a function key moves to a lower level in the menu hierarchy. (A→B→C→D)

   To return to the previous screen or menu level, press the ESC key. (See "4-4 Hierarchical Menu Structure" in this chapter.)

3) If an error occurs during operation, a buzzer sounds for approximately 1 second and an error message like that shown in Example E appears on the 3rd line of the screen. If this happens, check the contents of the error message and then press the ESC key. The error message will be cleared to allow continuing operation. To correct the error, refer to the message tables in Chapter 12.

4) If an alarm occurs during operation, its alarm message appears on the 3rd line of the screen and a buzzer keeps sounding. The TPB cannot be used in this state. Turn off the power to the controller and then correct the problem by referring to "13-2 Alarm and Countermeasures".

A

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

↓ F2 ↑ ESC

B

```
[OPRT]
select menu

1ORG 2STEP3AUTO
```

↓ F2 ↑ ESC

C

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

↓ F3 ↑ ESC

D

```
[OPRT-STEP] 100 0: 0
PGM  No   = _
(program No)  0→99
```

E

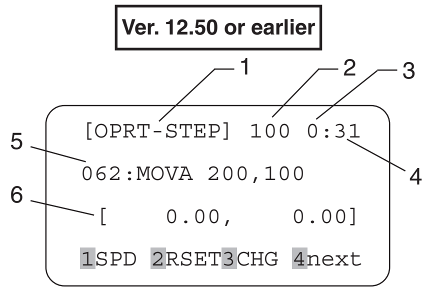
```
[OPRT-STEP]

32:origin incomplete
```

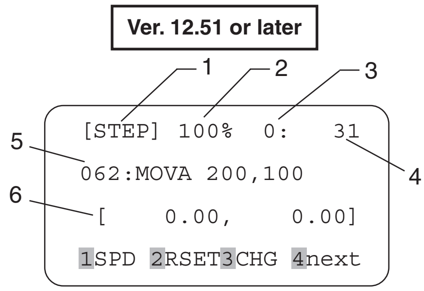
4-3 Reading the Screen

The following explains the basic screen displays and what they mean.

4-3-1 Program execution screen

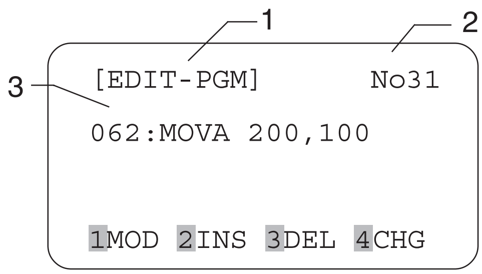
The display method slightly differs depending on the version of TPB.





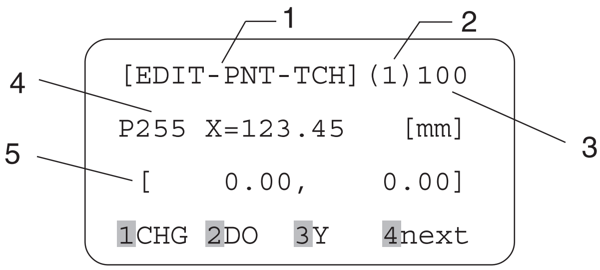
1. Current mode
2. Execution speed
3. No. of task being executed
4. No. of program being executed
   * On TPB version 12.51 or later, when switched from the lead program to another program, this area shows the program numbers as the "currently executed program / lead program".
5. No. of step being executed
6. Current position (left : X-axis, right : Y-axis)

4-3-2 Program edit screen



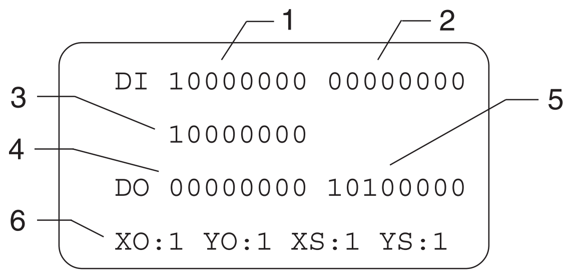
1. Current mode
2. No. of program being edited
3. No. of step being edited

4-3-3 Point edit screen (teaching playback)



1. Current mode
2. Speed selection number
3. Speed parameter (%)
4. Edit point number
5. Current position (left : X-axis, right : Y-axis)

4-3-4 DIO monitor screen



1. General-purpose input
   From left
   DI15 to DI8

2. General-purpose input
   From left
   DI7 to DI0

3. Dedicated input
   From left
   Interlock (LOCK)
   0: Locked state (robot movement not possible)
   1: Unlocked state (robot movement possible)
   Return-to-origin command (ORG-S)
   Reset command (RESET)
   Automatic operation start command (AUTO-R)
   Step operation start command (STEP-R)
   Absolute point movement command (ABS-PT)
   Relative point movement command (INC-PT)
   Servo recovery command (SERVO)

4. General-purpose output
   From left
   DO12 to DO5

5. Dedicated and general-purpose outputs
   From left
   READY, BUSY, END, DO4 to DO0

6. Origin sensor status and servo status
   From left
   XO: X-axis origin sensor status
   YO: Y-axis origin sensor status
   0: Off (Closed)
   1: On (Open)
   XS: X-axis servo status
   YS: Y-axis servo status
   0: Servo off
   1: Servo on

4-4 Hierarchical Menu Structure

- POWER ON
  - INFORMATION (System information)
  - EDIT (Editing)
    - PGM (Program Edit)
      - MOD (Step Edit)
      - INS (Step Insert)
      - DEL (Step Delete)
      - CHG (Program Change)
    - PNT (Point Edit)
      - MDI (Manual Data Input)
        - CHG (Point Change)
        - PLT (Pallet Number Change)
      - TCH (Teaching Playback)
        - CHG (Point Change)
        - DO (General-purpose Output Control)
        - Y (Y-axis)/X (X-axis)
        - SPD (Speed Change)
        - S_SET (Speed Set)
        - TRC (Point trace)
        - PLT (Pallet Number Change)
      - DTCH (Direct Teaching)
        - CHG (Point Change)
        - DO (General-purpose Output Control)
        - Y (Y-axis)/X (X-axis)
        - BRK (Brake)
        - PLT (Pallet Number Change)
      - DEL (Delete)
    - UTL (Utility)
      - COPY (Program Copy)
      - DEL (Program Delete)
      - LIST (Program List)
  - OPRT (Operation)
    - ORG (Origin Return)
      - ALL (All axes)
      - X (X-axis)
      - Y (Y-axis)
    - STEP (Step Run)
      - SPD (Execution Speed Change)
      - RSET (Program Reset)
      - CHG (Program Change)
      - VAL (Variable Monitor)
      - S_ON (Servo ON)
      - CHGT (Task Change)
      - MIO (Memory IO Monitor)
    - AUTO (Auto Run)
      - SPD (Execution Speed Change)
      - RSET (Program Reset)
      - CHG (Program Change)
      - VAL (Variable Monitor)
      - S_ON (Servo ON)
      - CHGT (Task Change)
      - MIO (Memory IO Monitor)
  - SYS (System)
    - PRM (Parameter Setting)
      - COM (Common Parameter)
      - X (X-axis Parameter)
      - Y (Y-axis Parameter)
    - B.UP (Backup)
      - SAVE (Save)
        - normal (Standard)
        - compati (Compatible)
      - LOAD (Load)
        - PGM (Program)
        - PNT (Point)
        - PRM (Parameter)
        - ALL (All Data)
      - FMT (Format)
        - normal (Standard)
        - compati (Compatible)
      - ID (Control No.)
    - INIT (Initialization)
      - PGM (Program)
      - PNT (Point)
      - PRM (Parameter)
        - XY (2 axes)
        - FLIP (Two single axis robots)
      - ALL (All Data)
        - XY (2 axes)
        - FLIP (Two single axis robots)
    - SAFE (Safety Setting)
      - ACLV (Access Level)
        - EDIT (Editing)
        - OPRT (Operation)
        - SYS (System)
        - CARD (Memory Card)
      - SVCE (SERVICE mode)
        - SET (Enable/Disable Setting)
        - DEV (Limitation to Operating Device)
        - SPD (Speed Limitation)
        - RUN (Step Run/Auto Run Limitation)
        - HtoR (Hold-to-Run Setting)
    - OPT (Option)
    - UTL (Utility)
      - HDPR (Hidden parameter display)
      - REC (Record)
        - ALM (Alarm)
        - ERR (Error)
  - MON (Monitor)
    - DIO (DIO Monitor)
    - DUTY (DUTY Monitor)
      - RUN (Monitor Start)
      - STOP (Monitor Stop)
      - RSLT (Result Display)

The menu hierarchy might slightly differ depending on the versions of the controller and TPB.

4-5 Restricting Key Operation by Access Level

The TPB key operations can be limited by setting the access levels (operation levels).
A person not trained in robot operation might accidentally damage the robot system or endanger others by using the TPB incorrectly. Set the access levels to restrict TPB key operations and prevent such accidents.

NOTE
The access level settings are protected by a password so that changes cannot be instantly made.

4-5-1 Explanation of access level

The access levels can be set individually for editing, operation, system and memory card.
The details of the key operations limited at each level are explained below.

Editing

Level	Description
0	All operations are permitted.
1	Program editing is prohibited. (Program data can be checked.)
2	In addition to Level 1, point data editing, manual release of brake and point trace (movement to registered data point) are prohibited. (The [X/Z -] and [X/Z +] keys can be used to move the robot and general-purpose outputs can be controlled.)
3	Any operation in EDIT mode is prohibited. (Cannot enter EDIT mode.)

Operation

Level	Description
0	All operations are permitted.
1	Changing the execution speed and program is prohibited.
2	In addition to Level 1, automatic operation, step operation and program reset are prohibited. (Return-to-origin can be performed and variables can be monitored.)
3	Any operation in OPRT mode is prohibited. (Cannot enter OPRT mode.)

System-related data

Level	Description
0	All operations are permitted.
1	Initialization is prohibited.
2	In addition to Level 1, changing the parameters and setting the option units are prohibited. (Parameter data and option unit settings can be checked.)
3	Parameter editing, initialization and option setting are prohibited. (Cannot enter SYS-PRM, SYS-INIT and SYS-OPT modes.)

Memory card

Level	Description
0	All operations are permitted.
1	Loading the parameters and all data to the DRCX is prohibited. (Point data or program data can be loaded.)
2	Loading any data to the DRCX is prohibited. (Data can be saved and the memory card formatted.)
3	Use of memory card is prohibited. (Cannot enter SYS-B.UP mode.)

4-5-2 Changing an access level

1) Press F3 (SYS) on the initial screen.

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Press F4 (next) to switch the menu display and then press F1 (SAFE).

```
[SYS]
select menu

1SAFE2OPT 3UTL 4next
```

3) When the password entry screen appears, enter the password and press ➡.

```
[SYS-SAFE]
 Password: 18.11_
input password
```

4) When the password is accepted, the screen shown on the right appears.

Press F1 (ACLV) here.

```
[SYS-SAFE]
select menu

1ACLV2SVCE
```

5) Select the item you want to change.

   To change the access level for editing, press F1 (EDIT).

   To change the access level for operation, press F2 (OPRT).

   To change the access level for system-related data, press F3 (SYS).

   To change the access level for memory card, press F4 (CARD).

```
[SYS-SAFE-ACLV]
select menu

1EDIT2OPRT3SYS 4CARD
```

6) The currently set access level appears.

   To change this setting, use the number key to enter the access level and then press ➡.

```
[SYS-SAFE-ACLV-EDIT]
access level : 0
all access OK
```

7) When the access level has been changed, the memory write screen appears.

   To save the change permanently (retain the change even after the controller power is turned off), press F1 (SAVE).

   To save the change temporarily (retain the change until the power is turned off), press F2 (CHG).

   To cancel changing of the setting, press F3 (CANCEL).

```
[SYS-SAFE-ACLV-EDIT]
access level : 1
change PGM invalid
1SAVE2CHG 3CANCEL
```

8) When writing is complete, the screen returns to step 6.

```
[SYS-SAFE-ACLV-EDIT]
access level : 1
change PGM invalid
```

NOTE

The password is identical to the DRCX controller's version number. For example, if the controller version is 18.11, enter 18.11 as the password. Once the password is accepted, it will not be requested unless the TPB is disconnected from the controller or the controller power is turned off.

NOTE

To avoid access level conflict between operation and others, the access levels may be automatically adjusted. For example, if the access levels related to editing, system and memory card are "0", they are automatically changed to "1" when the operation-related access level is "1" or "2" or "3". The access levels remain unchanged if they are "1" or "2" or "3".

Chapter 5 PARAMETERS

The DRCX controller uses a software servo system, so no adjustment of hardware components such as potentiometers or DIP switches are required. Instead, the DRCX controller uses parameters that can be easily set or changed by the TPB or PC (personal computer).
This chapter contains a detailed description of each of the parameters, and explains how to use the TPB to change and specify parameter settings.

SAFETY

Errors such as motor overload are detected by the software, so the controller parameters must be set correctly to match the connected robot model. The parameters are initialized to match the robot model when the robot is shipped, so confirm them before starting use. If there is any trouble, please contact our sales office or sales representative.

5-1 Setting the Parameters

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (PRM).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

3) When editing the common parameters, press F1 (COM).

```
[SYS-PRM]
select menu

1COM 2X   3Y
```

4) The current PRM0 (number of conditional input points) setting appears on the screen. Use the STEP UP and STEP DOWN keys to scroll the parameters until you find the parameter you want to set.

```
[SYS-PRM-COM]
PRM0 = 4
JMPF DI number
range 1 to 8
```

5) When the desired parameter is displayed, enter the new value with the number keys and then press ➡.

```
[SYS-PRM-COM]
PRM1 = 0
alarm output
0:OFF 1:ON  2:ON(DO6)
```

6) When editing the X-axis parameters, press F2 (X) in step 3. Likewise, when editing the Y-axis parameters, press F3 (Y) in step 3.

```
[SYS-PRM-X]
PRM48 = 450   [mm]
(+)soft limit
range -9999→9999
```

5-2 Parameter Description

Three types of parameters are described below: common parameters, X-axis parameters and Y-axis parameters.

CAUTION

Parameters not displayed on the TPB screen are automatically set or optimized to match the robot type when the robot parameters are initialized. You usually do not have to change these parameter settings.
If for some special reason you need to change or check these hidden parameters, use any of the following methods.

- ***Turn on the power to the controller while holding down the ESC key on the TPB.***
- ***Connect the TPB to the controller while holding down the ESC key on the TPB.***
- ***Use the system utility mode that allows you to display hidden parameters. (See "10-5-1 Viewing hidden parameters".)***

Take extra caution when handling hidden parameters.

5-2-1 Common parameters

PRM0: No. of conditional input points

This parameter specifies the number of effective points for the third data conditional input for executing the JMPF statement of the robot language.
For example, when the default setting is selected for this parameter, the four points from DI0 to DI3 are used as the conditional inputs for the JMPF statement.

Input range: 1 to 8 (points)
Default value: 4

No. of conditional input points versus general-purpose input and setting range

No. of conditional input points	General-purpose input	Setting range
1	DI0	0 to 1
2	DI0 to DI1	0 to 3
3	DI0 to DI2	0 to 7
4	DI0 to DI3	0 to 15
5	DI0 to DI4	0 to 31
6	DI0 to DI5	0 to 63
7	DI0 to DI6	0 to 127
8	DI0 to DI7	0 to 255

PRM1: Alarm number output

When an alarm is issued, this parameter selects whether the alarm number is to be output as a general-purpose output. When this parameter is set to 1, the alarm number is output as a 5-bit binary signal through DO0 to DO4.
When this parameter is set to 2, the alarm number is output as a 5-bit binary signal through DO0 to DO4 and alarm axis information is output through DO6.

Input range: 0 to 2
Meaning: 0: No output
1: Output (no axis output)
2: Output (axis output)
Default value: 0

Example of alarm Number - DO output

Alarm No.	Alarm Message	DO4	DO3	DO2	DO1	DO0
01	OVER LOAD	OFF	OFF	OFF	OFF	ON
02	OVER CURRENT	OFF	OFF	OFF	ON	OFF
03	OVER HEAT	OFF	OFF	OFF	ON	ON
04	POWER DOWN	OFF	OFF	ON	OFF	OFF
:	:	:	:	:	:	:
:	:	:	:	:	:	:
16	ABNORMAL VOLTAGE	ON	OFF	OFF	OFF	OFF
17	SYSTEM FAULT 2	ON	OFF	OFF	OFF	ON
18	FEEDBACK ERROR 3	ON	OFF	OFF	ON	OFF
19	SYSTEM FAULT 3	ON	OFF	OFF	ON	ON
:	:	:	:	:	:	:
:	:	:	:	:	:	:

Alarm axis-DO output

(Effective only when PRM1=2)

Alarm axis	DO6
X-axis	OFF
Y-axis	ON

For more details on the alarm No. and contents, refer to "13-2-2 Alarm message list".

PRM2: Operation at return-to-origin complete

Selects the operation to be executed simultaneously with completion of return-to-origin. A signal can be output as a general-purpose output indicating that return-to-origin has been completed or to reset the program.

Input range: 0 to 3
Meaning: 0: Nothing is executed
1: DO5 is turned on
2: Program reset is executed
3: DO5 turns on after program reset
Default value: 2

* When this parameter is set to 1 or 3, DO5 is not affected by program reset (in other words, DO5 does not turn off even when the program is reset). If you want to turn off DO5 after return-to-origin is complete, use the program command to execute DO 5,0 or manually operate the general-purpose output by using the TPB.
(See "7-4 Manual Control of General-Purpose Output".)
* DO5 turns on when all axes have returned to their origins.

PRM3: Return-to-origin order

This parameter determines the order of the axes when returning to origin.

Input range: 0 to 2
Meaning: 0: X →Y
1: Y →X
2: XY simultaneous
Default: Depends on robot type.

⚠ *CAUTION*

When the stroke end detection method (PRM55=1, PRM95=1) is selected for the 2 axes as the origin detection method, do not set this parameter to 2 (XY simultaneous). Otherwise, return-to-origin may not be performed normally.

PRM4: MOVF speed

This sets the speed at which the robot moves when the program language MOVF statement is executed.

Input range: 1 to 10000 (mm/sec)
Default value: 10

PRM5: I/O point movement command speed 1

This parameter specifies the movement speed (%) at which the robot moves when a point movement command (ABS-PT, INC-PT) is executed.
When "type 3" (point teaching type) is selected by the I/O assignment setting, this parameter specifies the jog speed at which the robot moves at a jog movement command (JOG+, JOG-).
This movement speed specified here is the speed used with DI10 turned ON and DI11 turned OFF.
When "type 2" (point number output type) or "type 3" (point teaching type) is selected by the I/O assignment setting and the speed is changeable by SPD1 and SPD2, this parameter specifies the speed when SPD1 is set to ON and SPD2 is set to OFF.

Input range: 1 to 100 (%)
Default value: 10

* The actual speed at which the robot moves with a point movement command (ABS-PT, INC-PT) is the speed obtained by multiplying the execution speed displayed in AUTO or STEP mode by this parameter. (Refer to "4-3-1 Program execution screen".)
For example, if the execution speed displayed in AUTO or STEP mode is 50 and this parameter is set to 10, then the actual speed will be:
3000 rpm × (50/100) × (10/100) = 150 rpm
(When PRM64 and PRM104 are set to 3000.)

* If this parameter is set to 10, then the jog speed will be:
100 × 10/100 = 10mm/sec.

PRM6: I/O point movement command speed 2

This parameter specifies the movement speed (%) at which the robot moves when a point movement command (ABS-PT, INC-PT) is executed.
When "type 3" (point teaching type) is selected by the I/O assignment setting, this parameter specifies the jog speed at which the robot moves at a jog movement command (JOG+, JOG-).
This movement speed specified here is the speed used with DI10 turned OFF and DI11 turned ON.
When "type 2" (point number output type) or "type 3" (point teaching type) is selected by the I/O assignment setting and the speed is changeable by SPD1 and SPD2, this parameter specifies the speed when SPD1 is set to OFF and SPD2 is set to ON.

Input range: 1 to 100 (%)
Default value: 30

* The actual speed at which the robot moves with a point movement command (ABS-PT, INC-PT) is the speed obtained by multiplying the execution speed displayed in AUTO or STEP mode by this parameter. (Refer to "4-3-1 Program execution screen".)
For example, if the execution speed displayed in AUTO or STEP mode is 50 and this parameter is set to 30, then the actual speed will be:
3000 rpm × (50/100) × (30/100) = 450rpm
(When PRM64 and PRM104 are set to 3000.)
* If this parameter is set to 30, then the jog speed will be:
100 × 30/100 = 30mm/sec.

PRM7: I/O point movement command speed 3

This parameter specifies the movement speed (%) at which the robot moves when a point movement command (ABS-PT, INC-PT) is executed.
When "type 3" (point teaching type) is selected by the I/O assignment setting, this parameter specifies the jog speed at which the robot moves at a jog movement command (JOG+, JOG-).
This movement speed specified here is the speed used with DI10 and DI11 turned ON.
When "type 2" (point number output type) or "type 3" (point teaching type) is selected by the I/O assignment setting and the speed is changeable by SPD1 and SPD2, this parameter specifies the speed when both SPD1 and SPD2 are set to ON.

Input range: 1 to 100 (%)
Default value: 70

* The actual speed at which the robot moves with a point movement command (ABS-PT, INC-PT) is the speed obtained by multiplying the execution speed displayed in AUTO or STEP mode by this parameter. (Refer to "4-3-1 Program execution screen".)
For example, if the execution speed displayed in AUTO or STEP mode is 50 and this parameter is set to 70, then the actual speed will be:
3000 rpm × (50/100) × (70/100) = 1050rpm
(When PRM64 and PRM104 are set to 3000.)
* If this parameter is set to 70, then the jog speed will be:
100 × 70/100 = 70mm/sec.

PRM8: English/Japanese selection

This parameter sets the language for the response messages displayed on the TPB or handled by RS-232C communications.

Input range: 0 or 1
Meaning: 0: English
1: Japanese
Default value: 0

PRM9: Pre-operation action selection

This parameter checks whether return-to-origin has been performed or resets the program before running automatic operation or step operation.
When set to 0 or 2, an error (return-to-origin incomplete) is issued if return-to-origin has not been performed and automatic operation and step operation are not accepted.
When set to 1 or 3, the program runs even when return-to-origin has not been performed. However, an error (return-to-origin incomplete) is issued when a movement command (MOVA, etc.) is executed if return-to-origin is still incomplete. To avoid this, perform return-to-origin in advance or insert the ORGN command into the program.

Input range: 0 to 3
Meaning: 0: Checks whether return-to-origin has been performed.
1: Nothing is executed.
2: Resets the program after checking return-to-origin.
3: Resets the program.
Default value: 1

* When set to 2 or 3, the program is reset only during automatic operation. (The program is not reset during step operation.)

PRM10: Control axis selection with I/O command

This parameter determines whether to select an axis when a dedicated I/O command (ABS-PT, INC-PT, ORG-S, SERVO) is executed. If Type 3 (Point teaching type) is selected by the I/O assignment change function, this parameter also determines whether to select an axis when a jog movement command (JOG+, JOG-) is executed.
When this parameter is set to "0" (invalid) for executing a dedicated I/O command (ABS-PT, INC-PT, ORG-S, SERVO), all axes are always controlled so that no specific axis can be selected. When set to "1" (Valid), the desired axis can be selected with DI13 and DI14 (pin No. B11 and A12).
When this parameter is set to "0" (invalid) for executing a jog movement command (JOPG+, JOG-), the X-axis is always selected. When set to "1" (Valid), the desired axis can be selected with DI13 and DI14 (pin No. B11 and A12), and the Y-axis can therefore be selected.

Input range: 0 or 1
Meaning: 0: Invalid
1: Valid
Default value: 0

Axis selection example

DI14 (Pin No.A12)	DI13 (Pin No.B11)	Selected axis	
		ABS-PT、INC-PT、ORG-S、SERVO	JOG+、JOG-
OFF	OFF	All axes	X-axis
OFF	ON	X-axis	X-axis
ON	OFF	Y-axis	Y-axis
ON	ON	All axes	X-axis

PRM11: Not used

Default value: 1

PRM12: Teaching movement data

This parameter is used to move the robot with a communication command @X+, @XINC, @Y+ or @YINC.

Input range: 1 to 100 (%)
Default value: 100

PRM13: Teaching movement data 1 (for TPB)

This parameter is used only by the TPB and is unavailable to users.

Input range: 1 to 100 (%)
Default value: 100

* The TPB writes the contents of PRM13 into PRM12 when connected to the controller.

PRM14: Teaching movement data 2 (for TPB)

This parameter is used only by the TPB and is unavailable to users.

Input range: 1 to 100 (%)
Default value: 50

PRM15: Teaching movement data 3 (for TPB)

This parameter is used only by the TPB and is unavailable to users.

Input range: 1 to 100 (%)
Default value: 10

PRM16: Teaching count data (written in TPB)

This parameter is used only by the TPB and is unavailable to users.

Default value: 1

PRM17: Maximum program speed

The speed data defined in MOVA, MOVI and MOVM statements in a program is multiplied by this parameter value to determine the maximum speed at which the robot actually moves. This parameter is used to lower the speed of the overall program. When the TPB is used, any speed changes in the AUTO and STEP modes will also change this parameter value.
Max. speed (%) = PRM17 × speed operand (%) of movement command / 100

Input range: 1 to 100 (%)
Default value: 100

PRM18: Robot type number

This parameter shows the robot number currently used. (See "15-1-2 Robot number list".) This is a read-only parameter. When changing the robot number or if the memory contents are corrupted, perform parameter initialization. (See "10-1 Initialization".)

* When the robot setting is made for a Multi-Flip, this parameter value is set to 0 and the robot type number for each axis is shown by PRM72 and PRM112.

PRM19: Controller version 1

This parameter reads out the version information (1) on the control software in the controller.
This is a read-only parameter.

PRM20: System mode selection

This parameter specifies the system operation mode. When you want to use the DRCX controller in operating specifications that differ from normal mode, for example, to make it compatible with the conventional controllers, change this parameter as explained below. This parameter functions are allocated in bit units.

Input range: 0 to 255
Default value: 16

Function allocation in bit units

Bit	Function	Selected operating mode	Setting	Addition value
0	General-purpose input definition for using an I/O point movement command	Normal mode (DI0 to DI11)	0	0
		Conventional compatible mode (DI0 to DI9)	1	1
1	READY signal sequence setting	DRCA compatible output mode	0	0
		SRCA compatible output mode	1	2
2	END signal sequence setting when the controller has started normally	Normal mode (to be output)	0	0
		Conventional compatible mode (not to be output)	1	4
3	Voltage check setting for system backup battery	Check	0	0
		No check	1	8
4	Absolute backup function setting	Disable	0	0
		Enable	1	16
5 to 6	Reserved for system use		0	0
7	END output sequence setting at command execution completion	ON at normal command completion	0	0
		ON at command signal OFF at normal command completion	1	128
8 to 15	Reserved for system use		0	0

Example: For conventional compatibility with I/O point movement command general-purpose inputs, and the END sequence at normal controller startups, PRM20 should be set to "20" because of 0000000000010101(binary)=21(decimal)

Bit	15 to 8	7	6 to 5	4	3	2	1	0	PRM20
Setting	0	0	0	1	0	1	0	1	
Addition value	0	0	0	16	0	4	0	1	16+4+1=21

Bit 0: General-purpose input definition for using an I/O point movement command

This selects a general-purpose input used for an I/O point movement command (ABS-PT, INC-PT).
In normal mode, use DI0 to DI9 to specify the point number and DI10 to DI11 to select the speed. All points (P0 to P999) can be specified with a movement command.
In conventional model compatible mode, use DI0 to DI7 to specify the point number, and DI8 to DI9 to select the speed. Points P0 to P254 can be specified with a movement command but points P255 to P999 cannot be selected.

5 PARAMETERS

Bit 1: READY signal sequence setting

This selects whether to set the READY signal sequence compatible with the DRCA or SRCA controller.
In DRCA compatible mode, the READY signal turns on at the instant that emergency stop is released. In the SRCA compatible mode the READY signal turns on when the servo is turned on. (The READY signal will not turn on just by releasing emergency stop.)

Bit 2: END signal sequence setting when the controller has started normally

This selects whether to turn on the END signal when the controller has started normally.
In normal mode, the END signal turns on when the controller has started normally.
In conventional compatible mode, the END signal remains off even when the controller has started normally.

Bit 3: Voltage check setting for system backup battery

This selects whether to check the system backup battery voltage when the controller servo is turned on.
In such cases where you want to operate the robot immediately even when the battery needs to be replaced, you can temporarily disable this voltage check. (System backup batteries are located inside the controller and used to back up the parameters and point data.)

Bit 4: Absolute backup function setting

This selects whether to enable or disable the absolute backup function.
Normally, this is set to "enable" and a battery for absolute backup is required. If set to "disable", the controller can be operated without using an absolute backup battery.
When set to "enable", the robot position is maintained even after the power is turned off. When set to "disable", however, the origin position will be incomplete each time the power is turned off.

Bit 7: END output sequence setting at command execution completion (supported by Ver. 18.74 and later versions):

This selects the END output sequence at dedicated command completion.
With the standard setting ("0"), the command's execution result is output to the END output when the command is completed. When set to "1", the command's execution result is output to the END output when the command is completed, but only after the command signal turns off.

PRM21: Servo status output

This parameter selects whether to output the axis servo status as a general-purpose output. When this parameter is set to 1, DO7 turns on when all axes are servo-on and turns off when one axis is servo-off.

Input range:	0 or 1
Meaning:	0: Does not output the servo status.
	1: Outputs the servo status.
Default value:	0

* When this parameter is set to 1, DO7 is not affected by program reset (in other words, DO7 does not turn off even when the program is reset).

PRM22: Communication parameter setting

This sets communication parameters used for data transmission through RS-232C. For more details, see "11-1 Communication Parameter Specifications".

Default value: 0

PRM23: Lead program number (Available with Ver. 18.50 or later)

This parameter sets the lead program number.

Default value: 0

NOTE

The lead program is the program that has been selected as the execution program by the TPB or POPCOM. (See "9-4 Switching the Execution Program".)
The lead program can also be selected by executing a communication command "@SWI". It may also be switched when the program data is loaded into the controller from the memory card.

PRM24: Zone output selection (Available with Ver. 18.50 or later)

This parameter is used to select the output destination and output logic when the zone output function is enabled. The zone output is used to control the signal output when the robot's current position is within the specified range.
A maximum of 4 zone outputs are available by setting for PRM24. The output logic can also be changed.
This parameter functions are allocated in bit units.

Input range: 0 to 255
Default value: 0

Function allocation in bit units

Bit	Function	Selected value	Addition value
0	Zone 0 output enable setting	0: Disabled	0
		1: Enabled	1
1	Zone 1 output enable setting	0: Disabled	0
		1: Enabled	2
2	Zone 2 output enable setting	0: Disabled	0
		1: Enabled	4
3	Zone 3 output enable setting	0: Disabled	0
		1: Enabled	8
4	Zone 0 output logic setting	0: Positive logic	0
		1: Negative logic	16
5	Zone 1 output logic setting	0: Positive logic	0
		1: Negative logic	32
6	Zone 2 output logic setting	0: Positive logic	0
		1: Negative logic	64
7	Zone 3 output logic setting	0: Positive logic	0
		1: Negative logic	128
8 to 15	Reserved for system use		0

Example: To set zone 1 output to positive logic and zone 2 output to negative logic while enabling zone 1 output and zone 2 output, make the following settings.
PRM24 should be set to "70" because of 0000000001000110 (binary)=70 (decimal).

Bit	15 to 8	7	6	5	4	3	2	1	0	PRM24
Selected value	0	0	1	0	0	0	1	1	0	
Addition value	0	0	64	0	0	0	4	2	0	64+4+2=70

Zone output function

To use the zone output function, the desired zone must be specified with point data. (See Chapter 7, "EDITING POINT DATA".) When the robot enters the specified zone, its result is output to the specified port. Point numbers and output port that can be used for each zone output are listed below.

Zone setting range and output port

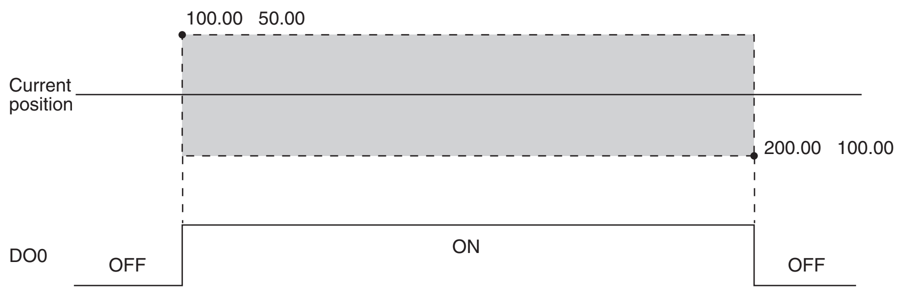
Zone No.	Specified range	Output port
Zone 0	P900-P901	DO0
Zone 1	P902-P903	DO1
Zone 2	P904-P905	DO2
Zone 3	P906-P907	DO3

CAUTION

The zone output function does not work if one item of the point data is unspecified or return-to-origin is incomplete.

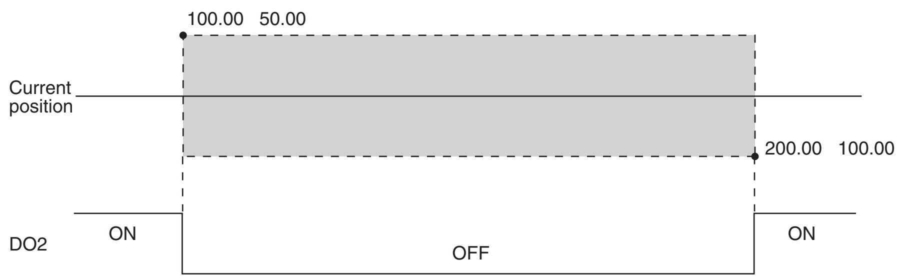
Example ①

PRM24=1 (Zone 0 output enabled, positive logic output)	
P900=100.00 50.00	P901=200.00 100.00



Example ②

PRM24=68 (Zone 2 output enabled, negative logic output)	
P900=100.00 50.00	P905=200.00 100.00



PRM25: Controller version 2 (Available with Ver. 18.50 or later)

This parameter reads out the version information (2) on the control software in the controller.
This is a read-only parameter.

PRM26: I/O assignment selection (Available with Ver. 18.57 or later)

This parameter selects the function to be assigned to each I/O signal.
This parameter setting allows changing the function assigned to each I/O signal. This makes it possible to output the destination point number and perform jog movement.
After changing the I/O assignment, the DRCX controller must be restarted to enable the changes.
For more details, see "3-7 I/O assignment change function".

Input range: 0 or another number (See the I/O assignment list)

Meaning: PRM26 = x x x x

↑ ↑ ↑

③ ② ①

① I/O assignment type selection

00 : Type 0 (Conventional type/standard)
20, 21 : Type 2 (Point number output type)
30, 31 : Type 3 (Point teaching type)

* Type 1 cannot be used with the DRCX controller.

② Point output selection

Make setting only for type 2 (point number output type) and type 3 (point teaching type).

0: Outputs when movement ends normally.
1: Outputs when movement command is received.
2: Point zone output
PO output occurs when the robot enters the point data (registered at controller) ± position judgment parameter range.
3: Movement point zone output (supported in Ver. 18.64 and later versions):
PO output occurs when the robot enters the ± position judgment parameter range for the point data registered at the controller and which serves as the movement point data for the point movement command (ABS-PT, INC-PT).

③ Point zone judgment method selection (supported in Ver. 18.64 and later versions):

The position judgment parameter is selected when the point output selection is "2" (point zone output) or "3" (movement point zone output).

0: OUT valid position
1: Positioning-completed pulse

Default value: 0

CAUTION

Any value other than the above is handled as a "0" (type 0).
Example: If set to 2331, this is handled as a "0" (type 0).
If set to 10, this is handled as a "0" (type 0).
Moreover, if Type 2 (point signal output type) or Type 3 (point teaching type) is selected in DRCX versions prior to Ver. 18.64, with the point output selection specified as "3", this is processed as a "0" (Type 0) setting.
Example: At DRCX version prior to Ver. 13.64:
If set to 331, this is handled as a "0" (type 0).

NOTE

In controller versions prior to Ver. 18.64, the "OUT valid position" is the only point zone judgment method.

I/O assignment list

Type		Type 0 (Conventional type)	Type 1	Type 2 (Point number output type)		Type 3 (Point teaching type)			
						Point trace mode	Teaching mode	Point trace mode	Teaching mode
Function / PRM26 Setting		0 (Standard)	–	xx20 *1	xx21 *1	xx30 *1		xx31 *1	
No. of points *3		1000	–	64	16	64		16	
No. of speed switching points *4		4	–	None	4	None		4	
Program operation by I/O		Yes	–	No	No	No		No	
	Pin No.								
Input	A1	ABS-PT		ABS-PT	ABS-PT	ABS-PT	JOG+	ABS-PT	JOG+
	B1	INC-PT		INC-PT	INC-PT	INC-PT	JOG-	INC-PT	JOG-
	A2	AUTO-R		-	-	-	PSET	-	PSET
	B2	STEP-R		-	-	CHG		CHG	
	A3	ORG-S		ORG-S	ORG-S	ORG-S		ORG-S	
	B3	RESET		RESET	RESET	RESET		RESET	
	A4	SERVO		SERVO	SERVO	SERVO		SERVO	
	B4	LOCK		LOCK	LOCK	LOCK		LOCK	
	A5	DI0		PI0	PI0	PI0		PI0	
	B5	DI1		PI1	PI1	PI1		PI1	
	A6	DI2		PI2	PI2	PI2		PI2	
	B6	DI3		PI3	PI3	PI3		PI3	
	A7	DI4		PI4	SPD1	PI4		SPD1	
	B7	DI5		PI5	SPD2	PI5		SPD2	
	A8	DI6		-	-	-		-	
	B8	DI7		-	-	-		-	
	A9	DI8		-	-	-		-	
	B9	DI9		-	-	-		-	
	A10	DI10		-	-	-		-	
	B10	DI11	Cannot be used.	-	-	-		-	
	A11	DI12		-	-	-		-	
	B11	DI13 *2		*2	*2	*2		*2	
	A12	DI14 *2		*2	*2	*2		*2	
	B12	DI15/SVCE		(SVCE)	(SVCE)	(SVCE)		(SVCE)	
Output	A16	DO0		PO0	PO0	PO0		PO0	
	B16	DO1		PO1	PO1	PO1		PO1	
	A17	DO2		PO2	PO2	PO2		PO2	
	B17	DO3		PO3	PO3	PO3		PO3	
	A18	DO4		PO4	ORG-O/ZONE0	PO4		ORG-O/ZONE0	
	A20	DO5		PO5	SRV-O/ZONE1	PO5		SRV-O/ZONE1	
	B20	DO6		-	-	-		-	
	A21	DO7		-	-	-		-	
	B21	DO8		-	-	-		-	
	A22	DO9		-	-	-		-	
	B22	DO10		-	-	-		-	
	A23	DO11		-	-	-		-	
	B23	DO12		-	-	-		-	
	B18	END		END	END	END		END	
	A19	BUSY		BUSY	BUSY	BUSY		BUSY	
	B19	READY		READY	READY	READY		READY	

*1: The PO output format varies depending on whether the PRM26 setting is specified in "hundreds" or "thousands" units.
*2: A desired axis can be specified when executing I/O dedicated commands (ABS-PT, INC-PT, ORG-S, SERVO), by specifying DI13 and DI14 (pin No. B11 and A12). To do this, the PRM10 (control axis selection with I/O command) must first be set to "Enable". The same applies when specifying a control axis for JOG motion commands (JOG+, JOG-).
*3: Specifies the permissible number of movement points for a point movement command (ABS-PT, INC-PT).
*4: Specifies the permissible number of speed switching points for a point movement command (ABS-PT, INC-PT).

PRM27 to 47: Spare

5-2-2 X-axis parameters

PRM48: (+) soft limit

The + side robot movement range is set.
Set a suitable value for safety purposes.

Input range: -9999 to 9999 (mm)
or -360 to 360 (°)
Default value: Depends on robot type.

⚠ *CAUTION*
The soft limit will not work unless return-to-origin has been completed.

PRM49: (-) soft limit

The - side robot movement range is set.
Set a suitable value for safety purposes.

Input range: -9999 to 9999 (mm)
or -360 to 360 (°)
Default value: Depends on robot type.

⚠ *CAUTION*
The soft limit will not work unless return-to-origin has been completed.

PRM50: Payload

This specifies the total weight of the workpiece and tool attached to the X-axis of a Multi-Flip robot. In cases where this weight varies, enter the maximum payload.
Based on this parameter, the controller determines the optimum acceleration speed for the robot, so ensure that the correct payload is set. If set too small, abnormal vibration or overheat may occur resulting in troubles with the robot or controller. Conversely, if this parameter is larger than the actual payload, a loss of the cycle time occurs which lowers productivity.

Input range: Depends on robot type. Units are in kilograms (kg).
Default value: 0

* PRM50 can be specified only when the robot setting is for multiple FLIP-X. When specifying the payload of a dual axis robot, do not use this parameter but use PRM90.
* This parameter is set to X-axis maximum payload when the DRCX controller is shipped from factory (in case of Multi-Flip specifications).

PRM51: Acceleration

This parameter sets the acceleration.
The controller will automatically set optimum acceleration according to the robot type and payload. Change this parameter when the acceleration is to be decreased beyond this state.

Input range: 1 to 100 (%)
Default value: 100

PRM52: Return-to-origin direction

This parameter sets the return-to-origin direction.
Return-to-origin is usually performed toward the motor side when this parameter is set to 0, and toward the non-motor side when set to 1. However, this direction may be reversed depending on the robot variations (such as bent model and vertical type model).

Input range: 0 or 1
Default value: Depends on robot type.

* In terms of motor rotation, when this parameter is 0, the return-to-origin direction is CCW (counterclockwise) as seen from the load.

CAUTION

The return-to-origin direction cannot be changed in some robot types. Before attempting to change this parameter for the robot you are using, be sure to read the robot mechanical manual or catalog specs to check whether the return-to-origin direction can be changed.

PRM53: Positioning-completed pulse

This specifies the range in which the controller determines that positioning is complete.
When a movement command is executed, the robot moves toward the target position. The controller then determines that the positioning has been completed when the remaining distance to the target position is within this parameter setting. However, the robot continues moving until it reaches the target position even after the robot enters the "positioning-completed pulse" range.
Since executing the next movement command is not allowed until the positioning is complete, setting a large value for this parameter can reduce cycle time in cases where critical positioning accuracy is not required.

Input range: 1 to 4000 (pulses)
Default value: 80

* If the range specified by this parameter is larger than the range of the OUT valid position, the controller does not decide that the "positioning-completed pulse" range is entered until the axis reaches the OUT valid position.

PRM54: Return-to-origin speed

This specifies the movement speed during return- to-origin.

Input range: 1 to 100 (mm/sec)
Default value: 20

CAUTION

When the return-to-origin speed is increased, an alarm might be issued during return-to-origin depending on the robot type. We recommend using the default value as much as possible.

PRM55: Origin detection method

This parameter is used to select the origin (reference point) detection method.
There are two methods for detecting the origin: search method and mark method. The search method is further divided into the origin sensor method and stroke-end detection method. In the mark method, you can move the robot to a desired position (mark position) and set it as the particular coordinate position to determine a reference point.
Set this parameter to "0" when detecting the origin position with an origin sensor (sensor method), or set to "1" when detecting the origin by the stroke-end detection method, or set to "2" when using the mark method.

Input range: 0 to 2
Meaning: 0: Sensor method
1: Stroke-end detection method
2: Mark method
Default value: Depends on robot type.

CAUTION
When you are using a robot with this parameter set to "sensor method" by default, do not change it to "stroke-end detection method". Failure to follow this instruction might damage the robot mechanical parts.

PRM56: OUT valid position

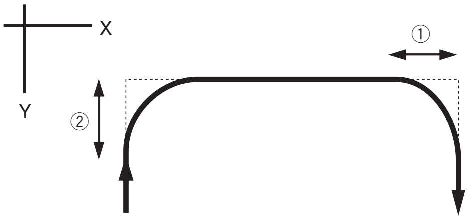
This specifies the range in which the controller determines that movement command is complete.
When a movement command is executed, the robot moves toward the target position. The controller then determines that the movement command has ended when the remaining distance to the target position is within this parameter setting. The controller then initiates the subsequent step processing when the robot reaches this OUT valid position, so setting this parameter to a larger value can reduce cycle time.
However, if the subsequent command is a movement command, it is not executed until the ongoing positioning is complete.

Input range: 0 to 9999 (mm)
0 to 360 (°)
Default value: 1

PRM57: Arch position

When moving the robot in an arch motion, this parameter sets how many millimeters (degrees) prior to the X-axis target position, to start Y-axis movement. This parameter determines a form of arch motion.

Input range : 0 to 9999 (mm)
0 to 360 (°)
Default value : 5



① determined by PRM57
② determined by PRM97

In the above case, the Y-axis moves upward and the X-axis starts to move at the remaining distance position ②.
Then, when the X-axis reaches the remaining distance position ①, the Y-axis starts to move downward.

PRM58: Auxiliary axis stroke

Default value: Depends on robot type.

PRM59: No. of encoder pulses (4× mode)

This parameter sets the number of signal pulses (resolver resolution) per one turn of the motor.

Default value: 16384 (pulse/rev.)

PRM60: Lead length

This parameter sets the robot lead length (distance the robot moves while the motor makes one turn). For rotational type robots such as the FROP, this parameter is set to an angle through which the robot rotates while the motor makes one turn.

Default value: Depends on robot type. (Unit: 0.01mm or 0.01deg.)

PRM61: Overload current

This sets the reference current value used to detect an overload.

Default value: Equal to the motor rated current.

PRM62: Overload time

This specifies conditions such as time required to detect an overload.
The default value is set so that an overload alarm is issued when a current three times higher than the overload current (PRM61) flows for a period of 3 seconds or an equivalent condition is detected.

Default value: 240

PRM63: Current limit

This sets the maximum motor input current.

Default value: Depends on robot type.

PRM64: Maximum speed setting

This parameter sets the maximum motor revolution speed.

Input range: 1 to 4500 (rpm)
Default value: Depends on robot type.

⚠ ***CAUTION***

Changing this parameter carelessly might shorten the robot service life or cause other problems.

PRM65: Speed proportional gain

This sets the speed control gain. Typically, PRM65 and PRM66 should be input at a ratio of 3 : 2.
Generally, the larger the gain, the higher the acceleration will be. However, if the gain is set too high, abnormal oscillation or noise might be generated, causing serious problems in the robot and controller. Use caution when selecting this parameter to avoid such problems.

Default value: Depends on robot type.

PRM66: Speed integration gain

This sets the speed control gain. Typically, PRM65 and PRM66 should be input at a ratio of 3 : 2.
Generally, the larger the gain, the higher the acceleration will be. However, if the gain is set too high, abnormal oscillation or noise might be generated, causing serious problems in the robot and controller. Use caution when selecting this parameter to avoid such problems.

Default value: Depends on robot type.

PRM67: Speed delay compensation gain

Default value: Depends on robot type.

PRM68: Position proportional gain

This sets the position control gain.
If this parameter is changed carelessly, serious problems may occur in the robot and controller.

Default value: Depends on robot type.

PRM69: Interpolation movement speed gain coefficient

Default value: Depends on robot type.

PRM70: Position data unit

This parameter sets the units in which point data is to be displayed. It also specifies whether to enable the limitless movement function.

Input range: 0 to 3
Meaning: 0: mm (millimeters); limitless movement function disabled (off)
1: ° (deg.); limitless movement function disabled (off)
2: mm (millimeters), limitless movement function enabled (on)
3: ° (deg.); limitless movement function enabled (on)
Default value: Depends on robot type.

For more details, see "8-3-2 Limitless movement function".

PRM71: Payload-dependent acceleration coefficient

The value calculated from PRM58, PRM72 and PRM60, PRM50, is set automatically for this parameter.

Default value: Depends on robot type.

PRM72: Axis robot type number

This parameter shows the robot type number currently used as the X-axis. (See "15-1-2 Robot number list".)
This is a read-only parameter. When changing the robot number or if the memory contents are corrupted, perform parameter initialization. (See "10-1 Initialization".)

* If the robot setting is made for a 2-axis robot, the robot type number is shown by PRM18.

PRM73: Origin search data

This specifies the performance data for detecting the origin position during return-to-origin by the origin search method.

Default value: Depends on robot type.

PRM74: Open-circuit fault detection level

This parameter sets the sensitivity for detecting an open-circuit fault. The upper limit of this parameter is 254. The sensitivity lowers as the parameter value increases. Leave this parameter set to 255 if you want to disable this detection function.

Input range: 1 to 255
Default value: 255 (This function is disabled.)

PRM75: Not used

Default value: 0

PRM76: Not used

Default value: 0

PRM77: Origin shift

This parameter specifies a shift to the origin position after return-to-origin is complete. When return-to-origin is complete, the origin position is usually "0" (specified value when the mark method is used). If for some reason the origin position needs to be shifted by a particular amount, then change this parameter. For example, if an unwanted position shift occurred, then reteaching of all point data needs to be performed. However, the time and effort needed for this reteaching can be eliminated by setting the shift amount for this parameter to quickly correct the point data.

Input range: -9999 to 9999 (0.01mm)
or -9999 to 9999 (0.01°)
Default value: 0

* The parameter change is enabled after reperforming return-to-origin.

PRM78: QP band width

This parameter specifies the control switching point (pulse width) that compensates for the frictional resistance during deceleration.

Input range: 1 to 1000 (pulses)
Default value: Depends on robot type.

PRM79: No. of motor poles

Default value: Depends on robot type.

PRM80: Feed forward gain

Default value: Depends on robot type.

PRM81: Deceleration (Available with Ver. 18.25 or later)

Use this parameter to reduce only the deceleration.
When this parameter is left set to the default value (100), the deceleration is the same as the acceleration. If vibration occurs during positioning, then set this parameter to a smaller value to reduce only the deceleration.
This parameter value can be changed in 1% steps, with 100% equal to the value determined by PRM51.

Input range: 1 to 100 (%)
Default value: 100

PRM82: Hold gain (Available with Ver. 18.50 or later)

Default value: Depends on the robot.

PRM83: Not used

Default value: 0

PRM84: Not used

Default value: 0

PRM85: Not used

Default value: Depends on robot type.

CAUTION
Do not change the setting.

PRM86 to 87: Spare

5-2-3 Y-axis parameters

PRM88: (+) soft limit

The + side robot movement range is set.
Set a suitable value for safety purposes.

Input range: -9999 to 9999 (mm)
or -360 to 360 (°)
Default value: Depends on robot type.

CAUTION
The soft limit will not work unless return-to-origin has been completed.

PRM89: (-) soft limit

The - side robot movement range is set.
Set a suitable value for safety purposes.

Input range: -9999 to 9999 (mm)
or -360 to 360 (°)
Default value: Depends on robot type.

CAUTION
The soft limit will not work unless return-to-origin has been completed.

PRM90: Payload

This specifies the total weight of the workpiece and tool attached to the Y-axis of a Multi-Flip robot. In cases where this weight varies, enter the maximum payload.
Based on this parameter, the controller determines the optimum acceleration speed for the robot, so ensure that the correct payload is set. If set too small, abnormal vibration or overheat may occur resulting in troubles with the robot or controller. Conversely, if this parameter is larger than the actual payload, a loss of the cycle time occurs which lowers productivity.

Input range: Depends on robot type. Units are in kilograms (kg).
Default value: 0

* This parameter is set to maximum payload when the controller is shipped from factory.

PRM91: Acceleration

This parameter sets the acceleration.
The controller will automatically set optimum acceleration according to the robot type and payload. Change this parameter when the acceleration is to be decreased beyond this state.

Input range: 1 to 100 (%)
Default value: 100

PRM92: Return-to-origin direction

This parameter sets the return-to-origin direction.
Return-to-origin is usually performed toward the motor side when this parameter is set to 0, and toward the non-motor side when set to 1. However, this direction may be reversed depending on the robot variations (such as bent model and vertical type model).

Input range: 0 or 1
Default value: Depends on robot type.

* In terms of motor rotation, when this parameter is 0, the return-to-origin direction is CCW (counterclockwise) as seen from the load.

CAUTION
The return-to-origin direction cannot be changed in some robot types. Before attempting to change this parameter for the robot you are using, be sure to read the robot mechanical manual or catalog specs to check whether the return-to-origin direction can be changed.

PRM93: Positioning-completed pulse

This specifies the range in which the controller determines that positioning is complete.
When a movement command is executed, the robot moves toward the target position. The controller then determines that the positioning has been completed when the remaining distance to the target position is within this parameter setting. However, the robot continues moving until it reaches the target position even after the robot enters the "positioning-completed pulse" range.
Since executing the next movement command is not allowed until the positioning is complete, setting a large value for this parameter can reduce cycle time in cases where critical positioning accuracy is not required.

Input range: 1 to 4000 (pulses)
Default value: 80

* If the range specified by this parameter is larger than the range of the OUT valid position, the controller does not decide that the "positioning-completed pulse" range is entered until the axis reaches the OUT valid position.

PRM94: Return-to-origin speed

This specifies the movement speed during return- to-origin.

Input range: 1 to 100 (mm/sec)
Default value: 20

CAUTION
When the return-to-origin speed is increased, an alarm might be issued during return-to-origin depending on the robot type. We recommend using the default value as much as possible.

PRM95: Origin detection method

This parameter is used to select the origin (reference point) detection method.
There are two methods for detecting the origin: search method and mark method. The search method is further divided into the origin sensor method and stroke-end detection method. In the mark method, you can move the robot to a desired position (mark position) and set it as the particular coordinate position to determine a reference point.
Set this parameter to "0" when detecting the origin position with an origin sensor (sensor method), or set to "1" when detecting the origin by the stroke-end detection method, or set to "2" when using the mark method.

Input range: 0 to 2
Meaning: 0: Sensor method
1: Stroke-end detection method
2: Mark method
Default value: Depends on robot type.

CAUTION
When you are using a robot with this parameter set to "sensor method" by default, do not change it to "stroke-end detection method". Failure to follow this instruction might damage the robot mechanical parts.

PRM96: OUT valid position

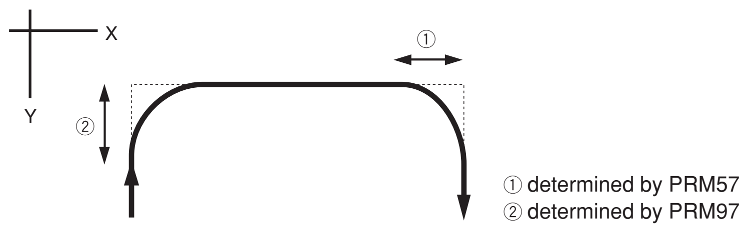
This specifies the range in which the controller determines that movement command is complete.
When a movement command is executed, the robot moves toward the target position. The controller then determines that the movement command has ended when the remaining distance to the target position is within this parameter setting. The controller then initiates the subsequent step processing when the robot reaches this OUT valid position, so setting this parameter to a larger value can reduce cycle time.
However, if the subsequent command is a movement command, it is not executed until the ongoing positioning is complete.

Input range: 0 to 9999 (mm)
0 to 360 (°)
Default value: 1

PRM97: Arch position

When moving the robot in an arch motion, this parameter sets how many millimeters (degrees) prior to the Y-axis target position, to start X-axis movement. This parameter determines a form of arch motion.

Input range : 0 to 9999 (mm)
0 to 360 (°)
Default value : 5



In the above case, the Y-axis moves upward and the X-axis starts to move at the remaining distance position ②.
Then, when the X-axis reaches the remaining distance position ①, the Y-axis starts to move downward.

PRM98: Auxiliary axis stroke

Default value: Depends on robot type.

PRM99: No. of encoder pulses (4× mode)

This parameter sets the number of signal pulses (resolver resolution) per one turn of the motor.

Default value: 16384 (pulse/rev.)

PRM100: Lead length

This parameter sets the robot lead length (distance the robot moves while the motor makes one turn). For rotational type robots such as the FROP, this parameter is set to an angle through which the robot rotates while the motor makes one turn.

Default value: Depends on robot type. (Unit: 0.01mm or 0.01deg.)

PRM101: Overload current

This sets the reference current value used to detect an overload.

Default value: Equal to the motor rated current.

PRM102: Overload time

This specifies conditions such as time required to detect an overload.
The default value is set so that an overload alarm is issued when a current three times higher than the overload current (PRM101) flows for a period of 3 seconds or an equivalent condition is detected.

Default value: 240

PRM103: Current limit

This sets the maximum motor input current.

Default value: Depends on robot type.

PRM104: Maximum speed setting

This parameter sets the maximum motor revolution speed.

Input range: 1 to 4500 (rpm)
Default value: Depends on robot type.

CAUTION

Changing this parameter carelessly might shorten the robot service life or cause other problems.

PRM105: Speed proportional gain

This sets the speed control gain. Typically, PRM105 and PRM106 should be input at a ratio of 3 : 2.
Generally, the larger the gain, the higher the acceleration will be. However, if the gain is set too high, abnormal oscillation or noise might be generated, causing serious problems in the robot and controller. Use caution when selecting this parameter to avoid such problems.

Default value: Depends on robot type.

PRM106:Speed integration gain

This sets the speed control gain. Typically, PRM105 and PRM106 should be input at a ratio of 3 : 2.
Generally, the larger the gain, the higher the acceleration will be. However, if the gain is set too high, abnormal oscillation or noise might be generated, causing serious problems in the robot and controller. Use caution when selecting this parameter to avoid such problems.

Default value: Depends on robot type.

PRM107:Speed delay compensation gain

Default value: Depends on robot type.

PRM108:Position proportional gain

This sets the position control gain.
If this parameter is changed carelessly, serious problems may occur in the robot and controller.

Default value: Depends on robot type.

PRM109:Interpolation movement speed gain coefficient

Default value: Depends on robot type.

PRM110:Position data unit

This parameter sets the units in which point data is to be displayed. It also specifies whether to enable the limitless movement function.

Input range:	0 to 3
Meaning:	0: mm (millimeters); limitless movement function disabled (off)
	1: ° (deg.); limitless movement function disabled (off)
	2: mm (millimeters), limitless movement function enabled (on)
	3: ° (deg.); limitless movement function enabled (on)
Default value:	Depends on robot type.

For more details, see "8-3-2 Limitless movement function".

PRM111: Payload-dependent acceleration coefficient

The value calculated from PRM98, PRM112 and PRM100, PRM90, is set automatically for this parameter.

Default value: Depends on robot type.

PRM112:Axis robot type number

This parameter shows the robot type number currently used as the Y-axis. (See "15-1-2 Robot number list".)
This is a read-only parameter. When changing the robot number or if the memory contents are corrupted, perform parameter initialization. (See "10-1 Initialization".)

* If the robot setting is made for a 2-axis robot, the robot type number is shown by PRM18.

PRM113: Origin search data

This specifies the performance data for detecting the origin position during return-to-origin by the origin search method..

Default value: Depends on robot type.

PRM114: Open-circuit fault detection level

This parameter sets the sensitivity for detecting an open-circuit fault. The upper limit of this parameter is 254. The sensitivity lowers as the parameter value increases. Leave this parameter set to 255 if you want to disable this detection function.

Input range: 1 to 255
Default value: 255 (This function is disabled.)

PRM115: Not used

Default value: 0

PRM116: Not used

Default value: 0

PRM117: Origin shift

This parameter specifies a shift to the origin position after return-to-origin is complete. When return-to-origin is complete, the origin position is usually "0" (specified value when the mark method is used). If for some reason the origin position needs to be shifted by a particular amount, then change this parameter. For example, if an unwanted position shift occurred, then reteaching of all point data needs to be performed. However, the time and effort needed for this reteaching can be eliminated by setting the shift amount for this parameter to quickly correct the point data.

Input range: -9999 to 9999 (0.01mm)
or -9999 to 9999 (0.01°)
Default value: 0

* The parameter change is enabled after reperforming return-to-origin.

PRM118: QP band width

This parameter specifies the control switching point (pulse width) that compensates for the frictional resistance during deceleration.

Input range: 1 to 1000 (pulses)
Default value: Depends on robot type.

PRM119: No. of motor poles

Default value: Depends on robot type.

PRM120: Feed forward gain

Default value: Depends on robot type.

PRM121: Deceleration (Available with Ver. 18.25 or later)

Use this parameter to reduce only the deceleration.
When this parameter is left set to the default value (100), the deceleration is the same as the acceleration. If vibration occurs during positioning, then set this parameter to a smaller value to reduce only the deceleration.
This parameter value can be changed in 1% steps, with 100% equal to the value determined by PRM91.

Input range: 1 to 100 (%)
Default value: 100

PRM122: Hold gain (Available with Ver. 18.50 or later)

Default value: Depends on the robot.

PRM123: Not used

Default value: 0

PRM124: Not used

Default value: 0

PRM125: Not used

Default value: Depends on robot type.

CAUTION
Do not change the setting.

PRM126 to 127: Spare

Chapter 6 PROGRAMMING

In this chapter we will try programming some operations. First, you will learn how to enter a program using the TPB programming box.

6-1 Basic Contents

6-1-1 Robot language and point data

The DRCX controller uses the YAMAHA robot language that is very similar to BASIC. It allows you to easily create programs for robot operation.
In programs created with the YAMAHA robot language, the robot position data (absolute position, amount of movement) are not expressed in terms of direct numeric values. Instead, point numbers are used to express the position data indirectly. Point numbers and their corresponding position information are stored as point data separately from programs. This means that when you want to change only the position information while using the same program, all that you have to do is edit the point data.

Example

	Program	Point Data
	:	
005:	MOVA 0, 100	P0 = 50.00,20.00
006:	MOVI 1, 50	P1 = 100.00,200.00
	:	

In the above example, the robot first travels 50mm on the X-axis, then 20mm on the Y-axis, then again 100mm on the X-axis, and 200mm on the Y-axis.

To change the above operation so that the robot first moves to a position 50.5mm on the X-axis, then 21mm on the Y-axis from the origin point and then moves to another point 100mm on the X-axis, 200mm on the Y-axis away from that position, just change the P0 point data to P0=50.50,21.00.

6-1-2 Using the TPB to enter the robot language

Robot language commands frequently used to create programs are printed on the lower part of each number key on the TPB. When creating or editing a program, you can enter robot language commands simply by pressing these keys. To select other robot language commands not printed on these keys, use the function key matching that command.
During program editing, you can enter numbers (numerical values) with the number keys except when the edit cursor for robot language command input appears on the TPB screen.

6-1-3 Program specifications

The DRCX controller has the following memory capacity:

Total number of programs	: 100 programs (NO0 to NO99)
Max. number of steps per program	: 255 steps
Max. number of steps in all programs together	: 3000 steps
Max. number of points	: 1000 points (P0 to P999)

6-2 Editing Programs

"Program editing" refers to operations such as creating a program right after initialization, creating a new program, changing an existing program, and deleting or copying a program. In this section, you will learn the basic procedures for program editing using the TPB.

"Creating a program right after initialization" means creating a program for the first time after purchasing the controller or creating a program right after initialization while there are still no programs stored in the controller (see "10-1 Initialization").
"Creating a new program" means creating or editing a new program while at least one program has already been created and stored.
"Changing an existing program" means correcting, adding, deleting, or inserting steps in a program to change only part of it.

This section explains all the above program editing procedures, and also describes how to view program information such as the number of steps left in a program.

6-2-1 Creating programs after initialization

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (PGM).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) Since no program is registered after initialization, an error message appears on the screen, indicating that no program exists.

```
[EDIT]
select menu
43:cannot find PGM
1PGM 2PNT 3UTL
```

4) Press the ESC key to reset the error. A confirmation message then appears asking whether to create a new program as program No. 0.
To select and edit program No. 0, press F1 (yes).
To select and edit a program other than No. 0, press F2 (no).

```
[EDIT-PGM]
PGM  No = 0
New entry OK ?
1yes 2no
```

5) When you selected F2 (no) in step 4, enter the number of the program to be edited with the number keys and press ⇨. The screen returns to step 4. Make sure the program number is correct and press F1 (yes).

```
[EDIT-PGM]
PGM  No = _
(Program No) 0→99
```

6) Select F1 to F3 or a robot language command shown on the lower part of each number key.
To change the robot language menu display, press F4 (next). To go back to the previous menu display, press the BS key.

```
[EDIT-PGM]      No 0
001:_

1MAT2MSEL3MOVM 4next
```

6
PROGRAMMING

7) After selecting the robot language command, enter the operand data.

When you press X Z+, the cursor moves to operand 1, so enter the data with the number keys. (Do not press ➡ at this point.)

While pressing X Z+ or X Z− to move the cursor, enter all necessary operand data as needed.

```
[EDIT-PGM]       No 0
001:MOVA 0   ,100
(point No)0→999
1P
```

8) After entering the operand data, press ➡.

```
[EDIT-PGM]       No 0
001:MOVA 1   ,80_
(speed)1→100
```

9) When entry is completed correctly, the cursor moves to the operation code part.

To edit the next step, press STEP UP to scroll the step and repeat the procedure from step 6.

```
[EDIT-PGM]       No 0
001:MOVA 1   ,80

1MAT2MSEL3MOVM 4next
```

6-2-2 Creating a new program

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (PGM).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) The execution program number and step are displayed on the screen. Press F4 (CHG) here.

```
[EDIT-PGM]      No10
017:MOVA 254,100

1MOD 2INS 3DEL 4CHG
```

4) Enter the new program number with the number keys and press ➡.

```
[EDIT-PGM]
PGM  No = _
(Program No) 0→99
```

5) A confirmation message appears. Make sure the program number is correct and press F1 (yes).

```
[EDIT-PGM]
PGM  No = 14
New entry OK ?
1yes 2no
```

6) Proceed with program editing by following step 6 onward in "6-2-1 Creating programs after initialization."

```
[EDIT-PGM]      No14
001:_

1MAT2MSEL3MOVM 4next
```

6-2-3 Adding a step

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (PGM).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) The execution program number and step are displayed on the screen. Press F4 (CHG) here.

```
[EDIT-PGM]      No10
017:MOVA 254,100

1MOD 2INS 3DEL 4CHG
```

4) Enter the program number you want to edit with the number keys and press ➡.

```
[EDIT-PGM]
PGM  No = _
(Program No) 0→99
```

5) Enter the last step number with the number keys and press ➡.

```
[EDIT-PGM]
PGM  No = 10
STEP No = _
(REG.steps) 50
```

6) When the last step is displayed, press STEP UP.

```
[EDIT-PGM]       No10
050:WAIT 3   ,1

1MOD 2INS 3DEL 4CHG
```

7) Select F1 to F3 or a robot language command shown on the lower part of each number key.

To change the robot language menu display, press F4 (next). To go back to the previous menu display, press the BS key.

```
[EDIT-PGM]        No10
051:_


1MAT2MSEL3MOVM 4next
```

8) After selecting the robot language command, enter the operand data.

When you press X/Z+, the cursor moves to operand 1, so enter the data with the number keys. (Do not press ➡ at this point.)

While pressing X/Z+ or X/Z− to move the cursor, enter all necessary operand data as needed.

```
[EDIT-PGM]        No10
051:JMPF 0  ,10 ,1
(label No) 0→255
```

9) After entering the operand data, press ➡.

```
[EDIT-PGM]        No10
051:JMPF 10 ,31 ,5_
(DI condition) 0→255
```

10) When the program has been edited correctly, the screen returns to step 6.

When you want to add another step, press STEP UP to scroll to the next step and then repeat from step 7.

```
[EDIT-PGM]       No10
051:JMPF 10 ,31 ,5


1MOD 2INS 3DEL 4CHG
```

6-2-4 Correcting a step

1) Use the same procedure up to step 4 in "6-2-3 Adding a step".

2) Enter the number of the step you want to correct with the number keys and press ⇨.

```
[EDIT-PGM]
PGM  No = 10
STEP No = _
(REG.steps) 50
```

3) Press F1 (MOD).

```
[EDIT-PGM]       No10
010:MOVA 999,100

1MOD 2INS 3DEL 4CHG
```

4) Select F1 to F3 or a robot language command shown on the lower part of each number key.

   To change the robot language menu display, press F4 (next). To go back to the previous menu display, press the BS key.

```
[EDIT-PGM]       No10
010:MOVA 999,100

1MAT2MSEL3MOVM 4next
```

5) After selecting the robot language command, enter the operand data.

   When you press X/Z+, the cursor moves to operand 1, so enter the data with the number keys. (Do not press ⇨ at this point.)

   While pressing X/Z+ or X/Z− to move the cursor, enter all necessary operand data as needed.

```
[EDIT-PGM]       No10
010:MOVA 10_,100
(point No) 0→999
1P
```

6) After entering the operand data, press ⇨.

```
[EDIT-PGM]       No10
010:MOVA 10 ,100
(speed) 1→100
```

7) When entry is completed correctly, the cursor moves to the operation code part.

   If you want to change another step, press STEP UP to scroll the step and repeat the procedure from step 4.

```
[EDIT-PGM]      No10
010:MOVA 10 ,100

1MAT2MSEL3MOVM 4next
```

6 PROGRAMMING

6-2-5 Inserting a step

1) Use the same procedure up to step 4 in "6-2-3 Adding a step".

2) Enter the number of the step where you want to insert a step with the number keys and press ➡.

```
[EDIT-PGM]
PGM  No = 10
STEP No = _
(REG steps) 50
```

3) Press F2 (INS).

```
[EDIT-PGM]      No10
010:MOVA 999,100

1MOD 2INS 3DEL 4CHG
```

4) Select F1 to F3 or a robot language command shown on the lower part of each number key.

   To change the robot language menu display, press F4 (next). To go back to the previous menu display, press the BS key.

```
[EDIT-PGM]      No10
010:_

1MAT 2MSEL3MOVM4next
```

5) After selecting the robot language command, enter the operand data.

   When you press X+/Z+, the cursor moves to operand 1, so enter the data with the number keys. (Do not press ➡ at this point.)

   While pressing X+/Z+ or X-/Z- to move the cursor, enter all necessary operand data as needed.

```
[EDIT-PGM]      No10
010:MOVA 10_,100
(point No) 0→999
1P
```

6) After entering the operand data, press ➡.

```
[EDIT-PGM]      No10
010:MOVA 10 ,100
(speed) 1→100
```

7) When entry is completed correctly, the screen returns to step 3.

```
[EDIT-PGM]      No10
010:MOVA 10 ,100

1MOD 2INS 3DEL 4CHG
```

6-2-6 Deleting a step

1) Use the same procedure up to step 4 in "6-2-3 Adding a step".

2) Enter the number of the step you want to delete with the number keys and press ⇨.

```
[EDIT-PGM]
PGM  No = 10
STEP No = _
(REG steps) 50
```

3) Press F3 (DEL).

```
[EDIT-PGM]       No10
010:MOVA 999,100

1MOD 2INS 3DEL 4CHG
```

4) A confirmation message appears.
To delete the step, press F1 (yes).
To cancel the deletion, press F2 (no).

```
[EDIT-PGM]       No10
010:MOVA 999,100
delete OK ?
1yes 2no
```

5) When the step has been deleted, the screen returns to step 3.

```
[EDIT-PGM]       No10
010:WAIT 3  ,1

1MOD 2INS 3DEL 4CHG
```

6

PROGRAMMING

6-3 Program Utility

6-3-1 Copying a program

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F3 (UTL).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) Press F1 (COPY).

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

4) Enter the program number you want to copy from with the number keys, and then press ➡.

```
[EDIT-UTL-COPY]
Copy from No = _
(Program No) 0→99
```

5) Enter the program number you want to copy to with the number keys, and then press ➡.

```
[EDIT-UTL-COPY]
Copy from No = 0
Copy to   No = 99_
(Program No) 0→99
```

6) If program data is already registered with the selected program number, a confirmation message appears.
   To overwrite the program, press F1 (yes).
   To cancel, press F2 (no).

```
[EDIT-UTL-COPY]
Copy from No = 0
No99 overwrite OK ?
1yes 2no
```

7) When the program has been copied, the screen returns to step 3.

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

6-3-2 Deleting a program

1) Use the same procedure up to step 2 in "6-3-1 Copying a program".

2) Press F2 (DEL).

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

3) Enter the number of the program you want to delete with the number keys and press ➡.

```
[EDIT-UTL-DEL]
delete PGM No = _
(Program No) 0→99
```

4) A confirmation message appears asking whether to delete the selected program.
   To delete the program, press F1 (yes).
   To cancel the deletion, press F2 (no).

```
[EDIT-UTL-DEL]
delete PGM No = 22
delete OK ?
1yes 2no
```

5) If the program has been deleted, the screen returns to step 2.

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

6-3-3 Viewing the program information

1) Use the same procedure up to 2 in "6-3-1 Copying a program".

2) Press F3 (LIST).

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

3) The program numbers are displayed on the screen, along with the number of registered steps and the number of available remaining steps.

   To view other program information, press the STEP UP and STEP DOWN keys to scroll the screen.

```
[EDIT-UTL-LIST]
  free   678 steps
  No 0    57 steps
  No 1   255 steps
```

4) Press the ESC key to return to the screen of step 2.

```
[EDIT-UTL]
select menu

1COPY2DEL 3LIST
```

* In addition to the number of existing steps, the steps equivalent to the number of programs are used internally as the program control steps. For example, if two programs are registered and their respective 50 and 100 steps are registered, then the number of available remaining steps will be as follows:
3000 – 2 – 50 – 100 = 2848 steps

Chapter 7 EDITING POINT DATA

There are three methods to enter point data: manual data input (MDI), teaching playback, and direct teaching.
Manual data input allows you to directly enter point data with the TPB number keys.
Teaching playback moves the robot in manual operation to a desired position and then obtains that position as point data.
Direct teaching is basically the same as teaching playback, except that you move the robot by hand.

7-1 Manual Data Input

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F2 (PNT).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) Press F1 (MDI).

```
[EDIT-PNT]
select menu

1MDI 2TCH 3DTCH4DEL
```

4) The currently selected X-axis point data in the execution program is displayed on the screen.
If you want to edit another point data, press the STEP UP and STEP DOWN keys to scroll the point data.
To directly select the point data, press F1 (CHG).

```
[EDIT-PNT-MDI]
P0    X=0.00      [mm]
 [0.00    ,0.00    ]
1CHG 2PLT
```

5) Enter the point number you want to edit with the number keys, and press ➡.

```
[EDIT-PNT-MDI]
 Pn : n = _
(point No) 0→999
```

6) Enter the point data with the number keys.
To edit the Y-axis data, press X/Z+.
To edit the X-axis data again, press X/Z−.
After entering the X and Y axis data, press ➡.

```
[EDIT-PNT-MDI]
P500 X=-19.27    [mm]
 [21.76_   ,54.31    ]
1CHG 2PLT
```

7) The input data is then registered as point data.

```
[EDIT-PNT-MDI]
P500 X=21.76     [mm]
 [21.76    ,54.31    ]
1CHG 2PLT
```

7 EDITING POINT DATA

7-2 Teaching Playback

1) On the initial screen, press F1 (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F2 (PNT).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) Press F2 (TCH).

```
[EDIT-PNT]
select menu

1MDI 2TCH 3DTCH4DEL
```

4) The currently selected point data in the execution program appears on the screen.

   If you want to edit another point data, press the STEP UP and STEP DOWN keys to scroll the point data.

   To directly select the point data, press F1 (CHG).

```
[EDIT-PNT-TCH](1) 50
P0   X=0.00      [mm]
 [    0.00,     0.00]
1CHG  2DO  3Y  4next
```

5) Enter the point number you want to edit with the number keys, and press ➡.

```
[EDIT-PNT-TCH](1) 50
 Pn : n = _
(point No)  0→999
```

6) Move the robot to the teaching position. Each time the X/Z− or X/Z+ key is pressed, the robot moves a certain amount in X-axis direction and then stops.

Holding down the X/Z− or X/Z+ key moves the robot continuously in the X-axis direction at a constant speed until the key is released.

To move the robot in the Y-axis direction, press Y/R− or Y/R+.

The amount of robot movement and the speed are proportional to the number (teaching movement data) displayed on the upper right of the screen.

In the example at the right, the teaching movement data is 50 (%), so the robot moves 0.5mm each time the X/Z− or X/Z+ key is pressed, as calculated below:

1mm (constant) × (50/100) = 0.5mm

If the X/Z− or X/Z+ key is kept pressed, the robot continuously moves at a speed of 50mm/s, as calculated below:

100mm/s (constant) × (50/100) = 50mm/s

```
[EDIT-PNT-TCH](1) 50
P0   X=0.00     [mm]
 [    0.00,    0.00]
1CHG 2DO  3Y   4next
```

7) Three different speed settings, SPEED (1), SPEED (2), and SPEED (3), are selectable as the teaching movement data. After pressing F4 (next), each time F1 (SPD) is pressed, the settings change in the order of 1→2→3→1.

To change the teaching movement data setting, press F4 (next) and then press F2 (S_SET). Then enter the desired speed with the number keys and press ➡.

The screen returns to step 6 when the speed parameter has been changed correctly.

```
[EDIT-PNT-TCH](1) 50
SPEED(1) = _
(speed) 1→100
```

8) Move the robot to the teaching position in this way and press the ➡ key. The current position is input as point data.

When registering again the point which is already registered, the axis selection screen appears.

To register both X and Y axes, press F1 (XY).

To register the X-axis only, press F2 (X). To register the Y-axis only, press F3 (Y).

```
[EDIT-PNT-TCH](1) 100
P500 X=19.27    [mm]
 [  167.24,  -51.58]
1XY  2X   3Y
```

9) The screen will return to step 6, and the current position will be input as the point data.

```
[EDIT-PNT-TCH](1) 100
P500 X=167.24   [mm]
 [  167.24,  -51.58]
1CHG 2DO  3Y   4next
```

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***Robot movement speed is limited to 10mm/s or less (10 deg/s for rotary robot) in "SERVICE mode state" when the robot movement speed limit is enabled.***

7-3 Direct Teaching

1) On the initial screen, press [F1] (EDIT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press [F2] (PNT).

```
[EDIT]
select menu

1PGM 2PNT 3UTL
```

3) Press [F3] (DTCH).

```
[EDIT-PNT]
select menu

1MDI 2TCH 3DTCH4DEL
```

4) Following the message, press the emergency stop button on the TPB.

```
[EDIT-PNT-DTCH]
press EMG.button
```

5) The currently selected X-axis point data in the execution program is displayed on the screen.
If you want to edit another point data, press the [STEP UP] and [STEP DOWN] keys to scroll the point data.
To directly select the point data, press [F1] (CHG).

```
[EDIT-PNT-DTCH]
P0    X=0.00     [mm]
 [    0.00,    0.00]
1CHG 2DO  3Y  4next
```

6) Enter the point number you want to edit with the number keys, and press ⇨.

```
[EDIT-PNT-DTCH]
 Pn : n = _
(point No)  0→999
```

7 EDITING POINT DATA

7) Move the robot to the teaching position by hand.

   To check the point data which is already registered, press F3 (X (Y)) and change the axis for point data display.

```
[EDIT-PNT-DTCH]
P500 X=19.27     [mm]
 [    0.00,     0.00]
1CHG 2DO  3Y   4next
```

8) Move the robot to the teaching position in this way and press the ➡ key. The current position is input as point data.

   When registering again the point which is already registered, the axis selection screen appears.

   To register both X and Y axes, press F1 (XY).

   To register the X-axis only, press F2 (X). To register the Y-axis only, press F3 (Y).

```
[EDIT-PNT-DTCH]
P500 X=19.27     [mm]
 [  167.24,   -51.58]
1XY  2X   3Y
```

9) The screen will return to step 7, and the current position will be input as the point data.

   When all points have been edited in the same way, press the ESC key.

```
[EDIT-PNT-DTCH]
P500 X=167.24    [mm]
 [  167.24,   -51.58]
1CHG 2DO  3Y   4next
```

10) Following the message, release the emergency stop button on the TPB.

```
[EDIT-PNT-DTCH]
release EMG.button
```

11) A confirmation message appears asking whether to turn the servo on.

    To turn the servo on, press F1 (yes).

    To leave the servo off, press F2 (no).

```
[EDIT-PNT-DTCH]
servo on ready ?

1yes 2no
```

12) The screen returns to step 3.

```
[EDIT-PNT]
select menu

1MDI 2TCH 3DTCH4DEL
```

7-4 Manual Control of General-Purpose Output

When performing teaching playback or direct teaching with systems that use a general-purpose output through the I/O interface to operate a gripper or other tools, you may want to check the position of workpiece by actually moving it.
For this reason, the DRCX controller is designed to allow manual control of general-purpose outputs from the TPB.

1) Move the robot with the same procedure up to step 6 in "7-2 Teaching Playback" or up to step 7 in "7-3 Direct Teaching".

   The following steps are explained using the teaching playback screen.

2) When the robot reaches the point where you want to use a general-purpose output, temporarily stop the operation and press F2 (DO).

```
[EDIT-PNT-TCH](1) 50
P0   X=0.00      [mm]
 [    0.00,     0.00]
1CHG 2DO  3Y   4next
```

3) The current status of the general-purpose output appears on the screen.

   Press the function key that matches the DO number to switch the output on and off (on=1, off=0).

   If selecting DO3 to DO12, press F4 (next) a few times to change the menu display.

```
[EDIT-PNT-TCH](1) 50
DO 0=0 DO 1=0 DO 2=0
DO 3=0 DO 4=0 DO 5=0
1DO0 2DO1 3DO2 4next
```

4) Press ESC to return to step 2.

```
[EDIT-PNT-TCH](1) 50
P0   X=0.00      [mm]
 [    0.00,     0.00]
1CHG 2DO  3Y   4next
```

7-5 Manual Release of Holding Brake

The holding brake on the vertical type robot can be released. Since the movable part will drop when the brake is released, attaching a stopper to protect the tool tip from being damaged is recommended.

1) Use the same procedure up to step 4 in "7-3 Direct Teaching".

2) Press F4 (next) to change the function display, then press F1 (BRK).

```
[EDIT-PNT-DTCH]
P0   X=0.00      [mm]
 [    0.00,     0.00]
1BRK 2PLT 3Y   4next
```

3) The axis selection screen appears to release the brake.

   To release the brakes on all axes, press F1 (ALL). To release the brake on the X-axis only, press F2 (X). To release the brake on the Y-axis only, press F3 (Y).

```
[EDIT-PNT-DTCH]
select axis for
brake control
1ALL 2X   3Y
```

4) A confirmation message appears asking whether to release the brake.

   To release the brake, press F1 (yes).

   To cancel releasing the brake, press F2 (no).

```
[EDIT-PNT-DTCH]
take off the brake ?

1yes 2no
```

5) The screen returns to step 2.

   The brake stays released until F1 (BRK) is pressed again or the robot servo is turned on.

```
[EDIT-PNT-DTCH]
P0   X=0.00      [mm]
 [    0.00,     0.00]
1BRK 2PLT 3Y   4next
```

7-6 Deleting Point Data

1) Use the same procedure up to step 2 in "7-1 Manual Data Input".

2) Press F4 (DEL).

```
[EDIT-PNT]
select menu


1MDI 2TCH 3DTCH4DEL
```

3) Enter the point number at the start to delete point data with the number keys and press ➡.

```
[EDIT-PNT-DEL]
DEL range P_  -P
(point No) 0→999
```

4) Enter the point number at the end to delete point data with the number keys and press ➡.

```
[EDIT-PNT-DEL]
DEL range P100-P_
(point No) 0→999
```

5) A confirmation message appears asking whether to delete the data.
   To delete the data, press F1 (yes).
   To cancel the deletion, press F2 (no).

```
[EDIT-PNT-DEL]
DEL range P100-P110
delete OK ?
1yes 2no
```

6) When the point data has been deleted, the screen returns to step 2.

```
[EDIT-PNT]
select menu


1MDI 2TCH 3DTCH4DEL
```

7-7 Editing the Pallet Data

A matrix coordinates definition of a palletizing program is used for editing. When a pallet number of the matrix is input, points are switched automatically. This function is convenient for editing pallet data.
See the robot language "8-4-17 MAT" about the relation of the pallet number and point number which are input as the 4-corner coordinates of a matrix.

1) Use the same procedure up to step 3 in "7-1 Manual Data Input", or up to step 3 in "7-2 Teaching Playback", or up to step 4 in "7-3 Direct Teaching".

   The following steps are explained using the manual data input screen.

2) In the case of manual data input, press F2 (PLT).

   In the case of teaching playback or direct teaching, press F4 (next) to change the function display, then press F2 (PLT).

```
[EDIT-PNT-MDI]
P0   X=0.00        [mm]
 [0.00    ,0.00     ]
1CHG 2PLT
```

3) Enter the desired pallet number with number keys and then press ➡.

```
[EDIT-PNT-MDI]
input pallet No _
(pallet No) 0→31
```

4) The screen shows the point number for the matrix reference coordinates (row 1, column 1) of the specified pallet.

   Enter the coordinate data according to the respective input methods for MDI, TCH, DTCH.

   After entering the data, press the STEP UP key to scroll the point numbers. Then enter the coordinate data of the remaining 3 corners.

```
[EDIT-PNT-MDI]
P251 X=            [mm]
 [_        ,          ]
1CHG 2PLT
```

7-8 Tracing Points (Moving to a registered data point)

The robot can be moved to the position specified by a registered data point. You can check the input point data by actually moving the robot.

1) Use the same procedure up to step 5 in "7-2 Teaching Playback".

2) Press F4 (next) to change the menu display and then press F1 (TRC).

```
[EDIT-PNT-TCH](1)100
P10  X=350.00    [mm]
 [    0.00,     0.00]
1CHG 2DO  3Y   4next
```

3) The coordinate data of the movement destination and the movement speed appear.
   To move the robot in the X and Y axis directions, press F1(XY).
   To move in the X-axis direction only, press F2(X).
   To move in the Y-axis direction only, press F3(Y).
   The movement speed will be 10% of the value (speed parameter) displayed on the upper right of the screen.

```
[EDIT-PNT-TCH](1)100
 [  350.00,  250.00]
trace by VEL10% OK?
1XY  2X   3Y
```

4) A confirmation message appears.
   To move the robot, press F1(yes).
   To cancel moving the robot, press F2(no).

```
[EDIT-PNT-TCH](1)100
 [  350.00,  250.00]
trace by VEL10% OK?
1yes 2no
```

5) When the movement is completed, the screen returns to step 2.

```
[EDIT-PNT-TCH](1)100
P10  X=350.00    [mm]
 [  350.00,  250.00]
1TRC 2PLT 3Y   4next
```

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***Robot movement speed is limited to 3% or less of maximum speed in "SERVICE mode state" when the robot movement speed limit is enabled.***
- ***If the hold-to-run function is enabled, robot movement stops upon releasing F1 (yes) in "SERVICE mode state". (You must hold down F1 (yes) in step 4 until the robot reaches the target point.)***

MEMO

Chapter 8 ROBOT LANGUAGE

This chapter explains the robot language. It describes what kind of commands are available and what they mean.
The DRCX series uses the YAMAHA robot language. This is an easy-to-learn BASIC-like programming language. Even a first-time user can easily create programs to control complex robot and peripheral device movements.
This robot language is an upper version of the DRC, DRCA and DRCH series robot language but fully compatible with it, so that the previous programs used with these conventional controllers can be easily upgraded.
At the beginning of this chapter, you will find a convenient table of robot language commands. At the end of this chapter, sample programs are listed for just your reference.

8-1 Robot Language Table

Instruction	Description and Format
MOVA	Moves to point data position. MOVA <point number>, <maximum. speed>
MOVI	Moves from current position by amount of point data. MOVI <point number>, <maximum. speed>
MOVF	Moves until specified DI input is received. MOVF <point number>, <DI number>, <DI status>
JMP	Jumps to a specified label in a specified program. JMP <label number>, <program number>
JMPF	Jumps to a specified label in a specified program according to the input condition. JMPF <label number>, <program number>, <input condition>
JMPB	Jumps to a specified label when general-purpose input or memory input is in the specified state. JMPB <label number>, <DI or MI number>, <input status>
L	Defines the jump destination for a JMP or JMPF statement, etc. L <label number>
CALL	Runs another program. CALL <program number>, <number of times>
DO	Turns general-purpose output or memory output on and off. DO <DO or MO number>, <output status>
WAIT	Waits until a general-purpose input or memory input is set in the specified state. WAIT <DI or MI number>, <input status>
TIMR	Waits the specified amount of time before advancing to the next step. TIMR <time>
P	Defines a point variable. P <point number>
P+	Adds 1 to a point variable. P+
P-	Subtracts 1 from a point variable. P-
SRVO	Turns the servo on or off for all axes or a specified axis. SRVO <servo status> [,<axis>]
STOP	Temporarily stops program execution. STOP
MAT	Defines a matrix. MAT <number of rows>, <number of columns>, <pallet number>
MSEL	Specifies a matrix to move. MSEL <pallet number>
MOVM	Moves to a specified pallet work position on matrix. MOVM <pallet work position>, <maximum speed>
JMPC	Jumps to a specified label when counter array variable C equals the specified value. JMPC <label number>, <counter value>
JMPD	Jumps to a specified label when counter variable D equals the specified value. JMPD <label number>, <counter value>
CSEL	Specifies the array element of counter array variable C. CSEL <array element number>
C	Defines counter array variable C. C <counter value>
C+	Adds a specified value to counter array variable C. C+ [<addition value>]
C-	Subtracts a specified value from counter array variable C. C- [<subtraction value>]
D	Defines counter variable D. D <counter value>
D+	Adds a specified value to counter variable D. D+ [<addition value>]
D-	Subtracts a specified value from counter variable D. D- [<subtraction value>]
ORGN	Executes return-to-origin on all axes or a specified axis. ORGN [<axis>]
ACHA	Defines an arch motion by setting a position. ACHA <axis>, <position>
ACHI	Defines an arch motion by setting a distance. ACHI <axis>, <distance>
DRVA	Moves a specified axis to a specified point data position. DRVA <axis>, <point number>, <max. speed>
DRVI	Moves a specified axis a distance equal to specified point data from current position. DRVI <axis>, <point number>, <maximum speed>
SHFT	Shifts the coordinate position by amount of specified point data. SHFT <point number>
TON	Runs a specified task. TON <task number>, <program number>, <start type>
TOFF	Stops a specified task. TOFF <task number>
JMPP	Jumps to a specified label when the axis positional relation meets the specified conditions. JMPP <label number>, <axis position condition>
MOVL	Executes linear interpolation motion. MOVL <point number>, <maximum speed>
MOVC	Executes circular interpolation motion. MOVC <point number>, <maximum speed>, <locus type>

Values in brackets [ ] can be omitted.

8-2 Robot Language Syntax Rules

8-2-1 Command statement format

The robot language command statement format for the DRCX controller is as follows. When creating a program using the TPB, each command statement can be automatically entered in this format, so you do not have to be aware of this format while creating the program.

```
<operation code>    [<operand 1>][,<operand 2>][,<operand 3>] [;<comment>]
```

- A command statement is basically composed of an operation code and an operand. Depending on the command statement, either no operand is used, or up to three operands are used.

  A comment can be written following the operand. (But, no comment can be written with the TPB.) A line consisting of only a comment cannot be created.

  Items in [ ] (brackets) can be omitted.

- A command statement must be entered with one-byte characters (alphanumeric characters, special characters) except for comment. Input characters can be upper case or lower case. The controller automatically converts the input characters to upper case.

- One command statement must be described within one line. It cannot be written over multiple lines. Multiple command statements cannot be described on one line. Up to 80 one-byte characters (including carriage line return) can be described on one line.

- One or more spaces must be inserted between the operation code and the operand.

- Operands enclosed in < > marks must be specified by the user. Check the description of each robot language and enter the appropriate data. (Refer to "8-4 Robot Language Description".)

- When two or more operands are entered, insert a comma (,) between them.

- Any entry after a semicolon (;) is recognized as a comment. When creating a program using a PC (personal computer), a comment is helpful to easily identify the program. Note, however, that the comment is not stored in the controller. A comment can be any number of characters as long as it is within one line. Characters that can be used as a comment are one-byte characters (alphanumeric characters, special characters) and two-byte characters (full space characters).

8-2-2 Variables

Variable are used in a program to hold data. The following variables can be used with the DRCX controller.

■ Point variable P

A point variable can contain a point number. It is used in movement commands such as MOVA and MOVI statements instead of specifying the point number directly. Sometimes the number of program steps can be reduced by using point variables.

■ Counter array variable C, Counter variable D

A counter variable can contain counter values and is used to specify the pallet work position number in a palletizing program and to count the number of runs. A counter array variable is an array of a total of 32 counter variables that can be selected by the CSEL statement of robot language.

■ Flag variable: memory input/output 100 to 147

A flag variable can only have a data value of 1 (ON) or 0 (OFF). It is used in a multi-task program to synchronize between tasks or in a condition judgement program.

Memory I/O from 100 to 131 can be freely turned on or off by the user or their values can be referenced. However, outputs 132 to 147 are controlled by the system so the user can only refer to their values.

Memory I/O description

Type	Memory I/O No.	Meaning
General-purpose	100 to 131	Memory I/O available to the user The user can freely set this with a DO statement.
Dedicated	132	Task 0 (main task) status Always set to 1.
	133	Task 1 status 1: Task has started. 0: Task has ended or has not yet started.
	134	Task 2 status 1: Task has started. 0: Task has ended or has not yet started.
	135	Task 3 status 1: Task has started. 0: Task has ended or has not yet started.
	136 to 139	Reserved for system use (Always set to 0.)
	140	X-axis hold status 1: Hold 0: Non hold
	141	Y-axis hold status 1: Hold 0: Non hold
	142 to 143	Reserved for system use (Always set to 0.)
	144	X-axis constant movement status 1: Constantly moving 0: Accelerating, decelerating or in stop
	145	Y-axis constant movement status 1: Constantly moving 0: Accelerating, decelerating or in stop
	146 to 147	Reserved for system use (Always set to 0.)

8-3 Program Function

8-3-1 Multi-task function

A multi-task function allows simultaneous executing two or more programs (tasks). The DRCX controller can execute a maximum of 4 programs at the same time.
Since the multi-task function simultaneously executes two or more programs, the following processing can be performed.

- Other processing can be performed during robot movement.

  For example, a general-purpose output can be turned on or off while a robot movement command such as MOVA or MOVI statement is being executed. This reduces the cycle time.

- Each axis can be controlled independently.

  The multi-task function allows each axis of a Multi-Flip robot to operate independently. Since the program can be created for each axis, it will be very simple.

A multi-task program can be written by the same method as normal programs. A TON statement used as the task start command is written in the main program and the subtask program is registered as another program number. When the TON command is processed during the program execution, the subtask starts to perform multiple tasks. The subtask will end when its last step has been executed or the TOFF command is issued.

Each task and data have the following relation.

The following data is independent of each task:
Point variable P, coordinate shift amount specified with the SHFT command, definition amount of arch motion, element designation of the counter array variable C specified with the CSEL command, and pallet designation number specified with the MSEL command.

The following data is shared with each task:
Point data, general-purpose input, general-purpose output, memory input/output, counter array variable C, counter variable D, pallet definition data.

⚠ ***CAUTION***

In addition to the tasks (up to 4 tasks) specified by the user, the system task starts inside the controller, so a maximum of 5 tasks are executed.
In general, the multi-task is defined as a function that simultaneously executes two or more programs (tasks). Strictly speaking, if the CPU is one unit, it executes two or more programs (tasks) while switching between them in an extremely short time almost as if they were being simultaneously executed. The DRCX controller uses this multi-task function to perform multiple tasks while switching the programs within a very short time (5ms maximum). Because of this, if 4 tasks are executed with this function, there is a maximum time of 20ms during which no processing is performed on one task. So the user must take this time into account when designing a system having multi-task functions.

8-3-2 Limitless movement function

The limitless movement function allows multiple turns in the same direction along the robot axis. The DRCX controller incorporates the soft limit function that prohibits any robot motion which exceeds the soft limits specified by the parameters. This soft limit function is very useful for linear movement type robots such as FLIP-X series. However, this function is sometimes undesirable for rotary type robots such as FROP because it limits multi-turn movements in the same direction. In such cases, the limitless movement function will prove useful.

To enable the limitless movement function, set the position data unit parameters (Setting PRM21 to 2 is suitable for applications using servo conveyors, and setting to 3 is suitable for applications using FROP or index tables.) to 2 or 3. When these parameters are set to 0 or 1, the limitless movement function does not work. In other words, the limitless movement function works only for the axis with the position data unit parameter set to 2 or 3. It is therefore possible to allow limitless movement on only one axis and perform normal movement on the other axis. Of course, two axes can be operated with the limitless movement function. Setting the position data unit parameters to 2 is suitable for applications using servo conveyors, and setting them to 3 is suitable for applications using FROP or index tables.

■ When the position data unit parameter is set to 2:

When this parameter is set to 2, the current position is expressed in millimeters from 0 to the "plus soft limit - 0.01mm" as a basic cycle. Therefore, even if the robot moves to the plus soft limit point, that position sets to 0mm so that the robot can move continuously in the same direction.

In limitless movement, the movement direction can also be selected with a movement command *1) such as MOVA which specifies a target position.

To select a movement direction opposite the return-to-origin direction, add 5000mm to the target point for point setting. To select the same movement direction as when performing return-to-origin, add 5000mm to the target point and give a minus sign to this value. When the movement direction is not specified, the robot moves in the direction of shorter distance. For example, when just 30mm is specified for the point setting, the movement direction differs depending on the current position. However, when 5030mm is specified, the robot always moves in the direction opposite the return-to-origin direction. In contrast, when -5030mm is specified, the robot always moves in the same direction as when performing return-to-origin.

In the case of a movement command such as MOVI which specifies the amount of movement, the movement direction is determined by the plus/minus sign of the point data, just as with normal movement.

⚠ ***CAUTION***

- ***When the parameter is 2, set the plus soft limit to always be an integer multiple of the lead equivalent value.*2) If it is not a value multiplied by an integer, positioning at the desired point may sometimes be impossible. The + soft limit setting range is 1 to 4999.***
- ***The maximum distance of one movement is a distance equal to one cycle (+ soft limit). To move a distance longer than one cycle, divide the movement distance into two or more portions.***
- ***In limitless movement, @XINC (@XDEC) or @YINC (@YDEC) allows moving a distance equal to one cycle. The movement speed setting and stop method are just the same as for normal movement.***
- ***If the target point is the same as the current position on the program when executing a movement command such as MOVA which specifies a position, the robot motion differs depending on whether the movement direction is selected by point setting, as follows:***
  When the movement direction is selected: Moves a distance equal to one cycle in the selected direction and stops.
  When no movement direction is selected: Does not move.
- ***Limitless movement cannot be performed in combination with the ACHA or ACHI statement.***
- ***Use caution when operating the robot since the soft limits are disabled during limitless movement.***

*1) These movement commands include MOVA, MOVF, MOVD, MOVM, DRVA and DRVD statements. The MOVD and DRVD statements are provided only for communication commands and can directly specify the target point.

*2) The lead equivalent value can be checked with the lead length parameters (PRM60, PRM100). The lead length parameter indicates a distance the robot axis (or workpiece on the axis) moves while the motor makes one turn, in units of 1/100mm.

■ When the position data unit parameter is set to 3:

When this parameter is set to 3, the current position is expressed in degrees (°) from 0 to 359.99 as a basic cycle. Therefore, even if the robot moves to the 360° point, that position sets to 0° (=360°) so that the robot can rotate continuously in the same direction.

In limitless movement, the rotation direction can also be selected with a movement command *1) such as MOVA which specifies a target position.

To select the rotation direction opposite the return-to-origin direction, add 5000° to the target point for point setting. To select the same rotation direction as when performing return-to-origin, add 5000° to the target point and give a minus sign to this value. When the rotation direction is not specified, the robot moves in the direction of shorter distance. For example, when just 30° is specified for the point setting, the rotation direction differs depending on the current position. However, when 5030° is specified, the robot always rotates in the direction opposite the return-to-origin direction. In contrast, when -5030° is specified, the robot always rotates in the same direction as when performing return-to-origin.

In the case of a movement command such as MOVI which specifies the amount of movement, the movement direction is determined by the plus/minus sign of point data, just as with normal movement.

⚠ ***CAUTION***

- ***The maximum distance of one movement is a distance equal to one cycle (360°). To move a distance longer than one cycle, divide the movement distance into two or more portions.***
- ***In limitless movement, @XINC (@XDEC) or @YINC (@YDEC) allows moving a distance equal to one cycle. The movement speed setting and stop method are just the same as for normal movement.***
- ***If the target point is the same as the current position on the program when executing a movement command such as MOVA which specifies a position, the robot motion differs depending on whether the rotation direction is selected by point setting, as follows:***
  When the rotation direction is selected: Rotates through 360° in the selected direction and stops.
  When no rotation direction is selected: Does not move.
- ***Limitless movement cannot be performed in combination with the ACHA or ACHI statement.***
- ***Use caution when operating the robot since the soft limits are disabled during limitless movement.***

*1) These movement commands include MOVA, MOVF, MOVD, MOVM, DRVA and DRVD statements. The MOVD and DRVD statements are provided only for communication commands and can directly specify the target point.

8-4 Robot Language Description

8-4-1 MOVA

Function: Moves to a point specified by a point number (Moves to an absolute position relative to the origin point).
Format: MOVA <point number>, <maximum speed>
Example: MOVA 51, 80
Moves to P51 at speed 80.
Explanation: This command moves the robot to a position on the absolute coordinates whose origin position is defined as 0.
The robot starts moving when all axes enter the positioning-completed pulse range, and stops when all axes reach the OUT valid position.
(1) Point number
The point number is a number designated to each point from 0 to 999, a total of 1000 points, and used to create point date in PNT (point) mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (See "8-4-12 P".)
(2) Maximum speed
The maximum speed can be set to any level between 1 and 100. If the execution speed in OPRT mode is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).

8-4-2 MOVI

Function: Moves a distance specified by a point number from the current position.
Format: MOVI <point number>, <maximum speed>
Example: MOVI 10, 80
Moves an amount equal to point data P10 from the current position at speed 80.
Explanation: This command moves the robot on the relative coordinates with the current position viewed as 0.
The robot starts moving when all axes enter the positioning-completed pulse range, and stops when all axes reach the OUT valid position.
(1) Point number
The point number is a number designated to each point from 0 to 999, a total of 1000 points, and used to create point date in PNT (point) mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (See "8-4-12 P".)
(2) Maximum speed
The maximum speed can be set to any level between 1 and 100. If the execution speed in OPRT mode is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).

8-4-3 MOVF

Function: Moves until a specified DI number input is received.
Format: MOVF <point number> <DI number> <DI status>
Example: MOVF 1, 2, 1
The robot moves toward P1 and stops when DI2 turns on. Program execution then proceeds to the next step.
Explanation: This is used when searching for a target position using sensors or other devices.
The robot starts moving when all axes enter the positioning-completed pulse range, and stops when the DI conditions are met. Even if the DI conditions are not met, this command terminates when the robot reaches the specified point and proceeds to the next step.
(1) Point number
The point number is a number designated to each point from 0 to 999, a total of 1000 points, and used to create point date in PNT (point) mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (See "8-4-12 P".)
(2) DI number
Specify one of the 16 (0 to 15) general-purpose inputs.
(3) DI status
"1" means "on" and "0" means "off".
Other:
- The robot speed during execution of the MOVF movement can be specified by PRM4. (Refer to "PRM4: MOVF speed") Note that this will not be affected by the OPRT mode execution speed.

8-4-4 JMP

Function: Jumps to a specified step in a specified program.
Format: JMP <label number>, <program number>
Example: JMP 10, 8
Jumps to label 10 in program 8.
Explanation: This command is used to control the flow of program execution.
(1) Label number
The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L".)
(2) Program number
The program number is a number used to identify the 100 individual programs from 0 to 99.
Other:
- Even when the program number is changed by the JMP statement, resetting it will return to the original program number when program execution begins.

8-4-5 JMPF

Function: If the conditional jump input matches the setting value, program execution jumps to a specified label in a specified program.

Format: JMPF <label number>, <program number>, <input condition value>

Example: JMPF 12, 3, 5

If the conditional jump input is 5, program execution jumps to label 12 in program 3. If the jump input is not 5, program execution advances to the next step.

Explanation: This command is used to control the flow of program execution according to the conditional jump input.

(1) Label number

The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L".)

(2) Program number

The program number is a number used to identify the 100 individual programs from 0 to 99.

(3) Input condition value

This is the condition used to make a jump. A general-purpose input is viewed as binary code input, and if it meets this input condition value, a jump is executed.

The number of points that can be branched under the input condition depends on the number of conditional input points which is set with PRM0. (See "PRM0: No. of conditional input points".)

CAUTION

Select a number of conditional input points that is large enough to accommodate the actual number of input conditions to be used. If an error is made in setting the number of conditional input points, there will be a discrepancy between the input condition value required by the program and that recognized by the controller. This could keep the program from operating properly.

General-purpose input status and input condition value when the number of conditional input points is 4 (input range 0 to 15)

General use DI Input				Input Condition Value
DI3 (2^3)	DI2 (2^2)	DI1 (2^1)	DI0 (2^0)	
OFF	OFF	OFF	OFF	0
OFF	OFF	OFF	ON	1
OFF	OFF	ON	OFF	2
OFF	OFF	ON	ON	3
OFF	ON	OFF	OFF	4
OFF	ON	OFF	ON	5
OFF	ON	ON	OFF	6
OFF	ON	ON	ON	7
ON	OFF	OFF	OFF	8
ON	OFF	OFF	ON	9
ON	OFF	ON	OFF	10
ON	OFF	ON	ON	11
ON	ON	OFF	OFF	12
ON	ON	OFF	ON	13
ON	ON	ON	OFF	14
ON	ON	ON	ON	15

8-4-6 JMPB

Function: Jumps to a specified label when a specified general-purpose input or memory input is ON or OFF.
Format: JMPB <label number>, <DI or MI number>, <input status>
Example: JMPB 12, 8, 1
Jumps to label 12 when DI8 input is ON.
If DI8 is OFF, the program execution proceeds to the next step.
Explanation: This command controls the program flow according to the general-purpose input or memory input status.
(1) Label number
The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L".)
(2) DI or MI number
Specify one of the general-purpose input numbers from 0 to 15 (16 points) or memory input numbers from 100 to 147 (48 points).
(3) Input status
"1" means "on" and "0" means "off".

8-4-7 L

Function: Defines the jump destination for JMP, JMPF or JMPB statements, etc.
Format: L <label number>
Example: L 100
Defines label 100.
Explanation: This command is used to define the destination to which program execution jumps with a jump command. The label number may be any number between 0 and 255. The same label numbers may be used if they are in different programs.

8-4-8 CALL

Function: Calls and executes another program.

Format: CALL <program number>, <number of times>

Example: CALL 5, 2

Calls program 5 and executes it twice. Program execution then proceeds to the next step.

Explanation: When repeating the same operation a number of times, the CALL statement is used as needed to call and execute the subroutine defined as a separate program.

(1) Program number
The program number is a number used to identify the 100 individual programs from 0 to 99.

(2) Number of times
This is the number of times that the program is to be repeated. This can be specified from 1 to 255.

Other:
- The nesting level is 6.
- When the end of the program initiated by the CALL statement is detected, program execution advances to the step following the CALL statement in the main program.
- An error occurs and program execution stops if the program being executed is called by the CALL statement.
- Even when the program number is changed by the CALL statement, resetting it will return to the original program number when program execution begins.
- An error "stack overflow" might occur if a jump is made to another program by the JMP or JMPF statement in a program called as a subroutine by the CALL statement.

8-4-9 DO

Function: Controls ON/OFF of general-purpose output or memory output.

Format: DO <DO or MO number>, <output status>

Example: DO 3, 1

Turns on DO3.

Explanation: This command turns the general-purpose output or memory output on and off.

(1) DO or MO number
Specify one of the general-purpose output numbers from 0 to 12 (13 points) or memory output numbers from 100 to 131 (32 points).

(2) Output status
"1" means "on" and "0" means "off".

8-4-10 WAIT

Function: Waits until a specified general-purpose input or memory input changes to a specified state.

Format: WAIT <DI or MI number>, <input status>

Example: WAIT 5, 1

Waits until DI5 turns on.

Explanation: This command adjusts the timing according to the general-purpose input or memory input state.

(1) DI or MI number

Specify one of the general-purpose input numbers from 0 to 15 (16 points) or memory input numbers from 100 to 147 (48 points).

(2) Input status

"1" means "on" and "0" means "off",

8-4-11 TIMR

Function: Waits for a specified amount of time before advancing to the next step.

Format: TIMR <time>

Example: TIMR 100

Moves to the next step after waiting one second.

Explanation: This command is used when adjusting the time within the program. Time may be specified in lengths from 1 to 65535, in units of 10ms. In other words, time may be specified from 0.01 seconds up to 655.35 seconds.

8-4-12 P

Function: Sets a point variable P.
Format: P <point number>
Example: P 200
Sets a point variable P to 200.
Explanation: The point variable can contain a point number as a variable, which can be from 0 to 999. By using a movement command such as MOVA with a P+ or P- statement, the number of steps required to create a repeating program can be reduced.
Other:
- The contents of point variable P are retained even when the controller power is turned off, but when the program is reset or when the program reset is applied for example by switching the execution program, the point variable P will be initialized to 0.
- Point variables P in a task are independent of those in other tasks. For example, the definition and edited contents of a point variable used in task 1 do not affect the point variable used in task 0.

8-4-13 P+

Function: Adds 1 to a point variable P.
Format: P+
Example: P+
Adds 1 to a point variable P. (P←P+1)
Explanation: Adds 1 to a point variable P.

8-4-14 P-

Function: Subtracts 1 from a point variable P.
Format: P-
Example: P-
Subtracts 1 from a point variable P. (P←P-1)
Explanation: Subtracts 1 from a point variable P.

8-4-15 SRVO

Function: Turns the servo of a specified axis on or off.
Format: SRVO <servo status> [,<axis>]
Example: SRVO 1, 1
Turns on the servo of X-axis.
SRVO 0
Turns off the servo of all axes.
Explanation: This command is used to prevent an overload on the motor that may occur if the robot is locked mechanically after positioning is completed. This command is executed after the specified axis enters the positioning-completed pulse range.
(1) Servo status
"1" means "on" and "0" means "off."
(2) Axis
"0" means all axes, "1" means the X-axis, and "2" means the Y-axis. If this setting is omitted, all axes are selected.

8-4-16 STOP

Function: Temporarily interrupts program execution.
Format: STOP
Example: STOP
Temporarily interrupts program execution.
Explanation: This command temporarily interrupts execution of a program. If two or more tasks are being executed, then all those tasks are interrupted. This command can be used at any point in a program. The next execution will restart from the subsequent step.
Others:
- Normally, the program terminates when the last step is detected. At the same time, the program is reset and the execution step number will return to 1 (top line of the program).
- To interrupt only a subtask temporarily without stopping the main task, use the TOFF statement. (Refer to "8-4-36 TOFF".)

8-4-17 MAT

Function: Defines the number of rows and columns of the matrix.

Format: MAT <number of rows>, <number of columns>, <pallet number>

Example: MAT 3, 6, 0

Defines a matrix of 3 × 6 on pallet number 0.

Explanation: This command defines a matrix for palletizing movement. A palletizing program can be easily created by using this command with MSEL or MOVM, etc.

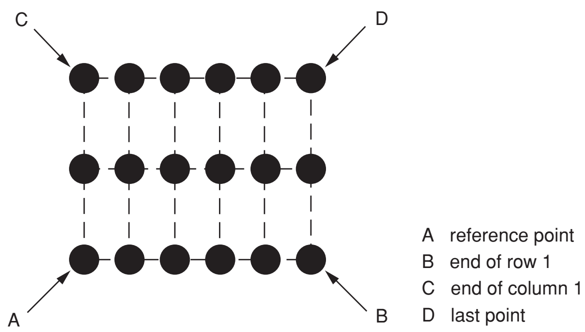
(1) Number of rows
Set any value from 1 to 255.

(2) Number of columns
Set any value from 1 to 255.

(3) Pallet number
This number is used for matrix identification and can be set from 0 to 31. A total of 32 matrix data can be handled.

Others:

- A common method for matrix coordinate definition specifies only the positions of the 4 corners of the matrix by 4-point teaching. The remaining points are then found by calculation. When teaching the positions of the 4 corners in PNT (point) mode to create point data, the point numbers are generally specified as follows: If pallet number is "n" for instance, enter the position of the reference point (row 1, column 1) in p(251-4n), the position at the end of row 1 in p(252-4n), the position at the end of column 1 in p(253-4n), and the position of the remaining corner in p(254-4n). To define a one-dimension matrix such as "row 1, column m", enter the position of the reference point (row 1, column 1) in p251, and the position of last point (row 1, column m) in p252. You do not have to enter any data in p253 and p254 (when pallet number is 0).
- The matrix definition contents are shared with each task.



Matching point numbers for inputting pallet numbers and coordinate values A to D

Pallet No.	0	1	n	31
A	p251	p247	p(251-4n)	p127
B	p252	p248	p(252-4n)	p128
C	p253	p249	p(253-4n)	p129
D	p254	p250	p(254-4n)	p130

8-4-18 MSEL

Function: Specifies a matrix where the robot moves with a MOVM statement.

Format: MSEL <pallet number>

Example: MSEL 0

Points where the robot moves with a MOVM statement are calculated based on matrix data of pallet number 0.

Explanation: This command selects a matrix and is always used with a MOVM statement as a pair.

(1) Pallet number

This number is used for matrix identification and can be set from 0 to 31.

Others:

- The pallet number assigned with the MSEL statement is independent of each task. For example, when different pallet numbers are assigned to task 0 and task 1, then task 0 and task 1 execute the MOVM statement based on different pallet data.

8-4-19 MOVM

Function: Moves to a point on the specified matrix.

Format: MOVM <pallet work position>, <maximum speed>

Example: MOVM 23, 100

Moves to the point at row 3, column 7 at speed 100 when a matrix of 5 × 8 is defined by the MAT statement.

Explanation: This command moves the robot to each point on a matrix specified by the MSEL statement.

This command allows the robot to start moving when all axes are within the "positioning-completed pulse" range. This command ends when all axes enter the OUT valid position.

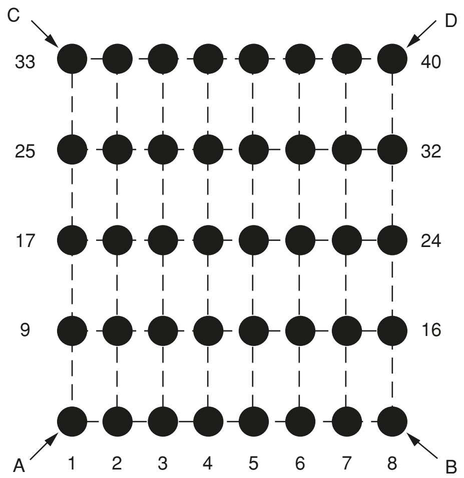
(1) Pallet work position

The pallet work position is a number used to identify each point on a matrix, and can be from 1 to 65025 (=255 × 255). For example, on a "row M, column N" matrix, the pallet work position at "row A, column B" is found by (A–1) × N+B. When a character "C" or "D" is entered here for special use, a counter variable is set in each pallet work position.

(2) Maximum speed

The maximum speed can be set to any level between 1 and 100. If the execution speed in OPRT mode is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).

Others:
- The MOVM statement performs calculation on the assumption that the robot operates on the Cartesian coordinate system. Because SCARA robots operate on a joint coordinate system, the desired motion cannot be obtained.



Example of pallet work position in 5 × 8 matrix

Work Position	Position No.
A: Reference point	1
B: End of row 1	8
C: End of column 1	33
D: Last point	40

8

ROBOT LANGUAGE

8-4-20 JMPC

Function: Jumps to a specified label when the counter array variable C matches a specified value.

Format: JMPC <label number>, <counter value>

Example: JMPC 5, 100

Jumps to label 5 when the counter array variable C is 100. Program execution proceeds to the next step except when the counter array variable C is 100.

Explanation: This command controls the program flow according to the counter array variable C. The counter array variable C to be compared is the element number specified with the CSEL statement.

(1) Label number

The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L")

(2) Counter value

Set any value from 0 to 65535.

8-4-21 JMPD

Function: Jumps to a specified label when the counter variable D matches a specified value.

Format: JMPD <label number>, <counter value>

Example: JMPD 5, 100

Jumps to label 5 when the counter variable D is 100. Program execution proceeds to the next step except when the counter variable D is 100.

Explanation: This command controls the program flow according to the counter variable D.

(1) Label number

The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L")

(2) Counter value

Set any value from 0 to 65535

8-4-22 CSEL

Function: Specifies an array element of the counter array variable C to be used.
Format: CSEL <array element number>
Example: CSEL 1
The counter array variable of element number 1 is used in the subsequent steps.
Explanation: This command designates an array element number of the counter array variable C.
The array element data designated with the CSEL statement is used in the C statement, C+ statement, C- statement, JMPC statement and MOVM statement.
(1) Array element number
This is a number used to designate the array element number of a counter array variable and can be any value from 0 to 31.
When a character "D" is entered here, the counter variable D is used to designate the element of the counter array variable.
Others:
- The array element designation is held even when the controller power is turned off, but when the program is reset or when the program reset is applied by switching the execution program, the element designation number will be initialized to 0.
- The element number designated with the CSEL statement is independent of each task. For example, when different array elements are designated for task 0 and task 1, the definition or change in the counter array variable C of task 1 does not affect task 0.

8-4-23 C

Function: Sets the counter array variable C.
Format: C <counter value>
Example: C 200
Sets the counter array variable C to 200.
Explanation: This command sets a counter value for the counter array variable specified with the CSEL statement. The counter array variable is an array variable containing 32 elements, and can be set to any value from 0 to 65535. This command can be used with a C+ or C– statement and a JMPC statement for a repeating program and also with a MOVM statement for a palletizing program.
Others:
- Counter array variable C is not initialized even if the program is reset or the controller power is turned off. To initialize, rewrite the program.
- The counter array variable C is a variable shared with all tasks. For example, task 0 and task 1 use a counter array variable with the same element number, the edited contents of task 1 affect task 0.

8-4-24 C+

Function: Adds a specified value to the counter array variable C.
Format: C+ [<addition value>]
Example: C+ 100
Adds 100 to the counter array variable C. (C←C+100)
C+
Adds 1 to the counter array variable C. (C←C+1)
Explanation: This command adds a specified value to the counter array variable C specified with the CSEL statement. The addition value can be set to any value from 1 to 65535. If the addition value is omitted, then 1 is added to the counter array variable C.

8-4-25 C-

Function: Subtracts a specified value from the counter array variable C.
Format: C- [<subtraction value>]
Example: C- 100
Subtracts 100 from the counter array variable C. (C←C-100)
C-
Subtracts 1 from the counter array variable C. (C←C-1)
Explanation: This command subtracts a specified value from the counter array variable C specified with the CSEL statement. The subtraction value can be set to any value from 1 to 65535. If the subtraction value is omitted, then 1 is subtracted from the counter array variable C.

8-4-26 D

Function: Sets the counter variable D.
Format: D <counter value>
Example: D 200
Sets the counter variable D to 200.
Explanation: The counter variable D can be set to any value by the user from 0 to 65535. This command can be used with a D+ or D– statement and a JMPD statement for a repeating program, and also with a MOVM statement for a palletizing program.
Others:
- The counter variable D is not initialized even if the program is reset or the controller power is turned off. To initialize, rewrite the program.
- The counter variable D is a variable shared with all tasks.
  For example, task 0 and task 1 use the counter variable D, the edited contents of task 1 affect task 0.

8-4-27 D+

Function: Adds a specified value to the counter variable D.
Format: D+ [<addition value>]
Example: D+ 100
Adds 100 to the counter variable D. (D←D+100)
D+
Adds 1 to the counter variable D. (D←D+1)
Explanation: This command adds a specified value to the counter variable D. The addition value can be set to any value from 1 to 65535. If the addition value is omitted, then 1 is added to the counter variable D.

8-4-28 D-

Function: Subtracts a specified value from the counter variable D.
Format: D- [<subtraction value>]
Example: D- 100
Subtracts 100 from the counter variable D. (D←D-100)
D-
Subtracts 1 from the counter variable D. (D←D-1)
Explanation: This command subtracts a specified value from the counter variable D. The subtraction value can be set to any value from 1 to 65535. If the subtraction value is omitted, then 1 is subtracted from the counter variable D.

8-4-29 ORGN

Function: Performs return-to-origin when the search method is selected as the origin detection method, or checks whether return-to-origin has been performed when the mark method is selected.

Format: ORGN [<axis>]

Example: ORGN 1
Returns only the X-axis to its origin position.
ORGN
Returns all axes to their origin positions.

Explanation: Return-to-origin is performed based on the return-to-origin parameter data when the search method is selected as the origin detection method. When the mark method is selected, this command checks whether return-to-origin has been performed and proceeds to the next step when it has been performed, but halts the operation as an error if not performed.
(1) Axis
"0" means all axes, "1" means the X-axis, and "2" means the Y-axis. If this setting is omitted, return-to-origin is performed on all axes.

Others:
- When the mark method is used for one axis and the search method for the other axis, then return-to-origin on the axis using the mark method must first be completed before performing return-to-origin on the axis using the search method. Use the TPB to perform return-to-origin on the axis using the mark method.
- Once return-to-origin is performed after the robot cable and absolute battery are connected, there is no need to repeat it even when the controller is turned off. (As an exception, return-to-origin becomes incomplete if the absolute backup function is disabled or a parameter relating to the origin is changed. Return-to-origin must be reperformed in this case.)
- When performing return-to-origin by the stroke-end detection method, do not interrupt the return-to-origin operation while detecting the origin (while contacting the mechanical limit). Otherwise, the operation will stop due to a controller overload alarm and the power will have to be turned on again.
- If return-to-origin must be repeated by the stroke-end detection method, wait at least 5 seconds before repeating it.

8-4-30 ACHA

Function: Defines an arch motion by setting a position (absolute position with respect to the origin).

Format: ACHA <axis>, <position>

Example: ACHA 2, 10

Defines an arch motion in which the robot temporarily moves to a point Y=10.00 before reaching the target position.

Explanation: This command defines an axis and a position to move the robot in an arch motion. This command is used with a MOVA, MOVI or MOVF statement.

(1) Axis

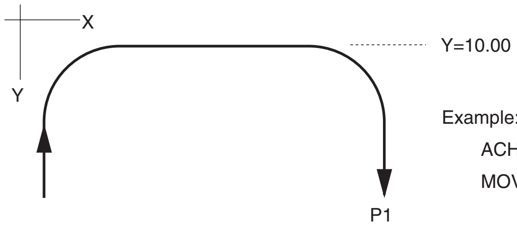
Specify the axis that performs an arch motion. "1" for the X-axis and "2" for the Y-axis.

(2) Position

This is the position (absolute position with respect to the origin) the arch motion axis moves to. The setting range is from -9999 to 9999 mm.

Others:

- The arch's arc size can be set with the arch position parameters (PRM57, PRM97).
- If an ACHA statement is executed independently, the subsequent movement commands (MOVA, MOVI or MOVM statements) are executed with an arch motion until the program is reset. Be careful when combining ACHA with conditional branch statements, etc.



Example: Motion for the following program

```
ACHA  2,10
MOVA  1,100
```

8-4-31 ACHI

Function: Defines an arch motion by setting an incremental distance (relative position with respect to the current position).

Format: ACHI <axis>, <distance>

Example: ACHI 2, –100

Defines an arch motion in which the robot temporarily moves a distance equal to Y=-100.00 before reaching the target position.

Explanation: This command defines an axis and a distance to move the robot in an arch motion. This command is used with a MOVA, MOVI or MOVF statement.

(1) Axis

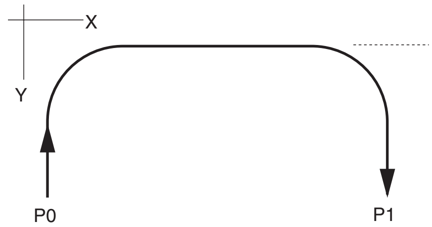
Specify the axis that performs an arch motion. "1" for the X-axis and "2" for the Y-axis.

(2) Distance

This is the distance (relative distance with respect to the current position) the arch motion axis moves. The setting range is from -9999 to 9999 mm.

Others:

- The arch's arc size can be set with the arch position parameters (PRM57, PRM97).
- If an ACHI statement is executed independently, the subsequent movement commands (MOVA, MOVI or MOVM statements) are executed with an arch motion until the program is reset. Be careful when combining ACHI with conditional branch statements, etc.



Example: Motion for the following program
(The current position is P0.)
ACHI 2,-100
MOVA 1,100

8-4-32 DRVA

Function: Moves a specified axis to a specified point data position (absolute position relative to the origin).

Format: DRVA <axis>, <point number>, <maximum speed>

Example: DRVA 1, 51, 80

Moves the X-axis to P51 at speed 80.

Explanation: This command moves the specified axis to a position on absolute coordinates whose origin is set to 0. Point data for the specified axis is used as the target position.

The motion starts when the specified axis enters the positioning completed pulse range, and stops when the axis reaches the OUT valid position.

(1) Axis

Specify the axis to be moved. "1" for the X-axis and "2" for the Y-axis.

(2) Point number

The point number is a number assigned to each point from 0 to 999 (1000 points), and used to create point data in point mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (Refer to "8-4-12 P".)

(3) Maximum speed

The maximum speed can be set to any value from 1 to 100 (100 steps). When the execution speed in OPRT mode is 100, then 100 will be equal to 3000rpm (when PRM64 and PRM104=3000).

Others:

- The point data of the axis which is not specified in the DRVA statement must be entered as dummy data. Otherwise an error will occur.

8-4-33 DRVI

Function: Moves a specified axis a distance equal to specified point data from the current position.

Format: DRVI <axis>, <point number>, <maximum speed>

Example: DRVI 2, 10, 80

Moves the Y-axis a distance equal to point data P10 from the current position at speed 80.

Explanation: This command moves the specified axis to a position on relative coordinates with the current origin viewed as 0. Point data for the specified axis is used as the distance to move.

The motion starts when the specified axis enters the positioning completed pulse range, and stops when the axis reaches the OUT valid position.

(1) Axis

Specify the axis to be moved. "1" for the X-axis and "2" for the Y-axis.

(2) Point number

The point number is a number assigned to each point from 0 to 999 (1000 points), and used to create point data in point mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (Refer to "8-4-12 P".)

(3) Maximum speed

The maximum speed can be set to any value from 1 to 100 (100 steps). When the execution speed in OPRT mode is 100, then 100 will be equal to 3000rpm (when PRM64 and PRM104=3000).

Others:

- The point data of the axis which is not specified in the DRVI statement must be entered as dummy data. Otherwise an error will occur.

8-4-34 SHFT

Function: Shifts the position data.

Format: SHFT <point number>

Example: SHFT 10

Shifts the coordinates on which the subsequent movement commands are executed, by a data amount defined by point 10.

Explanation: This command shifts position data in the subsequent movement commands to be executed, by coordinates equal to the specified point data. The shift data is valid until the SHFT statement is executed again or until the program reset is executed.

(1) Point number

The point number is a number designated to each point from 0 to 999, a total of 1000 points, and used to create point date in PNT (point) mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (Refer to "8-4-12 P".)

Others:

- When the program is reset or when the program reset is applied by switching the execution program, the shift data will be initialized to (0.00,0.00).
- The SHFT statement affects MOVA, MOVF, MOVM, DRVA, ACHA, MOVL and MOVC but does not affect MOVI, DRVI and ACHI.
- The coordinate shift amount specified with the SHFT statement is independent of each task. For example, when task 0 and task 1 are being executed, the coordinate shift of task 1 has no effect on the movement command for task 0.

```
004 :     :
005 : SHFT  1
006 : MOVA  0,100
007 :     :
```

For example, with a program shown on the left, the target position of MOVA statement in the 6th step will be the position of P0 + P1.

8-4-35 TON

Function: Executes a specified task.
Format: TON <task number>, <program number>, <start type>
Example: TON 1,2,0
Newly executes program 2 as task 1.
Explanation: This command starts multiple tasks and can be used to control the I/O signals in parallel with the axis movement and perform different processing for each axis.
(1) Task number
The task number is a number used to identify the four individual tasks from 0 to 3. Since task 0 is the main task, tasks numbers from 1 to 3 can be specified.
(2) Program number
The program number is a number used to identify the 100 individual programs from 0 to 99.
(3) Start type
This specifies whether to start a new task or suspended task. Set to 0 when executing a new task, and set to 1 when restarting a suspended task.
Others:
- A task number which is being executed cannot be specified. (A task number which has been suspended can be specified.)
- The task terminates when the last step is detected. When a subtask terminates, it does not affect operation of other tasks. But, if task 0 (main task) terminates, all other tasks in operation also terminate.

8-4-36 TOFF

Function: Suspends a specified task.
Format: TOFF <task number>
Example: TOFF 1
Suspends the program being executed as task 1.
Explanation: This command is used to suspend the execution of a particular task.
(1) Task number
The task number is a number used to identify the four individual tasks from 0 to 3. Since task 0 is the main task, tasks numbers from 1 to 3 can be specified.
Others:
- This command cannot suspend its own task.

8-4-37 JMPP

Function: Jumps to a specified label when the axis position relation meets the specified conditions.

Format: JMPP <label number>, <axis position condition>

Example: JMPP 3,21

Jumps to label 3 if the X-axis coordinate is smaller than the point specified with the point variable P but the Y-axis coordinate is larger than it.

Explanation: This command controls the program flow according to the specified position of the axis, by comparing it with the point specified with the point variable P.

(1) Label number

The label number is a number defined by the "L" statement, and indicates the destination to jump. Any number from 0 to 255 can be specified. (See "8-4-7 L".)

(2) Axis position condition

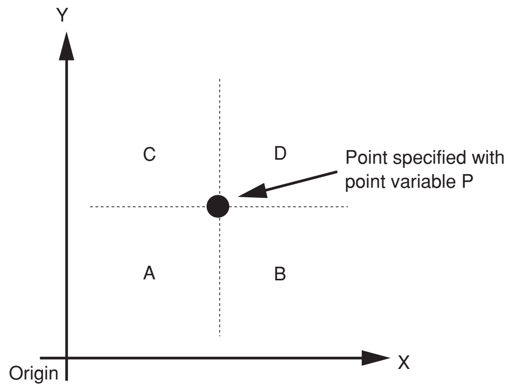
The units represent the X-axis comparison condition, and the tens represent the Y-axis comparison condition.

When set to 1, this establishes the condition that the robot should be closer to the origin than the specified position. When set to 2, this establishes the condition that the robot should be farther away from the origin than the specified position.

In the case of "0", the conditions are met whether the robot is positioned on either side of the specified coordinate.

Others:

- When the axis is at the specified coordinate position, this views that the condition is met.



Axis position condition	Robot position that meets the condition
1	The robot is in area A or C.
2	The robot is in area B or D.
10	The robot is in area A or B.
20	The robot is in area C or D.
11	The robot is in area A.
12	The robot is in area B.
21	The robot is in area C.
22	The robot is in area D.

8-4-38 MOVL

Function: Moves to the position specified by a point number (absolute position with respect to the origin) in a linear interpolation motion.

Format: MOVL <point number>,<maximum speed>

Example: MOVL 500,100
Moves to P500 at speed 100.

Explanation: This command moves the robot on absolute coordinates with the origin set to 0, while controlling the locus of linear interpolation.

(1) Point number
The point number is a number assigned to each point from 0 to 999 (1000 points), and used to create point data in point mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (Refer to "8-4-12 P".)

(2) Maximum speed
The maximum speed can be set to any value from 1 to 100 (100 steps). When the execution speed in OPRT mode is 100, then 100 will be equal to a speed of 1000mm/s (when PRM64 and PRM104 are set to 3000 and the ball screw lead is 20mm)

Others:
- The robot does not stop and continues operating when the MOVL state ment and MOVC statement are consecutively written.
- The MOVL and MOVC statements execute advanced reading (execute the steps that follow interpolation motion). Therefore, to issue the DO output after the end point of the interpolation motion is reached, the DO statement should be written after the MOVA statement following the MOVL and MOVC statements. In this case, set the specified point for MOVA to the same point as the end point of the MOVL and MOVC statements. (Specifying any speed is okay.)
  The same operation can be performed by writing WAIT 140,1 and WAIT 141, 1 instead of the MOVA statement.
- When the last movement command in the program is for interpolation motion, the DO output may be turned off during movement due to the advanced reading. If this happens, add the MOVA statement after the last interpolation motion as explained above.
- Interpolation motion cannot resume even if it is not yet finished.
  So, if interpolation motion is stopped by an interlock signal, the program should be reset and restarted from the beginning.
- Linear interpolation motion cannot be performed on SCARA robots which do not operate on a Cartesian coordinate system.

8-4-39 MOVC

Function: Performs a circular interpolation motion passing through the position specified by a point number.

Format: MOVC <point number>,<maximum speed>,<locus type>

Example: MOVC 10,100,0

Moves along a circular segment locus determined by the three points of the current position, P10 and P11. The speed is 100 and the end point is 11.

MOVC 10,100,1

Moves along a circular locus determined by the three points of the current position, P10 and P11. The speed is 100 and the end point is the same as the start point.

Explanation: This command moves the robot on absolute coordinates with the origin set to 0, while controlling the locus of circular interpolation. For example, when a circular segment is selected, if the point number is specified as n, the robot moves from the current position along a circular segment locus with the end point of point n+1, passing through point n. In the case of a circular locus, the robot moves from the current position along a circular locus with the same start and end points, passing through points n and n+1.

(1) Point number

The point number is a number assigned to each point from 0 to 999 (1000 points), and used to create point data in point mode. When a character "P" is entered here for special use, a point variable defined by the "P" statement is set as the point number. (Refer to "8-4-12 P".)

(2) Maximum speed

The maximum speed can be set to any value from 1 to 100 (100 steps). When the execution speed in OPRT mode is 100, then 100 will be equal to a speed of 1000mm/s (when PRM64 and PRM104 are set to 3000 and the ball screw lead is 20mm)

(3) Locus type

Select the locus type. "0" for the circular segment locus and "1" for the circular locus.

Others:

- The movable radius is 1000mm maximum and 2mm minimum.
- The robot does not stop and continues operating when the MOVL state ment and MOVC statement are consecutively written.
- The MOVL and MOVC statements execute advanced reading (execute the steps that follow interpolation motion). Therefore, to issue the DO output after the end point of the interpolation motion is reached, the DO statement should be written after the MOVA statement following the MOVL and MOVC statements. In this case, set the specified point for MOVA to the same point as the end point of the MOVL and MOVC statements. (Specifying any speed is okay.)
  The same operation can be performed by writing WAIT 140,1 and WAIT 141, 1 instead of the MOVA statement.
- When the last movement command in the program is for interpolation motion, the DO output may be turned off during movement due to the advanced reading. If this happens, add the MOVA statement after the last interpolation motion as explained above.
- Interpolation motion cannot resume even if it is not yet finished.
  So, if interpolation movement is stopped by an interlock signal, the program should be reset and restarted from the beginning.
- Circular interpolation motion cannot be performed on SCARA robots which do not operate on a Cartesian coordinate system.

8-5 Sample Programs

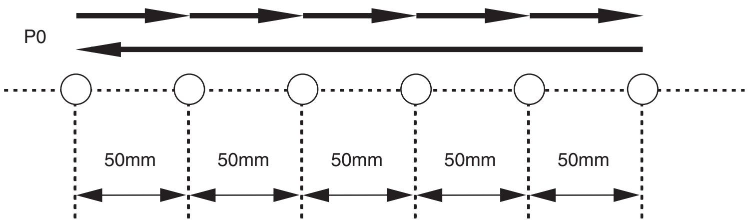
8-5-1 Moving between two points

Program **Comment**

[NO0]

```
001:L      0          ; Label definition
002:MOVA   1,   100   ; Moves to P1
003:MOVA   2,   100   ; Moves to P2
004:TIMR   100        ; Delays  for one second
005:JMP    0,   0     : Returns to L0
```

8-5-2 Moving at an equal pitch



Point

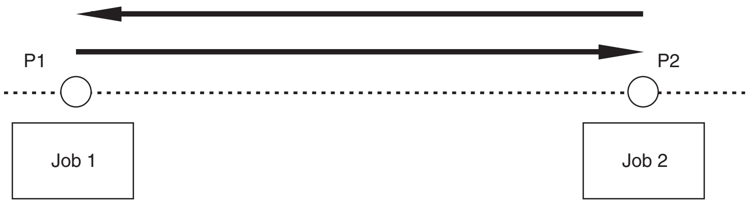
P0	Start position
P1	Movement distance: 50mm

Program **Comment**

[NO0]

```
001:L      0          ; Label definition
002:MOVA   0,   100   ; Moves to P0
003:MOVI   1,   100   ; Moves five times at a 50mm pitch
004:MOVI   1,   100   ; Moves five times at a 50mm pitch
005:MOVI   1,   100   ; Moves five times at a 50mm pitch
006:MOVI   1,   100   ; Moves five times at a 50mm pitch
007:MOVI   1,   100   ; Moves five times at a 50mm pitch
008:JMP    0,   0     ; Returns to L0
```

8-5-3 Positioning 2 points and sending job commands to a PLC at each position



Point

P1	Position at which job 1 is complete
P2	Position at which job 2 is complete

General-purpose input

DI1	Job 1 completion	1: Complete	0: Not complete
DI2	Job 2 completion	1: Complete	0: Not complete

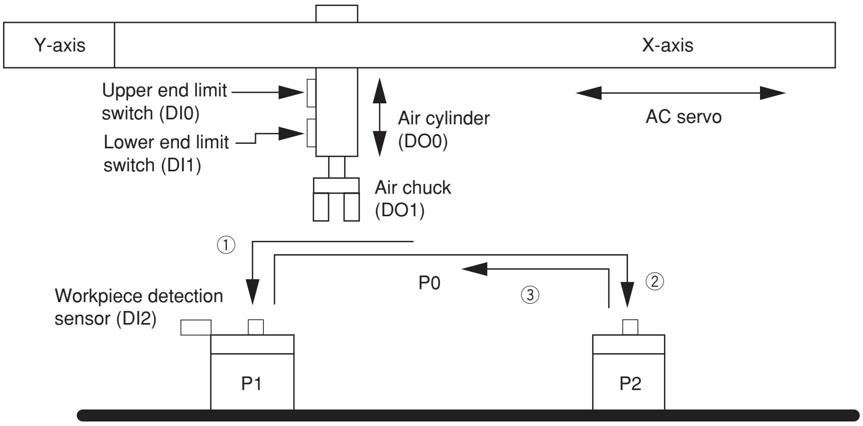
General-purpose output

DO1	Job 1 command	1: Output	0: Canceled
DO2	Job 2 command	1: Output	0: Canceled

Program **Comment**
[NO0]

```
001:DO      1,   0      ; Cancels job 1 command
002:DO      2,   0      ; Cancels job 2 command
003:L       1           ; Label definition
004:MOVA    1,   100    ; Moves to P1
005:DO      1,   1      ; Outputs job 1 command
006:WAIT    1,   1      ; Waits until job 1 is complete
007:DO      1,   0      ; Cancels job 1 command
008:MOVA    2,   100    ; Moves to P2
009:DO      2,   1      ; Outputs job 2 command
010:WAIT    2,   1      ; Waits until job 2 is complete
011:DO      2,   0      ; Cancels job 2 command
012:JMP     1,   0      ; Returns to L1
```

8-5-4 Robot stands by at P0, and moves to P1 and then to P2 to pick and place a workpiece



Operation

① Moves to the workpiece feed position from the standby position, and picks up a workpiece.
② Moves to the workpiece mount position and places the workpiece.
③ Returns to the standby position.

Actuator

Horizontal direction	AC servo motor
Vertical direction	Air cylinder
Hold	Air chuck

General-purpose input

DI0	Upper end limit switch	1: ON	0: OFF
DI1	Lower end limit switch	1: ON	0: OFF
DI2	Workpiece detection sensor	1: Detected	0: No

General-purpose output

DO0	Air cylinder	1: Down	0: Up
DO1	Air chuck	1: Close	0: Open

Point

P0	Robot standby position
P1	Workpiece feed position
P2	Workpiece mount position

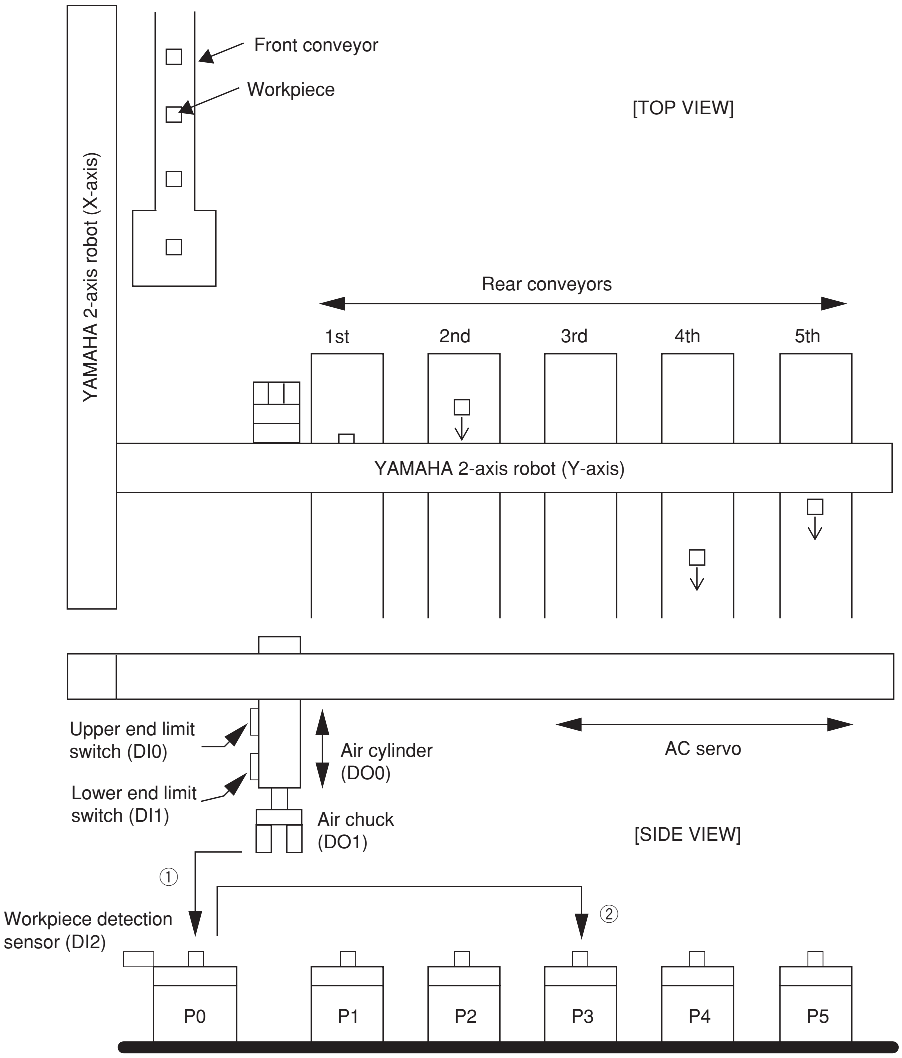
Program **Comment**

[NO1]

```
001:L      1         ; Label definition
002:MOVA   0,   100  ; Moves to the standby position
003:WAIT   2,   1    ; Waits for workpiece feed
004:MOVA   1,   100  ; Moves to the workpiece feed position
005:DO     0,   1    ; Air cylinder moves down                    ┐
006:WAIT   1,   1    ; Waits until the air cylinder moves down    │
007:DO     1,   1    ; Chuck closes                               │ Picks up
008:TIMR   100       ; Delays for one second                      │ workpiece
009:DO     0,   0    ; Air cylinder moves up                      │
010:WAIT   0,   1    ; Waits until the air cylinder moves up      ┘
011:MOVA   2,   100  ; Moves to the workpiece mount position
012:DO     0,   1    ; Air cylinder moves down                    ┐
013:WAIT   1,   1    ; Waits until the air cylinder moves down    │
014:DO     1,   0    ; Chuck opens                                │ Placing
015:TIMR   100       ; Delays for one second                      │ workpiece
016:DO     0,   0    ; Air cylinder moves up                      │
017:WAIT   0,   1    ; Waits until the air cylinder moves up      ┘
018:JMP    1,   1    ; Returns to L1
```

8 ROBOT LANGUAGE

8-5-5 Picking up workpieces flowing on the front conveyor and placing them sequentially on the five rear conveyors



Operation

① Moves to the workpiece feed position and picks up a workpiece.

② Moves to the workpiece mount position and places the workpiece.

Actuator

Horizontal direction	AC servo motor
Vertical direction	Air cylinder
Hold	Air chuck

General-purpose input

DI0	Upper end limit switch	1: ON	0: OFF
DI1	Lower end limit switch	1: ON	0: OFF
DI2	Workpiece A detection sensor	1: Detected	0: No

General-purpose output

DO0	Air cylinder	1: Down	0: Up
DO1	Air chuck	1: Close	0: Open

Point

P0	Workpiece feed position on the front conveyor
P1	Workpiece mount position on the 1st rear conveyor
P2	Workpiece mount position on the 2nd rear conveyor
P3	Workpiece mount position on the 3rd rear conveyor
P4	Workpiece mount position on the 4th rear conveyor
P5	Workpiece mount position on the 5th rear conveyor

Program	Comment
[NO1]	**<<Main routine>>**
`001:L      1`	; Label definition
`002:P      1`	; Initializes the point variable
`003:CALL   2,   5`	; Executes a subroutine five times repeatedly
`004:JMP    1,   1`	; Returns to L1
[NO2]	**<<Picking up and placing a workpiece>>**
`001:WAIT   2,   1`	; Waits for workpiece feed
`002:CALL   3,   1`	; Executes a [PICK] subroutine
`003:CALL   4,   1`	; Executes a [PLACE] subroutine
`004:P+`	; Point variable increment
[NO3]	**<<Picking up a workpiece>>**
`001:MOVA   0,   100`	; Moves to the workpiece feed position
`002:DO     0,   1`	; Air cylinder moves down
`003:WAIT   1,   1`	; Waits until the air cylinder moves down
`004:DO     1,   1`	; Chuck closes
`005:TIMR   100`	; Stays for one second
`006:DO     0,   0`	; Air cylinder moves up
`007:WAIT   0,   1`	; Waits until the air cylinder moves up
[NO4]	**<<Placing a workpiece>>**
`001:MOVA   P,   100`	; Moves to the workpiece mount position
`002:DO     0,   1`	; Air cylinder moves down
`003:WAIT   1,   1`	; Waits until the air cylinder moves down
`004:DO     1,   0`	; Chuck opens
`005:TIMR   100`	; Stays for one second
`006:DO     0,   0`	; Air cylinder moves up
`007:WAIT   0,   1`	; Waits until the air cylinder moves up

8-5-6 Switching the program from I/O

The DRCX series controller does not accept dedicated command inputs for program switching. To switch the program through the I/O, use the program selection signal as a conditional jump input as explained below.
The following method is an example for switching among 16 kinds of programs.

Parameter

Since the number of programs to be selected is 16, set the PRM0 (No. of conditional input points) to 4.

⚠ *CAUTION*
In actual programming, PRM0 must be set to match the number of programs you use. (See the table at the right.)

Number of programs to be selected	Number of DI points	DI numbers to be used
2 or less	1	DI0
4 or less	2	DI0 to DI1
8 or less	3	DI0 to DI2
16 or less	4	DI0 to DI3
32 or less	5	DI0 to DI4
64 or less	6	DI0 to DI5
100 or less	7	DI0 to DI6

General-purpose input

DI0	Program selection 0
DI1	Program selection 1
DI2	Program selection 2
DI3	Program selection 3
DI8	Confirmation of selected program

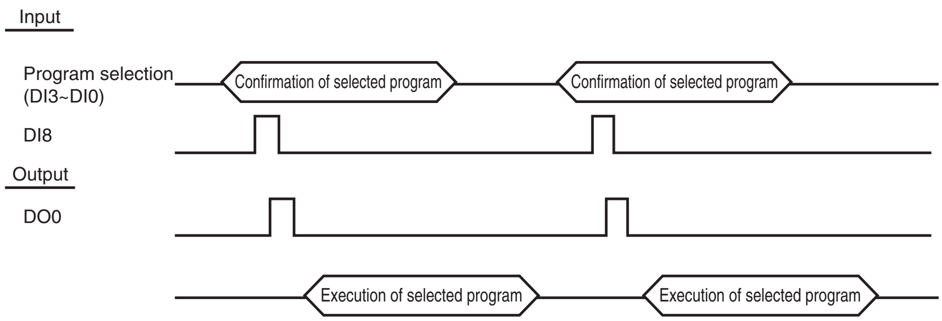
General-purpose output

DO0	Program selection start

<<Program description>>

When using the JMPF statement to select a program, select the general-purpose input/output points (DI8 and DO0 in this case) one at a time and perform the handshake. This is for synchronizing the DRCX controller program with an external device such as a PLC. If this part is omitted, the wrong program might be selected during program selection with the JMPF statement.
In specific operations, an external device should turn on DI8 after confirming DI3 to DI0. The DRCX controller then turns on DO0 just after detecting that DI8 is on, and informs the external device that the program is being selected. When the external device detects that DO0 is on, DI8 should turn off. (DI3 to DI0 should be retained.) Then, when DO0 turns off, this means that the program selection is complete, so it is okay to change DI3 to DI0. The program selection is now complete and actual program operations are executed.
When each selected program has been executed, the operation returns to the top of the program (L0 in the program NO0). The operation returns to the top of the program when there is no program matching the program selection input.

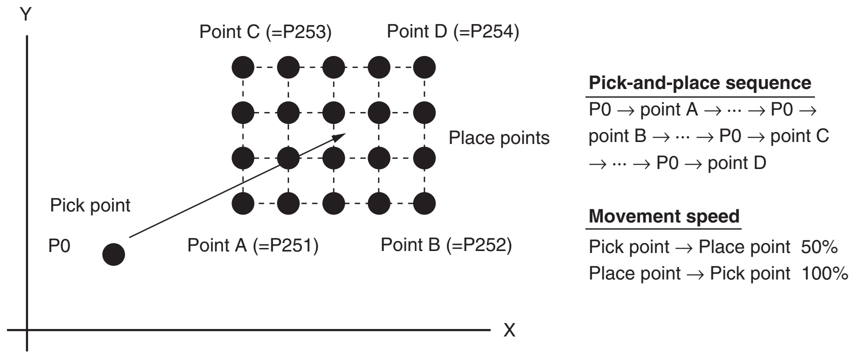
Time chart



```
Program                              Comment
[NO0]
001:L      0                         ; Label definition
002:WAIT   8,   1                    ; Waits for confirmation ON of the selected program  ┐
003:DO     0,   1                    ; Program selection start turns on                   ├Handshaking
004:WAIT   8,   0                    ; Waits for confirmation OFF of the selected program ┘
005:JMPF   1,   1,     0             ; Jumps to L1 of NO1 when input is 0
006:JMPF   1,   2,     1             ; Jumps to L1 of NO2 when input is 1
007:JMPF   1,   3,     2             ; Jumps to L1 of NO3 when input is 2
008:JMPF   1,   4,     3             ; Jumps to L1 of NO4 when input is 3
009:JMPF   1,   5,     4             ; Jumps to L1 of NO5 when input is 4
010:JMPF   1,   6,     5             ; Jumps to L1 of NO6 when input is 5
011:JMPF   1,   7,     6             ; Jumps to L1 of NO7 when input is 6
012:JMPF   1,   8,     7             ; Jumps to L1 of NO8 when input is 7
013:JMPF   1,   9,     8             ; Jumps to L1 of NO9 when input is 8
014:JMPF   1,   10,    9             ; Jumps to L1 of NO10 when input is 9
015:JMPF   1,   11,    10            ; Jumps to L1 of NO11 when input is 10
016:JMPF   1,   12,    11            ; Jumps to L1 of NO12 when input is 11
017:JMPF   1,   13,    12            ; Jumps to L1 of NO13 when input is 12
018:JMPF   1,   14,    13            ; Jumps to L1 of NO14 when input is 13
019:JMPF   1,   15,    14            ; Jumps to L1 of NO15 when input is 14
020:JMPF   1,   16,    15            ; Jumps to L1 of NO16 when input is 15
021:JMP    0,   0                    ; Returns to L0 of program NO0

[NO1]
001:L      1                         ; Label definition
002:DO     0,   0                    ; Program selection is complete (selection start OFF)
    - - - - -                        ; Actual program operation
    JMP    0,   0                    ; Returns to L0 of program NO0

[NO2]
001:L      1                         ; Label definition
002:DO     0,   0                    ; Program selection is complete (selection start OFF)
    - - - - -                        ; Actual program operation
    JMP    0,   0                    ; Returns to L0 of program NO0

   :                                 Programs NO3–NO15 should be created in the same way
   :

[NO16]
001:L      1                         ; Label definition
002:DO     0,   0                    ; Program selection is complete (selection start OFF)
    - - - - -                        ; Actual program operation
    JMP    0,   0                    ; Returns to L0 of program NO0
```

8-5-7 Palletizing for fixed point versus pallet

With this sample program, the robot picks up a workpiece supplied at P0 and places it sequentially on a 4×5 pallet.

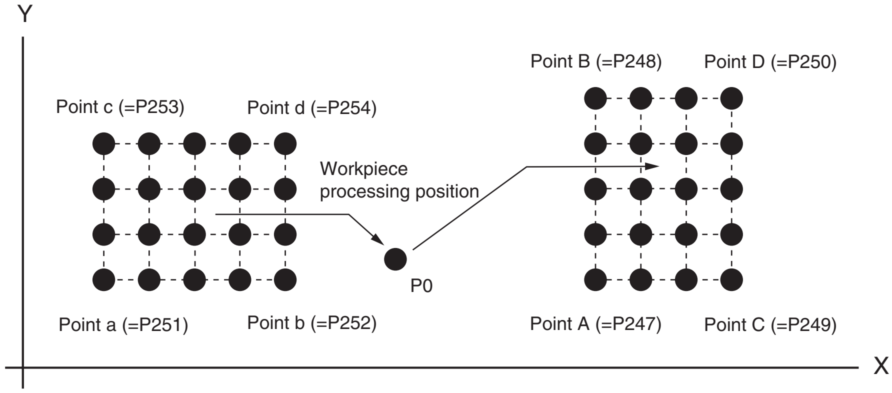


■ Teaching each point of P0 and P251 to P254 should be completed beforehand in PNT (point) mode. (Matrix is defined as pallet number 0 in this example.)

Program				Comment
[NO0]				
001:MAT	4,	5,	0	; Defines a 4×5 matrix as number 0.
002:C	1			; Sets counter variable to 1
003:L	0			; Label definition
004:MOVA	0,	100		; Moves to pick point at speed 100
005:CALL	1,	1		; PICK routine call
006:MSEL	0			; Specifies movement matrix
007:MOVM	C,	50		; Moves to place point (on pallet) at speed 50
008:CALL	2,	1		; PLACE routine call
009:JMPC	1,	20		; Jumps to L1 if counter variable is 20
010:C+				; Counter variable increment
011:JMP	0,	0		; Jumps to L0
012:L	1			; Label definition

8-5-8 Palletizing for pallet versus pallet

With this sample program, the robot picks up a workpiece from a pallet, places it in the processing position P0, and then picks up and places the processed workpiece on a transport pallet.



Pick-and-place sequence
point a → P0 → point A → ··· → point b → P0 → point B → ··· →
point c → P0 → point C → ··· → point d → P0 → point D

■ Teaching each point of P0 and P247 to P254 should be completed beforehand in PNT (point) mode. (Supply pallet is defined as pallet number 0, and transport pallet as pallet number 1 in this example.)

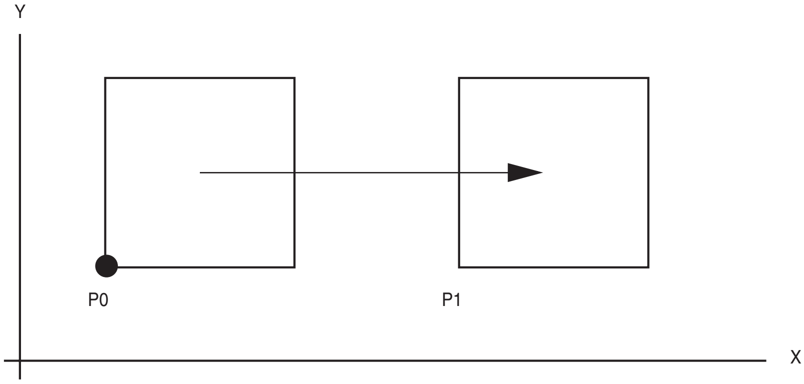
Program **Comment**

[NO0]

```
001:MAT   4,  5,   0     ; 4×5 matrix definition (for supply pallet)
002:MAT   4,  5,   1     ; 4×5 matrix definition (for transfer pallet)
003:C     1              ; Sets counter variable to 1
004:L     0              ; Label definition
005:MSEL  0              ; Matrix selection for feed pallet
006:MOVM  C,  100        ; Moves to supply pallet
007:CALL  1,  1          ; PICK routine call
008:MOVA  0,  100        ; Moves to processing position
009:CALL  2,  1          ; PLACE routine call
010:CALL  3,  1          ; Workpiece processing routine call
011:CALL  1,  1          ; PICK routine call
012:MSEL  1              ; Matrix selection for transfer pallet
013:MOVM  C,  100        ; Moves to transfer pallet
014:CALL  2,  1          ; PLACE routine call
015:JMPC  1,  20         ; Jumps to L1 if counter variable is 20
016:C+                   ; Counter variable increment
017:JMP   0,  0          ; Jumps to L0
018:L     1              ; Label definition
```

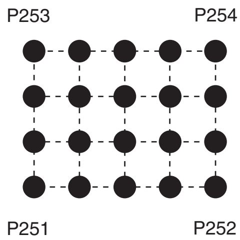
8-5-9 Palletizing for pallet versus pallet, using a SHIFT statement

With this sample program, the robot moves workpieces between two pallets with an equal pitch.



■ Teaching each point of P0, P1 and P251 to P254 should be completed beforehand in PNT (point) mode.
The sample program is written under the following conditions.

- The pallet is defined as pallet number 0 and the matrix is 4×5.
- When performing pallet teaching, data for P252 to P254 is input versus P251 which is viewed as (0.00, 0.00).
- P0 is the amount of shift to the supply pallet, and P1 is the amount of shift to the destination pallet.



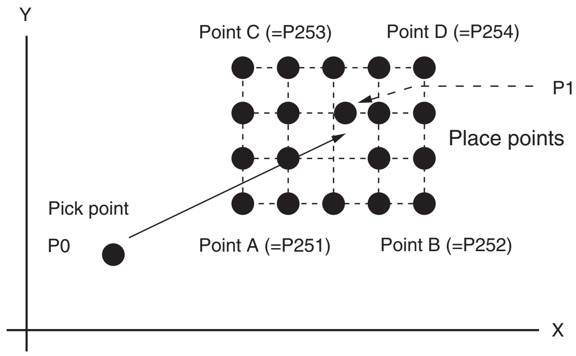
Program **Comment**

[NO0]

```
001:MAT    4,   5,     0     ; Defines 4×5 matrix as pallet number 0
002:MSEL   0                 ; Specifies movement matrix
003:C      1                 ; Sets counter variable to 1
004:L      0                 ; Label definition
005:SHFT   0                 ; Sets pick point shift
006:MOVM   C,   100          ; Moves to pick point (on pallet) at speed 100%
007:CALL   1,   1            ; PICK routine call
008:SHFT   1                 ; Sets place point shift
009:MOVM   C,   100          ; Moves to place point (on pallet) at speed 100%
010:CALL   2,   1            ; PLACE routine call
011:JMPC   1,   20           ; Jumps to L1 if counter variable is 20
012:C+                       ; Counter variable increment
013:JMP    0,   0            ; Jumps to L0
014:L      1                 ; Label definition
```

8-5-10 Palletizing for special pallets

With this sample program, the robot picks up a workpiece supplied at P0 and place it sequentially on a 4×5 pallet. However, the robot does not place a workpiece in the position at row 2 (from bottom), column 3. Moreover, the position at row 3, column 3 is slightly shifted, so it should have data as a different point (P1).



■ Teaching each point of P0, P1 and P251 to P254 should be completed beforehand in PNT (point) mode. (Matrix is defined as pallet number 0 in this example.)

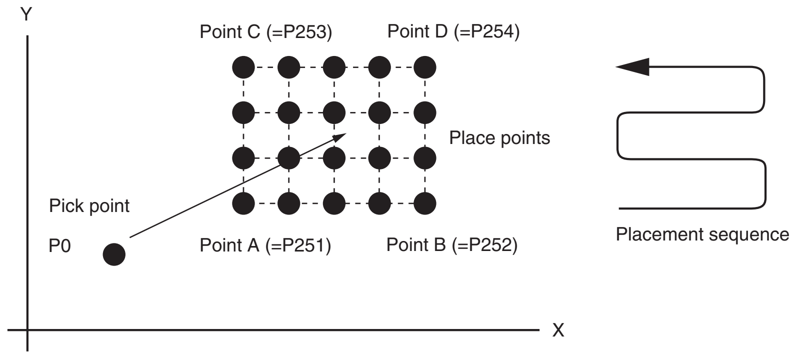
Program **Comment**

[NO0]

```
001:MAT    4,   5,     0    ; Defines 4×5 matrix as pallet number 0
002:C      1                ; Sets counter variable to 1
003:L      0                ; Label definition
004:JMPC   1,   8           ; Jumps to L1 if counter variable is 8
005:MOVA   0,   100         ; Moves to pick point
006:CALL   1,   1           ; PICK routine call
007:JMPC   2,   13          ; Jumps to L2 if counter variable is 13
008:MSEL   0                ; Specifies movement matrix
009:MOVM   C,   100         ; Moves to feed destination point (on pallet)
010:CALL   2,   1           ; PLACE routine call
011:L      1                ; Label definition
012:JMPC   3,   20          ; Jumps to L3 if counter variable is 20
013:C+                      ; Counter variable increment
014:JMP    0,   0           ; Jumps to L0
015:L      2                ; Label definition
016:MOVA   1,   100         ; Moves to position at row 3, column 3
017:CALL   2,   1           ; PLACE routine call
018:JMP    1,   0           ; Jumps to L1
019:L      3                ; Label definition
```

8-5-11 Changing the placement sequence for palletizing

With this sample program, the robot picks up a workpiece supplied at P0 and place it sequentially on a 4×5 pallet with the placement sequence shown below.

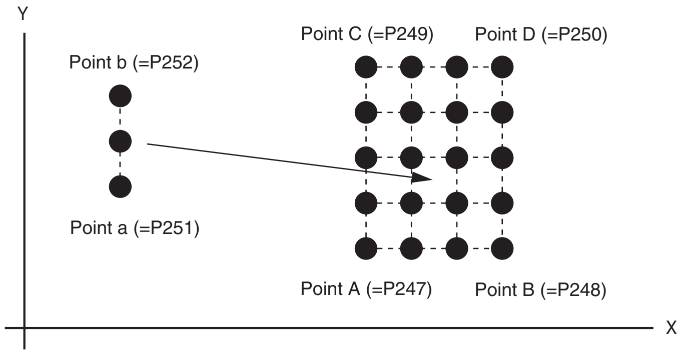


■ Teaching each point of P0 and P251 to P254 should be completed beforehand in PNT (point) mode. (Matrix is defined as pallet number 0 in this example.)

Program	Comment
[NO0]	**<<Main routine>>**
`001:MAT   4,  5,     0`	; Defines 4×5 matrix as pallet number 0
`002:C     1`	; Sets counter variable to 1
`003:CALL  1,  5`	; Calls subroutine NO1 five times
`004:C     10`	; Sets counter variable to 10
`005:CALL  2,  5`	; Calls subroutine NO2 five times
`006:C     11`	; Sets counter variable to 11
`007:CALL  1,  5`	; Calls subroutine NO1 five times
`008:C     20`	; Sets counter variable to 20
`009:CALL  2,  5`	; Calls subroutine NO2 five times
[NO1]	**<<Execution while counting up>>**
`001:MOVA  0,  100`	; Moves to pick point
`002:CALL  3,  1`	; PICK routine call
`003:MSEL  0`	; Specifies movement matrix
`004:MOVM  C,  100`	; Moves to place point (on pallet)
`005:CALL  4,  1`	; PLACE routine call
`006:C+`	; Counter variable increment
[NO2]	**<< Execution while counting down>>**
`001:MOVA  0,  100`	; Moves to feed source point
`002:CALL  3,  1`	; PICK routine call
`003:MSEL  0`	; Specifies movement matrix
`004:MOVM  C,  100`	; Moves to place point (on pallet)
`005:CALL  4,  1`	; PLACE routine call
`006:C-`	; Counter variable decrement

8 ROBOT LANGUAGE

8-5-12 Picking up workpieces from a 1×3 pallet conveyed by the conveyor and placing them on a 5×4 transfer pallet

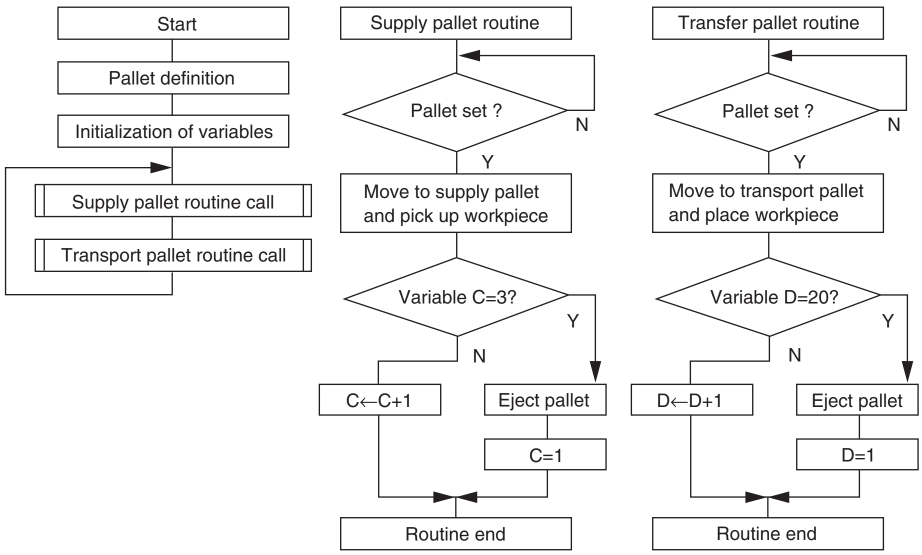


General-purpose input

DI0	Supply pallet	1:Set	0:NO
DI1	Transfer pallet	1:Set	0:NO

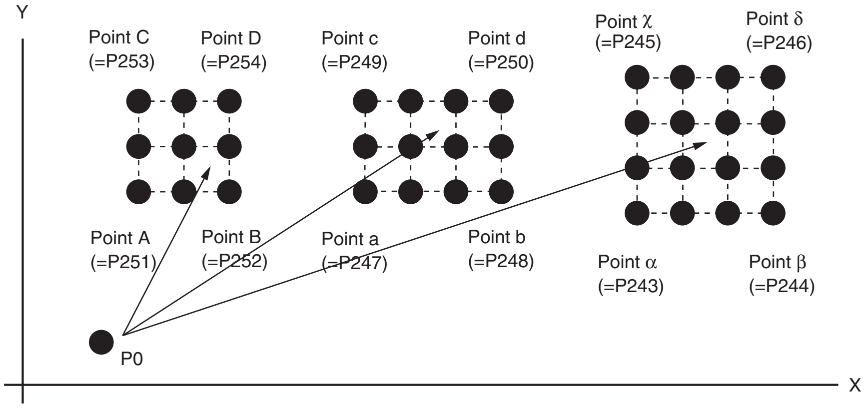
■ Teaching each point of P247 to P252 should be completed beforehand in PNT (point) mode. (Supply pallet is defined as pallet number 0, and transport pallet as pallet number 1 in this example.)

(* Since the supply pallet is a one-dimensional matrix, input of P253 and P254 is not necessary.)



Program				Comment
[NO0]				**<<Main routine>>**
001:MAT	1,	3,	0	; Defines 1×3 matrix (for supply pallet)
002:MAT	5,	4,	1	; Defines 5×4 matrix (for transfer pallet)
003:C	1			; Sets counter variable C to 1
004:D	1			; Sets counter variable D to 1
005:L	0			; Label definition
006:CALL	1,	1		; Calls subroutine NO1
007:CALL	2,	1		; Calls subroutine NO2
008:JMP	0,	0		; Jumps to L0
[NO1]				**<<Supply pallet routine>>**
001:WAIT	0,	1		; Waits until supply pallet is set
002:MSEL	0			; Selects matrix for feed pallet
003:MOVM	C,	100		; Moves to supply pallet
004:CALL	3,	1		; PICK routine call
005:JMPC	1,	3		; Jumps to L1 if counter variable C is 3
006:C+				; Counter variable C increment
007:JMP	2,	1		; Jumps to L2
008:L	1			; Label definition
009:CALL	5,	1		; Supply pallet eject routine call
010:C	1			; Initializes counter variable C to 1
011:L	2			; Label definition
[NO2]				**<<Transfer pallet routine>>**
001:WAIT	1,	1		; Waits until transfer pallet is set
002:MSEL	1			; Selects transport pallet matrix
003:MOVM	D,	100		; Moves to transport pallet
004:CALL	4,	1		; PLACE routine call
005:JMPD	1,	20		; Jumps to L1 if counter variable D is 20
006:D+				; Counter variable D increment
007:JMP	2,	2		; Jumps to L2
008:L	1			; Label definition
009:CALL	6,	1		; Transfer pallet eject routine call
010:D	1			; Initializes counter variable D to 1
011:L	2			; Label definition

8-5-13 Picking up 3 kinds of workpieces conveyed by the conveyor and placing them on the 3×3, 3×4, 4×4 transfer pallets while sorting



General-purpose input

DI1	Workpiece A identification	1: Detected 0: No
DI2	Workpiece B identification	1: Detected 0: No
DI3	Workpiece C identification	1: Detected 0: No

DI4	Workpiece A pallet	1: Set 0: No
DI5	Workpiece B pallet	1: Set 0: No
DI6	Workpiece C pallet	1: Set 0: No

■ Teaching each point of P0 and P243 to P254 should be completed beforehand in PNT (point) mode. (Pallet for workpiece A is defined as pallet 0, pallet for workpiece B as pallet 1, and pallet for workpiece C as pallet 2 in this example.)

Program	**Comment**
[NO0]	**<<Main routine>>**
`001:MAT 3, 3, 0`	; Defines 3×3 matrix (for workpiece A)
`002:MAT 3, 4, 1`	; Defines 3×4 matrix (for workpiece B)
`003:MAT 4, 4, 2`	; Defines 4×4 matrix (for workpiece C)
`004:CSEL 0`	; Sets array element of counter array variable C to 0
`005:C 1`	; Sets counter array variable C[0] to 1
`006:CSEL 1`	; Sets array element of counter array variable C to 1
`007:C 1`	; Sets counter array variable C[1] to 1
`008:CSEL 2`	; Sets array element of counter array variable C to 2
`009:C 1`	; Sets counter array variable C[2] to 1
`010:L 0`	; Label definition
`011:MOVA 0, 100`	; Moves to feed point
`012:CALL 10, 1`	; Waits for workpiece and calls workpiece identification routine
`013:CALL 5, 1`	; PICK routine call
`014:JMPB 1, 1, 1`	; Jumps to L1 when DI1=1 (workpiece A)
`015:JMPB 2, 2, 1`	; Jumps to L2 when DI2=1 (workpiece B)
`016:JMPB 3, 3, 1`	; Jumps to L3 when DI3=1 (workpiece C)
`017:JMP 0, 0`	; Jumps to L0
`018:L 1`	; Label definition
`019:CALL 1, 1`	; Calls subroutine NO1
`020:JMP 0, 0`	; Jumps to L0
`021:L 2`	; Label definition
`022:CALL 2, 1`	; Calls subroutine NO2
`023:JMP 0, 0`	; Jumps to L0
`024:L 3`	; Label definition
`025:CALL 3, 1`	; Calls subroutine NO3
`026:JMP 0, 0`	; Jumps to L0

8

ROBOT LANGUAGE

[NO1] **<<Transfer routine for workpiece A>>**

```
001:WAIT   4,   1          ; Waits until workpiece A pallet is set
002:MSEL   0               ; Selects matrix 0
003:CSEL   0               ; Selects counter array variable C[0]
004:MOVM   C,   100        ; Moves to workpiece A pallet
005:CALL   6,   1          ; PLACE routine call
006:JMPC   1,   9          ; Jumps to L1 if C[0] is 9
007:C+                     ; Counter array variable C[0] increment
008:JMP    2,   1          ; Jumps to L2
009:L      1               ; Label definition
010:CALL   7,   1          ; Eject routine call for workpiece A pallet
011:C      1               ; Initializes counter array variable C[0] to 1
012:L      2               ; Label definition
```

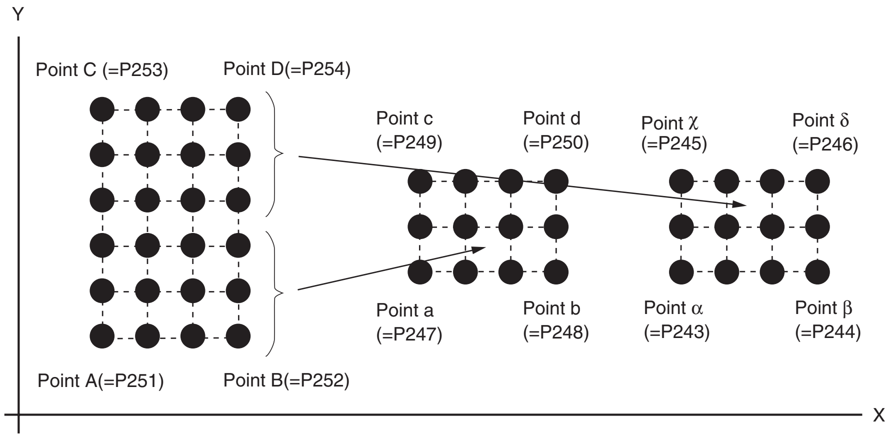
[NO2] **<<Transfer routine for workpiece B>>**

```
001:WAIT   5,   1          ; Waits until workpiece B pallet is set
002:MSEL   1               ; Selects matrix 1
003:CSEL   1               ; Selects counter array variable C[1]
004:MOVM   C,   100        ; Moves to workpiece B pallet
005:CALL   6,   1          ; PLACE routine call
006:JMPC   1,   12         ; Jumps to L1 if C[1] is 12
007:C+                     ; Counter array variable C[1] increment
008:JMP    2,   2          ; Jumps to L2
009:L      1               ; Label definition
010:CALL   8,   1          ; Eject routine call for workpiece B pallet
011:C      1               ; Initializes counter array variable C[1] to 1
012:L      2               ; Label definition
```

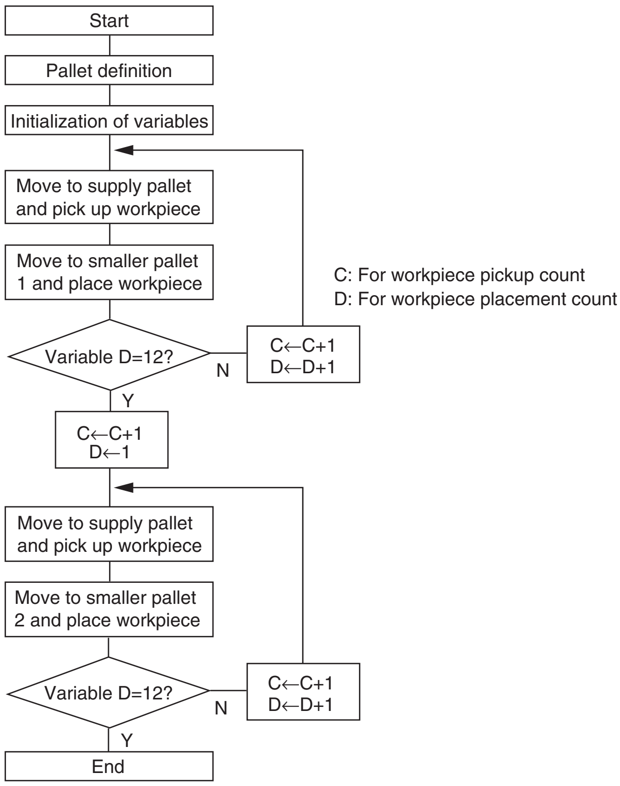
[NO3] **<<Transfer routine for workpiece C>>**

```
001:WAIT   6,   1          ; Waits until workpiece C pallet is set
002:MSEL   2               ; Selects matrix 2
003:CSEL   2               ; Selects counter array variable C[2]
004:MOVM   C,   100        ; Moves to workpiece C pallet
005:CALL   6,   1          ; PLACE routine call
006:JMPC   1,   16         ; Jumps to L1 if C[2] is 16
007:C+                     ; Counter array variable C[2] increment
008:JMP    2,   3          ; Jumps to L2
009:L      1               ; Label definition
010:CALL   9,   1          ; Eject routine call for workpiece C pallet
011:C      1               ; Initializes counter array variable C[2] to 1
012:L      2               ; Label definition
```

8-5-14 Picking up workpieces from a 6×4 pallet and placing them on two smaller 3×4 pallets.



■ Teaching each point of P243 to P254 should be completed beforehand in PNT (point) mode. (Supply pallet is defined as pallet 0, and smaller pallets as pallets 1 and 2 in this example.)



Program **Comment**

[NO0]

```
001:MAT    6,   4,     0     ; Defines 6×4 matrix (for supply pallet)
002:MAT    3,   4,     1     ; Defines 3×4 matrix (for small pallet 1)
003:MAT    3,   4,     2     ; Defines 3×4 matrix (for small pallet 2)
004:C      1                 ; Sets counter variable C to 1
005:D      1                 ; Sets counter variable D to 1
006:L      0                 ; Label definition
007:MSEL   0                 ; Selects matrix for feed pallet
008:MOVM   C,   100          ; Moves to supply pallet
009:CALL   1,   1            ; PICK routine call
010:MSEL   1                 ; Selects matrix for small pallet 1
011:MOVM   D,   100          ; Moves to small pallet 1
012:CALL   2,   1            ; PLACE routine call
013:JMPD   1,   12           ; Jumps to L1 if counter variable D is 12
014:C+                       ; Counter variable C increment
015:D+                       ; Counter variable D increment
016:JMP    0,   0            ; Jumps to L0
017:L      1                 ; Label definition
018:C+                       ; Counter variable C increment
019:D      1                 ; Sets counter variable D to 1
020:L      2                 ; Label definition
021:MSEL   0                 ; Selects matrix for supply pallet
022:MOVM   C,   100          ; Moves to supply pallet
023:CALL   1,   1            ; PICK routine call
024:MSEL   2                 ; Selects matrix for small pallet 2
025:MOVM   D,   100          ; Moves to small pallet 2
026:CALL   2,   1            ; PLACE routine call
027:JMPD   3,   12           ; Jumps to L3 if counter variable D is 12
028:C+                       ; Counter variable C increment
029:D+                       ; Counter variable D increment
030:JMP    2,   0            ; Jumps to L2
031:L      3                 ; Label definition
```

8-5-15 Specifying a position on a pallet from I/O

With this sample program the robot moves to a position on a 1×26 pallet. The position is specified from I/O. (This program can be applied to more than 256 conditional jumps which are not possible with the JMPF statement.)

- Pallet work positions should be specified with general-purpose inputs DI8 to DI0. (in binary digits)

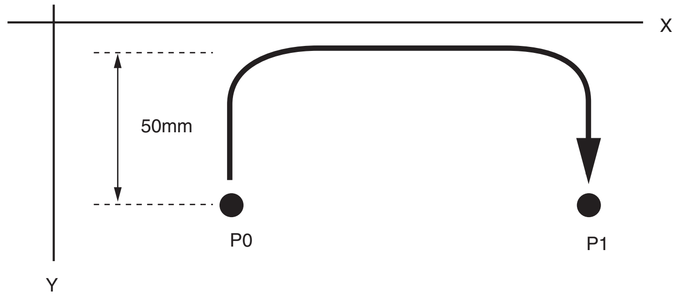
  DI8 to DI0 should be set before executing the program and retained until movement starts.
- Teaching each point of P251 to P254 should be completed beforehand in PNT (point) mode. (Matrix is defined as pallet number 0 in this example.)

Program **Comment**

[NO0]

```
001:MAT    10, 26,   0     ; Defines 10×26 matrix as pallet number 0
002:C      0               ; Sets counter variable to 0
003:JMPB   1,  0,    0     ; Jumps to L1 when DI0=0
004:C+     1               ; Adds 1 to counter variable
005:L      1               ; Label definition
006:JMPB   2,  1,    0     ; Jumps to L2 when DI1=0
007:C+     2               ; Adds 2 to counter variable
008:L      2               ; Label definition
009:JMPB   3,  2,    0     ; Jumps to L3 when DI2=0
010:C+     4               ; Adds 4 to counter variable
011:L      3               ; Label definition
012: JMPB  4,  3,    0     ; Jumps to L4 when DI3=0
013:C+     8               ; Adds 8 to counter variable
014:L      4               ; Label definition
015: JMPB  5,  4,    0     ; Jumps to L5 when DI4=0
016: C+    16              ; Adds 16 to counter variable
017: L     5               ; Label definition
018: JMPB  6,  5,    0     ; Jumps to L6 when DI5=0
019: C+    32              ; Adds 32 to counter variable
020: L     6               ; Label definition
021: JMPB  7,  6,    0     ; Jumps to L7 when DI6=0
022: C+    64              ; Adds 64 to counter variable
023: L     7               ; Label definition
024: JMPB  8,  7,    0     ; Jumps to L8 when DI7=0
025: C+    128             ; Adds 128 to counter variable
026: L     8               ; Label definition
027: JMPB  9,  8,    0     ; Jumps to L9 when DI8=0
028: C+    256             ; Adds 256 to counter variable
029: L     9               ; Label definition
030: MSEL  0               ; Specifies movement matrix
031: MOVM  C,  100         ; Moves to pallet at speed 100
```

8-5-16 Picking up a workpiece at P0 and placing it at P1

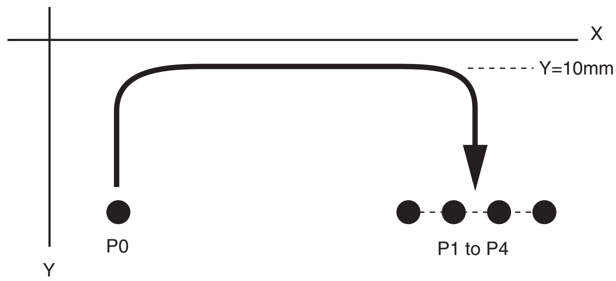


Program **Comment**

[NO0]

```
001:MOVA   0,   100          ; Moves to P0
002:CALL   1,   1            ; PICK routine call
003:ACHI   2,   -50          ; Specifies arch motion to move back Y-axis by -50mm
004:MOVA   1,   100          ; Moves to P1 (arch motion)
005:CALL   2,   1            ; PLACE routine call
```

8-5-17 Picking up workpieces at P0 and placing them sequentially at P1, P2, P3 and P4



Program **Comment**

[NO0]

```
001:P      1                 ; Set point variable to 1
002:CALL   1,   4            ; Calls routine NO1 four times
```

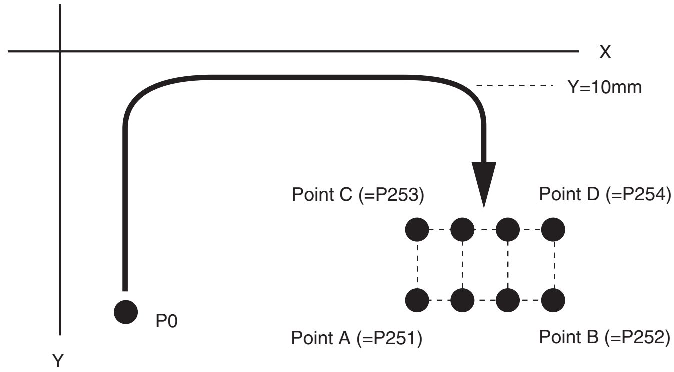
[NO1]

```
001:ACHA   2,   10           ; Specifies arch motion to move back Y-axis by Y=10mm
002:MOVA   0,   100          ; Moves to P0 (arch motion)
003:CALL   2,   1            ; PICK routine call
004:ACHA   2,   10           ; Specifies arch motion to move back Y-axis by Y=10mm
005:MOVA   P,   100          ; Moves to P1-P4 (arch motion)
006:CALL   3,   1            ; PLACE routine call
007:P+                       ; Point variable increment
```

8

ROBOT LANGUAGE

8-5-18 Picking up workpieces at P0 and placing them sequentially on a 2×4 pallet

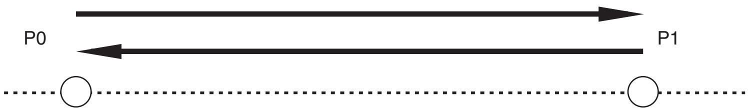


■ Teaching each point of P0 and P251 to P254 should be completed beforehand in PNT (point) mode. (Matrix is defined as pallet number 0 in this example.)

Program				**Comment**
[NO0]				
001:MAT	2,	4,	0	; Defines 2×4 matrix as pallet number 0
002:C	1			; Sets counter variable to 1
003:L	0			; Label definition
004:ACHA	2,	10		; Specifies arch motion to move back Y-axis by Y=10mm
005:MOVA	0,	100		; Moves to pick point
006:CALL	1,	1		; PICK routine call
007:MSEL	0			; Specifies movement matrix
008:ACHA	2,	10		; Specifies arch motion to move back Y-axis by Y=10mm
009:MOVM	C,	100		; Moves to place point (on pallet)
010:CALL	2,	1		; PLACE routine call
011:JMPC	1,	8		; Jumps to L1 if counter variable is 8
012:C+				; Counter variable increment
013:JMP	0,	0		; Jumps to L0
014:L	1			; Label definition

8-5-19 Axis movement and I/O multi-task

The robot moves between two points and performs multi-task I/O operation in asynchronous mode.



General-purpose input/output

DI0	Job status detection
DO0	Job instruction output

Program **Comment**
[NO0]

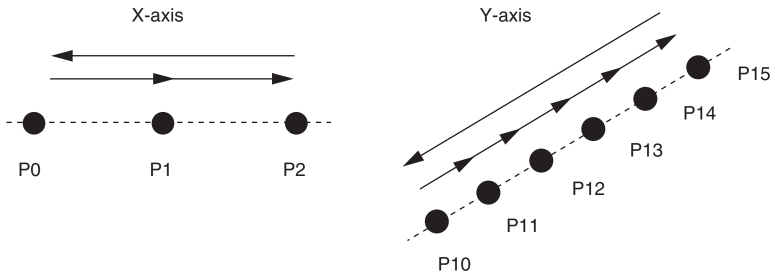
```
001:TON    1,  1,     0 ; Starts program NO1 as task 1
002:L      0            ; Label definition
003:MOVA   0,  100      ; Moves to P0 at speed 100
004:TIMR   100          ; Delays for one second
005:MOVA   1,  100      ; Moves to P1 at speed 100
006:TIMR   100          ; Delays for one second
007:JMP    0,  0        ; Returns to L0
```

Program **Comment**
[NO1]

```
001:L      0            ; Label definition
002:WAIT   0,  0        ; Waits until the job is finished
003:DO     0,  1        ; Issues the job start instruction
004:WAIT   0,  1        ; Confirms that the job has started
005:DO     0,  0        ; Turns off the job start signal
006:JMP    0,  1        ; Returns to L0
```

8-5-20 Multi-robot operation

Two single-axis robots are used to perform multi-task operation in asynchronous mode.



■ Teaching each point of P0 to P2 and P10 to P15 should be completed beforehand in PNT (point) mode. (Y-axis data at P0 to P2 and X-axis data at P10 to P15 can be any value since they are not used.)

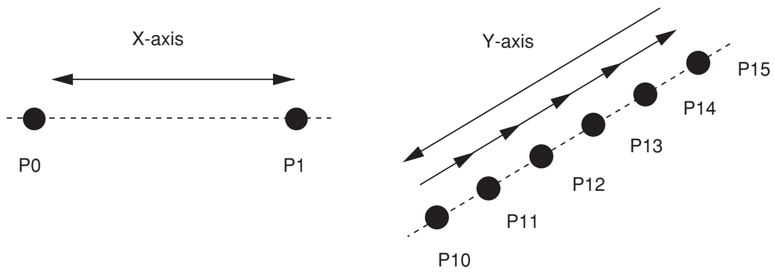
Program				Comment
[NO0]				**<<Task 0>>**
001:TON	1,	2,	0	; Starts program NO2 as task 1
002:L	0			; Label definition
003:P	0			; Sets point variable to 0
004:CALL	1,	3		; Calls subroutine NO1 three times
005:JMP	0,	0		; Returns to L0
[NO1]				
001:DRVA	1,	P,	100	; X-axis robot moves to P0-P2
002:P+				; Point variable increment
[NO2]				**<<Task 1>>**
001:L	0			; Label definition
002:P	10			; Sets point variable to 10
003:CALL	3,	6		; Calls subroutine NO3 six times
004:JMP	0,	2		; Returns to L0
[NO3]				
001:DRVA	2,	P,	100	; Y-axis robot moves to P10-P15
002:P+				; Point variable increment

8

ROBOT LANGUAGE

8-5-21 Synchronization in multi-robot operation

This sample program uses two single-axis robots for multi-task operation. One robot repeatedly moves between two points while the other robot moves at a certain pitch. In this case, each robot simultaneously starts to move from the start positions P0 and P10.



Memory input/output

DO100	X-axis	1: Standby	0: Moving
DO101	Y-axis	1: Standby	0: Moving

■ Teaching each point of P0, P1 and P10 to P15 should be completed beforehand in PNT (point) mode. (Y-axis data at P0 and P1 and X-axis data at P10 to P15 can be any value since they are not used.)

Program **Comment**

[NO0]

```
001:TON     1,   1,      0      ; Starts program NO1 as task 1
002:L       0                   ; Label definition
003:DRVA    1,   0,      100    ; X-axis robot moves to P0
004:DO      100,1               ; X-axis robot reaches start point start position
005:WAIT    101,1               ; Waits until Y-axis robot moves to start position
006:DO      100,0               ; X-axis robot is moving
007:DRVA    1,   1,      100    ; X-axis robot moves to P1
008:JMP     0,   0              ; Returns to L0
```

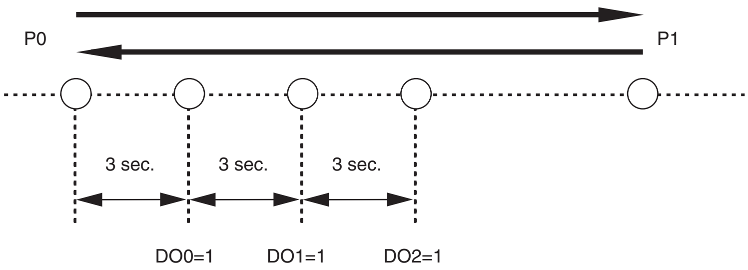
[NO1]

```
001:L       0                   ; Label definition
002:DRVA    2,   10,     100    ; Y-axis robot moves to P10
003:DO      101,1               ; Y-axis robot reaches start position
004:WAIT    100,1               ; Waits until X-axis moves to start position
005:DO      101,0               ; Y-axis robot is moving
006:P       11                  ; Sets point variable to 11
007:CALL    2,   5              ; Calls subroutine NO2 five times
008:JMP     0,   1              ; Returns to L0
```

[NO2]

```
001:DRVA    2,   P,      100    ; Y-axis robot moves to P11 to P15
002:P+                          ; Point variable increment
```

8-5-22 Turning ON general-purpose outputs during robot movement after a certain time has elapsed



Point

P0	Start position
P1	Target position

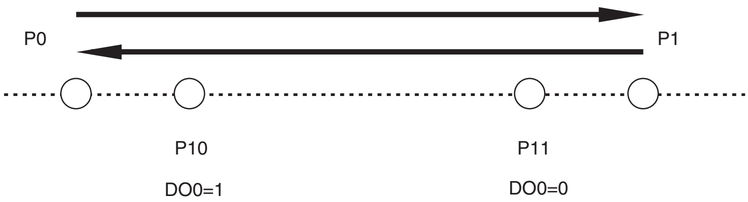
Program **Comment**
[NO0]

```
001:L     0              ; Label definition
002:MOVA  0,  100        ; Moves to P0 at speed 100
003:DO    0,  0          ; Turns DO0 off
004:DO    1,  0          ; Turns DO1 off
005:DO    2,  0          ; Turns DO2 off
006:TON   1,  1,      0  ; Starts program NO1 as task 1
007:MOVA  1,  10         ; Moves to P1 at speed 10
008:JMP   0,  0          ; Returns to L0
```

Program **Comment**
[NO1]

```
001:TIMR  300            ; Delays for 3 seconds
002:DO    0,  1          ; Turns DO0 on
003:TIMR  300            ; Delays for 3 seconds
004:DO    1,  1          ; Turns DO1 on
005:TIMR  300            ; Delays for 3 seconds
006:DO    2,  1          ; Turns DO2 on
```

8-5-23 Turning ON a general-purpose output during robot movement when it has passed a specified position



Point

P0	Start position
P1	Target position
P10	Position at DO0=1
P11	Position at DO0=0

■ When P1 is nearer to the plus side than P0 on both axes:

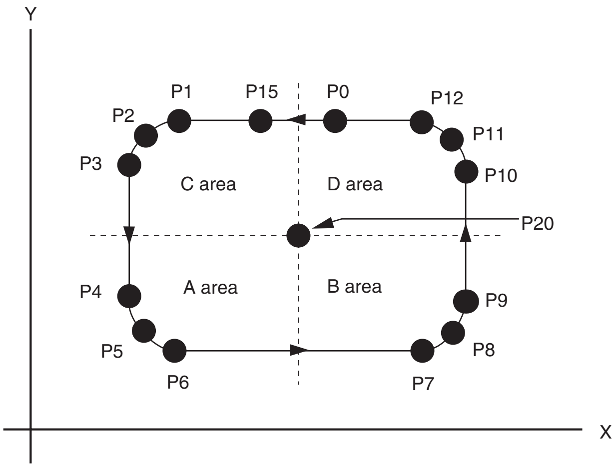
Program **Comment**
[NO0]

```
001:L      0                 ; Label definition
002:MOVA   0,   100          ; Moves to P0 at speed 100
003:TON    1,   1,        0  ; Starts program NO1 as task 1
004:MOVA   1,   10           ; Moves to P1 at speed 10
005:JMP    0,   0            ; Returns to L0
```

Program **Comment**
[NO1]

```
001:DO     0,   0            ; Turns DO0 off
002:P      10                ; Sets the point variable to 10
003:L      0                 ; Label definition
004:JMPP   0,   11           ; Jumps to L0 when the robot does not reach P10
005:DO     0,   1            ; Turns DO0 on
006:P      11                ; Sets the point variable to 11
007:L      1                 ; Label definition
008:JMPP   1,   11           ; Jumps to L1 when the robot does not reach P11
009:DO     0,   0            ; Turns DO0 off
```

8-5-24 Sealing



General-purpose output

DO0	Application 1: Start 0: Stop

Program **Comment**

[NO0]

```
001:MOVA   0,   100          ; Moves to start position
002:TON    1,   1,     0     ; Starts program NO1 as task 1
003:MOVL   1,   10           ; Moves to P1
004:MOVC   2,   10,    0     ; Moves along circular segment locus passing through
                               P1, P2 and P3
005:MOVL   4,   10           ; Moves to P4
006:MOVC   5,   10,    0     ; Moves along circular segment locus passing through
                               P4, P5 and P6
007:MOVL   7,   10           ; Moves to P7
008:MOVC   8,   10,    0     ; Moves along circular segment locus passing through
                               P7, P8 and P9
009:MOVL   10,  10           ; Moves to P10
010:MOVC   11,  10,    0     ; Moves along circular segment locus passing through
                               P10, P11 and P12
011:MOVL   15,  10           ; Moves to P15
012:MOVA   15,  10           ; Waits until movement is complete
```

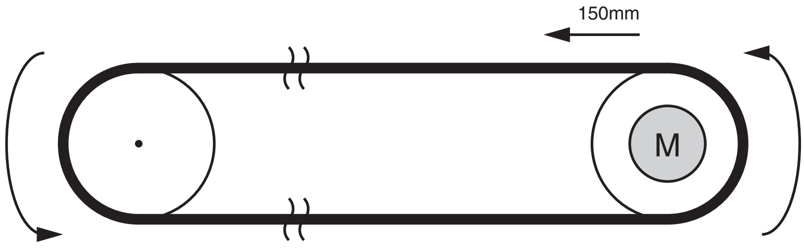
[NO1]

```
001:P      20                ; Sets point variable to 20
002:L      0                 ; Label definition
003:JMPP   1,   21           ; Jumps L1 when area C is entered
004:JMP    0,   1            ; Returns to L0
005:L      1                 ; Label definition
006:DO     0,   1            ; Starts application
007:L      2                 ; Label definition
008:JMPP   3,   12           ; Jumps L3 when area B is entered
009:JMP    2,   1            ; Returns to L2
010:L      3                 ; Label definition
011:JMPP   4,   21           ; Jumps L4 when area C is entered
012:JMP    3,   1            ; Returns to L3
013:L      4                 ; Label definition
014:DO     0,   0            ; End of application
```

8 ROBOT LANGUAGE

8-5-25 Limitless movement at same pitch

The X-axis can be moved continuously in the same direction at the same pitch (e.g. 150mm) for cycle conveyor applications.



■ Make the following settings in advance to enable the limitless movement function.
- Set the X-axis position data unit parameter to 2.
- Set the X-axis plus soft limit to 200, which is a multiple of the lead equivalent value. (The lead equivalent value is assumed to be 20mm.)

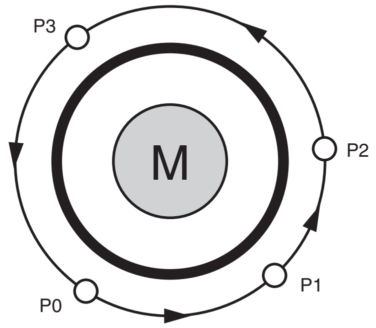
■ Set the X-axis coordinate data to P0=0, P1=150 in advance in PNT (point) mode.
(The Y-axis data can be any value since it is not used.)

Program **Comment**
[NO0]

```
001:DRVA   1,   0,     100   ; X-axis moves to P0 at speed 100
002:L      0                 ; Label definition
003:DRVI   1,   1,     100   ; X-axis moves 150mm
004:JMP    0,   0            ; Jumps to L0
```

8-5-26 Limitless rotation

The Y-axis can be moved continuously in the same direction for index table applications.



■ Make the following setting in advance to enable the limitless movement function.
  • Set the Y-axis position data unit parameter to 3.

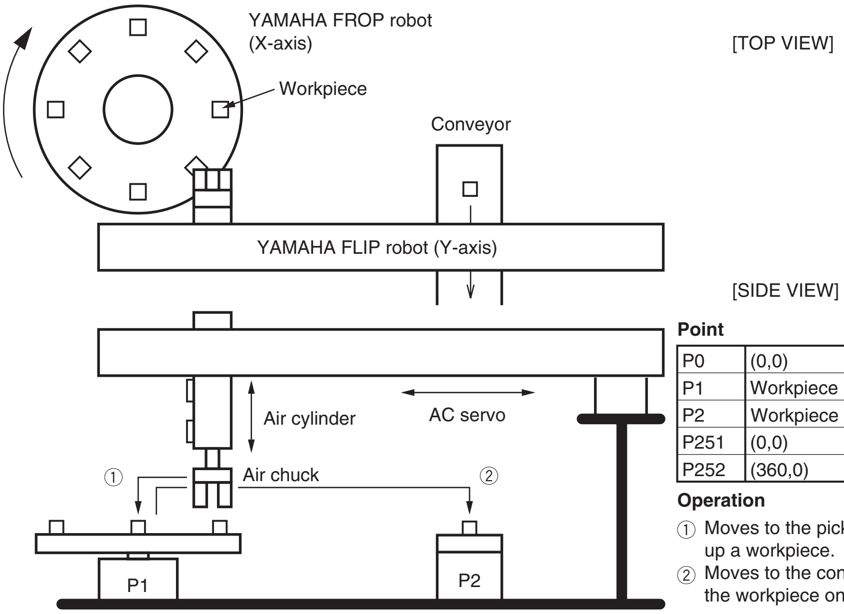
■ Teaching each of P0 to P3 should be completed in advance in PNT (point) mode. (The X-axis data can be any value since it is not used.)

Program **Comment**

[NO0]

```
001:L      1,                   ; Label definition
002:DRVA   2,   0,     100      ; Y-axis moves to P0 at speed 100
003:DRVA   2,   1,     100      ; Y-axis moves to P1 at speed 100
004:DRVA   2,   2,     100      ; Y-axis moves to P2 at speed 100
005:DRVA   2,   3,     100      ; Y-axis moves to P3 at speed 100
006:JMP    0,   0               ; Jumps to L0
```

8-5-27 Picking up workpieces supplied from an index table and placing them on a conveyor



Point

P0	(0,0)
P1	Workpiece pick point on table
P2	Workpiece place point on conveyor
P251	(0,0)
P252	(360,0)

Operation

① Moves to the pick point and picks up a workpiece.
② Moves to the conveyor and places the workpiece on the conveyor.

■ Make the following setting in advance to enable the limitless movement function.

- Set the X-axis position data unit parameter to 3.

■ Data settings for P0, P251 and P252 and also teaching for each point of P1 and P2 should be completed in advance in PNT (point) mode.

(The X-axis data at P2 can be any value since it is not used.)

■ In the program sample below, the pallet number is defined as 0 and the matrix is 1×9. (Workpieces are set at 8 points on one table, but the matrix is viewed as 1×9 assuming that one turn (0 to 360°) is divided.

Program **Comment**

[NO0]

```
001:MAT    9,  0            ; Defines 1×9 matrix as pallet number 0
002:L      0                ; Label definition
003:C      1                ; Sets counter variable to 1
004:L      1                ; Label definition
005:SHFT   1                ; Sets pick point shift
006:MSEL   0                ; Specifies movement matrix
007:MOVM   C,  100          ; Moves to supply point (on table) at speed 100
008:CALL   1,  1            ; PICK routine call
009:SHFT   0                ; Clears shift setting
010:DRVA   2,  2,    100    ; Y-axis moves to place point (on conveyor) at speed 100
011:CALL   2,  1            ; PLACE routine call
012:JMPC   0,  8            ; Jumps to L0 if counter variable is 8
013:C+                      ; Counter variable increment
014:JMP    1,  0            ; Jumps to L1
```

Chapter 9 OPERATING THE ROBOT

This chapter describes how to actually operate the robot. If the program has already been completed, you will be able to operate the robot by the time you finish reading this chapter.
There are two types of robot operation: step and automatic. In step operation, the program is executed one step at a time, with a step being carried out each time the RUN key on the TPB is pressed. This is used when you want to check the program as it is being carried out. In automatic operation, the entire program is executed without stopping, from beginning to end.
This chapter also covers how to initiate and recover from an emergency stop.

9-1 Performing Return-to-Origin

There are two methods for detecting the origin position (reference point): search method and mark method. The search method is further divided into the origin sensor method and stroke-end detection method. In the mark method, you can move the robot to a desired position (mark position) and set it as the particular coordinate position to determine a reference point.
The following sections explain how to perform return-to-origin by using the search method and mark method.

Once return-to-origin is performed after the robot cable and absolute battery are connected, there is no need to repeat it even when the controller is turned off. (As an exception, return-to-origin becomes incomplete if the absolute backup function is disabled or a parameter relating to the origin is changed. Return-to-origin must be reperformed in that case.)

9-1-1 Return-to-origin by the search method

When the search method is selected as the origin detection method (PRM55 and PRM95 are set to 0 or 1), perform return-to-origin with the procedure below.

1) On the initial screen, press F2 (OPRT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (ORG).

```
[OPRT]
select menu

1ORG 2STEP3AUTO
```

3) Select the axis to perform return-to-origin.
   To perform return-to-origin on all axes, press F1 (ALL). To perform return-to-origin on the X-axis only, press F2 (X). To perform return-to-origin on the Y-axis only, press F3 (Y).

```
[OPRT-ORG]
select axis for
origin control
1ALL 2X   3Y   4next
```

4) To perform return-to-origin, press F1 (yes).
   To cancel the operation, press F2 (no).

```
[OPRT-ORG-SEARCH]
ORG search OK ?

1yes 2no
```

5) This screen is displayed during return-to-origin. Pressing STOP during the operation brings the robot to a halt and displays a message. Then, pressing the ESC key returns to the screen of step 2.

```
[OPRT-ORG-SEARCH]
searching ···
```

6) When return-to-origin is completed normally, the machine reference appears on the lower right of the screen. Pressing the ESC key returns to the screen of step 2.

```
[OPRT-ORG-SEARCH]
origin complete
machine ref. X=50%
             Y=50%
```

NOTE

When the mark method is used for one axis and the search method for the other axis, return-to-origin on the axis using the mark method must first be completed before performing return-to-origin by selecting F1 (ALL) in step 3.
(When F1 (ALL) was selected, the controller checks whether return-to-origin has been performed on the axis using the mark method. If return-to-origin is incomplete on that axis, an error will be issued and return-to-origin operation will not be completed.)

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)
- ***Return-to-origin movement speed is limited to 10mm/s or less (10deg/s for rotary robots) in "SERVICE mode state" when the robot movement speed limit is enabled.***
- ***If the hold-to-run function is enabled, robot movement stops upon releasing F1 (yes) in step 4 in "SERVICE mode state". (You must hold down F1 (yes) until return-to-origin is complete.)***

CAUTION

When performing return-to-origin by the stroke-end detection method, do not interrupt the return-to-origin operation while detecting the origin (while contacting the mechanical limit). Otherwise, the operation will stop due to a controller overload alarm and the power will have to be turned on again.

CAUTION

If return-to-origin operation using the stroke-end detection method must be repeated, wait at least 5 seconds before repeating it.

9-1-2 Return-to-origin by the mark method

When the mark method is selected as the origin detection method (PRM55 and PRM95 are set to 2), perform return-to-origin with the procedure below.

1) Press F2 (OPRT) on the initial screen.

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F1 (ORG).

```
[OPRT]
select menu

1ORG 2STEP3AUTO
```

3) Select the axis to perform return-to-origin.

   To perform return-to-origin on the X-axis only, press F2 (X). To perform return-to-origin on the Y-axis only, press F3 (Y).

   (Return-to-origin on the axes using the mark method cannot be performed with F1 (ALL). When F1 (ALL) is selected, the controller checks only whether return-to-origin has been performed.)

```
[OPRT-ORG]
select axis for
origin control
1ALL 2X   3Y   4next
```

4) To use the teaching playback method for setting a mark position, press F1 (TCH). To use the direct teaching method to set a mark position, press F2 (DTCH).

```
[OPRT-ORG-MARK-X]
select menu

1TCH 2DTCH
```

5) When the teaching playback method was selected, use the X- and X+ keys to move the robot along the X-axis to the mark position, and use the Y- and Y+ keys to move along the Y-axis.

   The movement method is just the same as the teaching playback method for point data. (See "7-2 Teaching Playback".)

   When the robot reaches the mark position, press ➡. (At this point, check that the machine reference is in a range from 25 to 75%. Otherwise, the origin point cannot be set correctly.)

```
[ORG-MARK-TCH](1) 50
move at mark point
ref. 50%   [    0.00]
1DO  2SPD 3S_SET
```

6) When the direct teaching method is selected, a message appears asking you to press the emergency stop button. Press the emergency button on the TPB.

```
[ORG-MARK-DTCH]

press EMG. button
```

7) Move the robot by hand to the mark position, and then press ⇨.

(At this point, check that the machine reference is in a range from 25 to 75%. Otherwise, the origin point cannot be set correctly.)

```
[ORG-MARK-DTCH]
move at mark point
ref. 50%   [    0.00]
1DO  2BRK
```

8) When you press ⇨ after moving the robot to the mark position by the teaching playback method or direct teaching method, the screen changes to allow entering the coordinate values for the mark position. (The display at the right shows an example when the X-axis was selected in step 3.)

While checking the robot stays at the mark position, use the number keys to enter the coordinate values that you want the controller to recognize as the current position, and then press ⇨.

```
[ORG-MARK-TCH]
X axis
input mark position
        [_        ] [mm]
```

9) Press F1 (yes) to set the origin. To cancel, press F2 (no).

```
[ORG-MARK-TCH]
ORG mark OK ?

1yes 2no
```

10) If the machine reference is not in a range from 25 to 75%, the message at the right appears informing you that the origin cannot be set. In this case, press ESC and retry the above procedure.

```
[ORG-MARK-TCH]
cannot set origin
as bad machine ref.
```

11) When the origin has been set correctly, the message at the right appears. Pressing the ESC key returns to step 5 when the teaching playback method was used or to step 7 when the direct teaching method was used.

```
[ORG-MARK-TCH]
origin complete
```

NOTE

When you check the robot position after setting the mark position coordinates, the robot position is not always at the coordinates specified as the mark position. This is because the mark position is synchronized to prevent positional shift and make an exact match when the motor's electrical angle is "0".
When the motor's electrical angle is "0", the machine reference is just 50%. This means that as the machine reference deviates from 50%, the robot position moves away from the coordinates specified as the mark position.

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***Robot movement speed is limited to 10mm/s or less (10deg/s for rotary robots) in "SERVICE mode state" when the robot movement speed limit is enabled.***

9-2 Using Step Operation

The following procedure explains how to perform step operation. In the case of a multi-task program, only the task currently selected is executed in step operation.

1) On the initial screen, press **F2** (OPRT).

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

2) Next, press **F2** (STEP).

```
[OPRT]
select menu


1ORG 2STEP3AUTO
```

3) If the program number displayed on the screen is not the one to be run, press **F3** (CHG).

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

4) Using the number keys, enter the number of the program to be executed, and then press ➡.

```
[OPRT-STEP] 100 0: 0
PGM  No = _
(program No)  0→99
```

5) The first step of the selected program is displayed on the screen. To change the execution speed, press **F1** (SPD).

```
[OPRT-STEP] 100 0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

6) Enter the execution speed using the number keys, and press ➡.

```
[OPRT-STEP] 100 0:10
SPEED = _
(speed) 1→100
```

7) The screen returns to step 5. Pressing **RUN** at this point executes the first step.

```
[OPRT-STEP] 50  0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

8) This screen is displayed while the program is being executed.

```
[OPRT-STEP]
running ···
```

9) Pressing STOP during execution brings the robot to a halt and displays a message on the screen. To return to step 7, press the ESC key. Pressing RUN again executes the interrupted step.

```
[OPRT-STEP] 50  0:10
001:MOVA 999,50
 [  201.11,  164.89]
1SPD 2RSET3CHG 4next
```

10) When execution is finished, the second step is displayed. Each time RUN is pressed from this point on, the currently displayed step is executed. When the last step has been executed, the message "program end" is displayed. To return to the first step from the program end, press the ESC key.

```
[OPRT-STEP]

60:Program end
```

11) To switch the execution task in a multi-task program, press F4 (next) to change the function menu display and then press F3 (CHGT).

```
[OPRT-STEP] 50  1:11
001:WAIT 0  ,1
 [  250.00,  200.00]
1VAL 2S_ON3CHGT4next
```

12) Each time you press F3 (CHGT), the task currently in progress is switched. When the task you want to execute is selected, press RUN to execute the step displayed for the selected task.

```
[OPRT-STEP] 50  2:12
001:DO   1  ,1
 [  250.00,  200.00]
1VAL 2S_ON3CHGT4next
```

13) To return to the first step of the program from any other step and initiate execution again, press F2 (RSET).

```
[OPRT-STEP] 50  0:10
035:TIMR 100
 [  250.00,  200.00]
1SPD 2RSET3CHG 4next
```

14) The screen returns to step 5, and the process is repeated from that point.

```
[OPRT-STEP] 50  0:10
001:MOVA 999,50
 [  250.00,  200.00]
1SPD 2RSET3CHG 4next
```

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***Step operation cannot be performed in "SERVICE mode state" when automatic operation and step operation are prohibited.***
- ***Robot movement speed is limited to 3% or less of maximum speed in "SERVICE mode state" when the robot movement speed limit is enabled.***
- ***If the hold-to-run function is enabled, step operation stops upon releasing RUN in "SERVICE mode state". When one step has been executed, you must release RUN and then press RUN again to execute the next step.***

9-3 Using Automatic Operation

The following procedure explains how to perform automatic operation. All the tasks started in a multi-task program are executed by automatic operation.

1) On the initial screen, press F2 (OPRT).

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

2) Next, press F3 (AUTO).

```
[OPRT]
select menu


1ORG 2STEP3AUTO
```

3) If the program number displayed on the screen is not the one to be run, press F3 (CHG).

```
[OPRT-AUTO] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

4) Using the number keys, enter the number of the program to be executed and then press ➡.

```
[OPRT-AUTO] 100 0: 0
PGM  No = _
(program No) 0→99
```

5) The first step of the selected program is displayed on the screen. To change the execution speed, press F1 (SPD).

```
[OPRT-AUTO] 100 0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

6) Enter the execution speed with the number keys and press ➡.

```
[OPRT-AUTO] 100 0:10
SPEED = _
(speed) 1→100
```

7) The screen returns to step 5. Pressing RUN at this point executes the program all the way to the last step.

```
[OPRT-AUTO] 50  0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

8) This is the screen displayed while the program is being executed.

```
[OPRT-AUTO]
running ···
```

9) Pressing STOP during execution brings the robot to a halt and displays the message "stop key". Press the ESC key to display the step where execution was interrupted.

Pressing RUN will cause execution to resume from the step where it was interrupted. When the last step has been executed, the message "program end" is displayed. Pressing the ESC key returns the screen to that shown in Step 7.

```
[OPRT-AUTO]

60:program end
```

10) To switch to the display of another task in a multi-task program, press F4 (next) to change the function menu display and then press F3 (CHGT).

```
[OPRT-AUTO] 50  1:11
010:WAIT 0  ,1
 [  250.00,  200.00]
1VAL 2S_ON3CHGT4next
```

11) Each time you press F3 (CHGT), the task display is switched.

```
[OPRT-AUTO] 50  2:12
015:DO   1  ,1
 [  250.00,  200.00]
1VAL 2S_ON3CHGT4next
```

12) To return to the first step of the program from any other step and initiate execution again, press F2 (RSET).

```
[OPRT-AUTO] 50  0:10
035:TIMR 100
 [  250.00,  200.00]
1SPD 2RSET3CHG 4next
```

13) The screen returns to step 5, and the process is repeated from that point.

```
[OPRT-AUTO] 50  0:10
001:MOVA 999,50
 [  250.00,  200.00]
1SPD 2RSET3CHG 4next
```

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***Automatic operation cannot be performed in "SERVICE mode state" when automatic operation and step operation are prohibited.***
- ***Robot movement speed is limited to 3% or less of maximum speed in "SERVICE mode state" when the robot movement speed limit is enabled.***
- ***If the hold-to-run function is enabled, automatic operation stops upon releasing RUN in "SERVICE mode state".***

9-4 Switching the Execution Program

The following procedure explains how to switch the program in automatic operation. Use the same procedure in step operation.
The program selected by this procedure will be the lead program to which the execution sequence always returns after program reset.
When the program is switched, reset is automatically performed, so all general-purpose outputs are turned off.

* As exceptions, DO5 does not turn OFF when PRM2 (operation at return-to-origin complete parameter) is set to 1 or 3, and DO7 does not turn OFF when PRM21 (servo status output parameter) is set to 1.

1) If the program number displayed on the screen is not the one to be run, press F3 (CHG).

```
[OPRT-AUTO] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

2) Using the number keys, enter the number of the program to be executed and then press ➡.

```
[OPRT-AUTO] 100 0: 0
PGM  No = _
(program No) 0→99
```

3) The first step of the selected program is displayed on the screen. To change the execution speed, press F1 (SPD).

```
[OPRT-AUTO] 100 0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

4) Enter the execution speed using the number keys and press ➡.

```
[OPRT-AUTO] 100 0:10
SPEED = _
(speed) 1→100
```

5) The screen returns to step 3.

```
[OPRT-AUTO] 50  0:10
001:MOVA 999,50
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

9-5 Emergency Stop Function

There are two ways to trigger emergency stop on the DRCX controller. One way is by using the push-button on the TPB. The other is to use the I/O emergency stop input. In either case for safety reasons, a contact B (normally closed) input is used (when the contact is opened, emergency stop is triggered). The DRCX controller can recover from an emergency stop condition without turning off the power so return-to-origin is not necessary.
This section explains how to initiate and recover from an emergency stop using the TPB.

9-5-1 Initiating an emergency stop

If for any reason you want to immediately stop the robot while operating it with the TPB, press the emergency stop button on the TPB. The emergency stop button locks in the depressed position, and can be released by turning it to the right.
In emergency stop, the robot assumes a "free" state so that commands initiating robot motion (for example, return-to-origin command) cannot be executed.

9-5-2 Recovering from an emergency stop

When recovery from an emergency stop is required during TPB operation, that procedrª% automatically appears on the TPB. Follow those instructions to reset the emergency stop condition.

Recovery from an emergency stop is required during TPB operation when you are going to:

- Perform return-to-origin.
- Run step operation.
- Run automatic operation.
- Edit point data using teaching playback.
- Exit the direct teaching mode.

The following steps explain the procedure for running step operation after emergency stop.
As this example shows, the emergency stop condition cannot be cancelled by just releasing the emergency stop button.

1) Press RUN to start operation.

```
[OPRT-STEP] 100 0: 7
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

2) Following the message displayed on the screen, release the emergency stop button.

```
[OPRT-STEP] 100 0: 7
release EMG.button
```

3) After the emergency stop is released, a message appears asking whether to turn the servo on.

   To turn the servo on, press **F1** (yes).

   To leave the servo off , press **F2** (no).

```
[OPRT-STEP] 100 0: 7
servo on ready ?

1yes 2no
```

4) Then, another message appears asking if ready to operate.

   To restart operation, press **F1** (yes).

   To cancel restarting, press **F2** (no).

```
[OPRT-STEP] 100 0: 7
continue OK ?

1yes 2no
```

5) Operation starts when **F1** (yes) was pressed in step 4.

   If **F2** (no) was pressed, the screen returns to step 1.

```
[OPRT-STEP] 100 0: 7
001:MOVA 254,100
 [    0.00,    0.00]
1SPD 2RSET3CHG 4next
```

CAUTION

When the SERVICE mode function is enabled, the following safety control will function. (See "10-4 SERVICE mode function".)

- ***If the hold-to-run function is enabled, the robot stops upon releasing F1 (yes) in step 4 in "SERVICE mode state".***

9-6 Displaying the Variables

The values of point data variable "P", counter array variable "C" and counter variable "D" can be displayed on the TPB screen.

1) On the initial screen, press **F2** (OPRT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Press **F2** (STEP) or **F3** (AUTO).
The following explains the procedure for displaying the variables on the screens in step operation.

```
[OPRT]
select menu

1ORG 2STEP3AUTO
```

3) Press **F4** (next) to change the menu display and then press **F1** (VAL).

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1VAL 2S_ON3CHGT4next
```

4) Continue to indicate the value of each variable.

The item enclosed by brackets [ ] is an array element number selected with the CSEL statement.

```
[OPRT-STEP] 100 0: 0
   P = 0
   C = 1     [0 ]
   D = 2
```

5) To return to the previous screen, press **ESC**.

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1VAL 2S_ON3CHGT4next
```

6) To display the variable of another task in a multi-task program, press **F3** (CHGT) in step 3 so that the task is switched before pressing **F1** (VAL).

```
[OPRT-STEP] 100 1:12
   P = 10
   C = 1000  [1 ]
   D = 2
```

9-7 Displaying the Memory I/O Status

The memory I/O status can be displayed on the screen.

1) On the initial screen, press F2 (OPRT).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Press F2 (STEP) or F3 (AUTO).
The STEP or AUTO mode screen appears. The following steps are explained using the STEP mode screen.

```
[OPRT]
select menu

1ORG 2STEP3AUTO
```

3) Press F4 (next) twice to change the menu display and then press F1 (MIO).

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1MIO 2    3    4next
```

4) The I/O status of each memory is displayed.
From the left, the top line shows the status from 115 to 100, the middle line from 131 to 116, and the bottom line from 147 to 132.

```
[OPRT-STEP] 100 0: 0
M  00000000 00000000
   00000000 00000000
   00000000 00000001
```

5) To return to the previous screen, press ESC.

```
[OPRT-STEP] 100 0: 0
001:MOVA 254,100
 [    0.00,    0.00]
1MIO 2    3    4next
```

9

OPERATING THE ROBOT

MEMO

Chapter 10 OTHER OPERATIONS

The TPB has many convenient functions in addition to those already covered. For example, memories can be initialized, and options such as memory cards can be used. This chapter will describe these additional functions

10-1 Initialization

Initializing the programs and points erases all the program data and point data currently stored in the controller.
Initializing the parameters resets the parameters to their initial values.

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F3 (INIT).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

3) Select the data to be initialized.

   To initialize the program data, press F1 (PGM).

   To initialize the point data, press F2 (PNT).

   To initialize the parameter data, press F3 (PRM).

   To initialize all of the program, point and parameter data, press F4 (ALL).

```
[SYS-INIT]
select menu

1PGM 2PNT 3PRM 4ALL
```

4) If F3 (PRM) or F4 (ALL) was selected in step 3, the robot type must be selected.

   When using a dual-axis robot, press F1 (XY).

   When using two single-axis robots (Multi-Flip), press F2 (FLIP).

```
[SYS-INIT-PRM]
select menu

1XY  2FLIP
```

5) If F1 (XY) was selected, enter the dual-axis robot type number with the number keys and then press the ⇨ key.

   For the robot number, see "15-1-2 Robot number list".

   The screen at the right is an example showing that the SXYx robot is selected.

```
[SYS-INIT-PRM-XY]
robot type : 110_
refer to
robot type table
```

6) Enter the X-axis stroke with the number keys and then press the ➡ key.

```
[SYS-INIT-PRM-XY]
robot type : 110
X stroke : 450_  [mm]
```

7) Enter the Y-axis stroke with the number keys and then press the ➡ key.

```
[SYS-INIT-PRM-XY]
robot type : 110
X stroke : 450   [mm]
Y stroke : 350_  [mm]
```

8) Next, enter the robot payload

Use the number keys to enter the payload and then press the ➡ key.

```
[SYS-INIT-PRM-XY]
robot type : 110
Weight  :  3_  [Kg]
```

9) A confirmation message appears on the screen.

To execute the initialization, press **F1** (yes).

To cancel the initialization, press **F2** (no).

```
[SYS-INIT-PRM-XY]
parameter data
initialize OK ?
1yes 2no
```

10) When the initialization is complete, the screen returns to step 3.

```
[SYS-INIT]
select menu

1PGM 2PNT 3PRM 4ALL
```

11) If **F2** (FLIP) was selected in step 4, the screen shown on the right appears.

Enter the type number of the robot connected to the X-axis with the number keys and then press the ➡ key.

For the robot number, see "15-1-2 Robot number list".

```
[SYS-INIT-PRM-FLIP]
X robot type: _
refer to
robot type table
```

12) When the robot having two or more different lead lengths was selected in step 11, the display changes to the lead length selection screen. Press **F1**, **F2** or **F3** key to select the ball screw lead of the robot connected as the X-axis.

If the robot has 4 or more different lead lengths, press **F4** (next) to switch the menu display.

```
[SYS-INIT-PRM-FLIP]
X robot type: 20
select X lead type
15.0 210.0320.0
```

10

OTHER OPERATIONS

13) A confirmation message appears after selecting the lead length.

Press **F1** (yes) when the setting is correct. To select another lead length, press **F2** (no).

```
[SYS-INIT-PRM-FLIP]
X robot type: 20
X lead   : 20.0 [mm]
1yes 2no
```

14) Next, enter the X-axis stroke length.

Use the number keys to enter the stroke length and then press the ⇨ key.

```
[SYS-INIT-PRM-FLIP]
X robot type: 20
X stroke : 550_ [mm]
```

15) Next, enter the X-axis payload.

Use the number keys to enter the payload and then press the ⇨ key.

The display then changes to the Y-axis setting screen. Repeat the same procedure from steps 11 to 15 to enter settings for the Y-axis.

```
[SYS-INIT-PRM-FLIP]
X robot type: 20
X stroke : 550  [mm]
X weight : 3_   [kg]
```

16) When data input for the two axes is complete, a confirmation message appears on the screen.

To execute the initialization, press **F1** (yes).

To cancel the initialization, press **F2** (no).

```
[SYS-INIT-PRM-FLIP]
parameter data
initialize OK ?
1yes 2no
```

17) When the initialization process ends, the screen returns to that shown at Step 3.

```
[SYS-INIT]
select menu

1PGM 2PNT 3PRM 4ALL
```

10-2 DIO Monitor Display

Data indicating whether the I/O signals are on or off can be displayed on the screen. The operation procedure is explained below.

10-2-1 Display from the monitor menu

1) On the initial screen, press F4 (MON).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) The ON/OFF status of I/O signals is displayed.

For information about what the display shows, refer to "4-3-4 DIO monitor screen".

```
DI 10000000 00000000
   10000000
DO 00000000 10100000
XO:1 YO:1 XS:1 YS:1
```

3) To return to the initial screen, press ESC.

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

10-2-2 Display from the DIO key operation

1) Hold down the DIO key.

```
[OPRT-AUTO]
running...
```

2) The ON/OFF status of I/O signals is displayed as long as the key is held down.

   For information about what the display shows, refer to "4-3-4 DIO monitor screen".

```
DI 10000000 00000000
   10000000
DO 00000000 11000000
XO:1 YO:1 XS:1 YS:1
```

3) Releasing the key returns the screen to the previous screen.

```
[OPRT-AUTO]
running...
```

⚠ *CAUTION*

The DIO Monitor key does not function during system operation.

10-3 System Information Display

1) On the initial screen, press the ESC key.

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

2) The controller version number, TPB version number, and robot type are displayed. The screen returns to the initial screen after approximately two seconds.

```
[INFORMATION]
controller  V18. 11
TPB         V 2. 10
robot type    20/20
```

10

OTHER OPERATIONS

10-4 SERVICE mode function

The SERVICE mode function is explained in this section.

The robot operator or others sometimes need to enter the hazardous area in the robot safety enclosure and move the robot to perform maintenance or adjustment while using the TPB. This situation is referred to as "SERVICE mode state" and requires extra caution. Limits should be placed on controller operation at this time to ensure operator safety.
A safety function called "SERVICE mode function" places limits on controller operation when in "SERVICE mode state".

When the SERVICE mode function is enabled, the DRCX controller constantly monitors status to check whether "SERVICE mode state" occurs. In "SERVICE mode state", the SERVICE mode function does the following:

- Limits command input from any device other than the TPB.
- Limits robot movement speed.
- Prohibits automatic operation and step operation.
- Enables hold-to-run function.

The controller recognizes "SERVICE mode state" when the SERVICE mode function is enabled and the SERVICE mode input (SVCE) is OFF (contact is open). (See "3-2-3 SERVICE mode input (SVCE)".)

NOTE
The SERVICE mode function is protected by a password so that the settings cannot be changed easily.

10-4-1 Safety settings for SERVICE mode

Safety controls that work in "SERVICE mode state" are explained in detail below.

■ Limiting command input from any device other than TPB

When the operator is working within the robot safety enclosure using the TPB, permitting any command input from devices (such as via I/O) other than the TPB is very hazardous to the TPB operator.

(For example, a hazardous situation may occur if someone outside the safety enclosure runs an automatic operation start command (AUTO-R) without letting the TPB operator know about it.)

To avoid this kind of hazard, the TPB can only be used to operate the robot in "SERVICE mode state", and all other device command inputs are disabled.

However, this limitation can be cancelled even in "SERVICE mode state" under the user's responsibility.

Setting value	Details
0	Only commands input from the TPB are permitted in SERVICE mode state.
1	Only commands input from the TPB and parallel I/O are permitted in SERVICE mode state.
2	Only commands input from the TPB and option unit are permitted in SERVICE mode state.
3	Command inputs are not limited even in SERVICE mode state.

■ Limiting the robot movement speed

Moving the robot at a high speed while an operator is working within the robot safety enclosure is very dangerous to that operator. Setting the robot movement speed to a safety speed of 250mm/s or less is advisable because most robot operation while the operator is working within the safety enclosure is for maintaining or adjusting the robot. In view of this, the robot movement speed in "SERVICE mode state" is limited to below 3% of maximum speed.

However, this speed limitation can be cancelled even in "SERVICE mode state" under the user's responsibility.

Setting value	Details
0	The robot movement speed is limited to 3% or less of maximum speed in SERVICE mode state.
1	The robot movement speed is not limited even in SERVICE mode state.

■ Prohibiting the automatic operation and step operation

Running an automatic operation or step operation while an operator is working within the robot safety enclosure is very dangerous to that operator.

(For example, when the operator is in the safety enclosure, a hazardous situation may occur if someone runs a robot program without letting the operator know about it.)

To avoid this kind of hazard, automatic operation and step operation are basically prohibited in "SERVICE mode state".

However, this limitation can be cancelled even in "SERVICE mode state" under the user's responsibility.

Setting value	Details
0	Automatic operation and step operation are prohibited in SERVICE mode state.
1	Automatic operation and step operation are permitted even in SERVICE mode state.

■ Hold-to-run function

If the robot continues to move while an operator is working within the robot safety enclosure without using the TPB, the operator may be exposed to a dangerous situation.

(For example, a hazardous situation may occur if the operator working within the safety enclosure should trip or fall by accident and blackout.)

To prevent this kind of hazard, the Hold-to-Run function allows the robot to move only during the time that the TPB key is kept pressed in "SERVICE mode state".

However, this function can be cancelled even in "SERVICE mode state" under the user's responsibility.

Setting value	Details
0	Hold-to-run function works in SERVICE mode state.
1	Hold-to-run function does not work even in SERVICE mode state.

CAUTION

The above safety controls can be cancelled in part or in whole under the user's responsibility. However, extra caution must be taken to maintain safety since hazardous situations may occur.

NOTE

When parameter initialization is performed, all safety control settings are initialized. (All settings will be set to "0".) However, the SERVICE mode function setting will not change even after parameter initialization.

10-4-2 Enabling/disabling the SERVICE mode function

To enable or disable the SERVICE mode function, follow these steps.

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

2) Press F4 (next) to change the menu display and then press F1 (SAFE).

```
[SYS]
select menu


1SAFE2OPT 3UTL 4next
```

3) The password request screen appears. Enter the password and then press ➡.

```
[SYS-SAFE]
 Password: 18.11_
input password
```

4) When the password is correct, the screen shown on the right appears.

   Press F2 (SVCE) here.

```
[SYS-SAFE]
select menu


1ACLV2SVCE
```

5) Press F1 (SET).

```
[SYS-SAFE-SVCE]
select menu


1SET 2DEV 3SPD 4next
```

6) The current SERVICE mode function setting appears.

   To disable the SERVICE mode function, enter 0 with the number key. To enable it, enter 1. Then, press ➡.

```
[SYS-SAFE-SVCE-SET]
SERVICE mode = 0
0:Invalid 1:Valid
```

7) When writing is complete, the screen returns to step 6.

```
[SYS-SAFE-SVCE-SET]
SERVICE mode = 1
0:Invalid 1:Valid
```

NOTE

The password is identical to the DRCX controller's version number. For example, if the controller version is 18.11, enter 18.11 as the password. Once the password is accepted, it will not be requested unless the TPB is disconnected from the controller or the controller power is turned off.

10-4-3 Setting the SERVICE mode functions

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Press F4 (next) to change the menu display and then press F1 (SAFE).

```
[SYS]
select menu

1SAFE2OPT 3UTL 4next
```

3) The password request screen appears. Enter the password and then press ⇨.

```
[SYS-SAFE]
 Password: 18.11_
input password
```

4) When the password is correct, the screen shown on the right appears.

   Press F2 (SVCE) here.

```
[SYS-SAFE]
select menu

1ACLV2SVCE
```

5) Select the item whose setting you want to change.

   To change the setting that limits the operation device, press F2 (DEV).

   To change the setting that limits the speed, press F3 (SPD).

   To change the setting that limits step operation and automatic operation, press F4 (next) and then press F1 (RUN).

   To change the setting for the hold-to-run function, press F4 (next) and then press F2 (HtoR).

```
[SYS-SAFE-SVCE]
select menu

1SET 2DEV 3SPD 4next
```

6) The current setting for the selected item appears.

   To change the setting, enter the data with the number key and then press ⇨.

```
[SYS-SAFE-SVCE-DEV]
data = 0
PB only
```

7) When the setting has been changed, the memory write screen appears.
To save the change permanently (retain the change even after the controller power is turned off), press F1 (SAVE).
To save the change temporarily (retain the change until the power is turned off), press F2 (CHG).
To cancel changing the setting, press F3 (CANCEL).

```
[SYS-SAFE-SVCE-DEV]
data = 1
PB/DI valid
1SAVE2CHG 3CANCEL
```

8) When writing is complete, the screen returns to step 6.

```
[SYS-SAFE-SVCE-DEV]
data = 1
PB/DI valid
```

NOTE

The password is identical to the DRCX controller's version number. For example, if the controller version is 18.11, enter 18.11 as the password. Once the password is accepted, it will not be requested unless the TPB is disconnected from the controller or the controller power is turned off.

10-5 System utilities

10-5-1 Viewing hidden parameters

Parameters hidden in the normal state can be viewed.
Use extra caution to avoid accidentally changing the parameters when these hidden parameters are displayed.

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Press F4 (next) to change the menu display and then press F3 (UTL).

```
[SYS]
select menu

1SAFE2OPT 3UTL 4next
```

3) Press F1 (HDPR) here.

```
[SYS-UTL]
select menu

1HDPR
```

4) A confirmation message appears.

   To permit display of the hidden parameters, press F1 (yes)

   To hide the hidden parameters, press F2 (no).

```
[SYS-UTL-HDPR]
hidden parameter
displayed ?
1yes 2no
```

5) The screen returns to step 3.

   Display of hidden parameters is permitted until you press F1 (HDPR) and then F2 (no), or the DRCX controller is turned off, or until the TPB is disconnected.

```
[SYS-UTL]
select menu

1HDPR
```

NOTE

The hidden parameter display is also permitted by turning on the power to the controller while holding down the ESC key on the TPB, or by connecting the TPB to the controller while holding down the ESC key.

10-6 Using a Memory Card

A memory card can be used with the TPB to back up the data in the DRCX controller.
Refer to "16-2-1 Memory card" for the procedure for handling a memory card and for the number of data that can be stored.

10-6-1 Saving controller data to a memory card

1) Insert the memory card into the TPB.

2) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

3) Next, press F2 (B.UP).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

4) Press F1 (SAVE).

```
[SYS-B.UP]
select menu

1SAVE2LOAD3FMT 4ID
```

5) Specify the format to save the data.
   Press F1 (normal) to save the data in the DRCX standard format.
   Press F2 (compati) to save the data in DRC and DRCA compatible format.

```
[SYS-B.UP-SAVE]
select save type

1normal 2compati
```

6) Specify the save area on the memory card.
   Enter the save area with the number keys and press ➡.

```
[SYS-B.UP-SAVE]
select card AREA
AREA _        [0-2]
1ID
```

10

OTHER OPERATIONS

7) The saved status of data on the memory card can be checked by pressing F1 (ID) in step 6.
To check the saved status in AREA 3 onward, press STEP UP or STEP DOWN to scroll the screen. To return to the screen in step 6, press ESC.

```
[SYS-B.UP-ID]
AREA 0 : 00.06.01
AREA 1 : 00.06.10
AREA 2 :
```

8) If data already exists in the area specified in step 6, a confirmation message appears.
To overwrite the data in the selected area, press F1 (yes).
To change the selected area, press F2 (no).

```
[SYS-B.UP-SAVE]
AREA 1 already saved
delete OK ?
1yes 2no
```

9) Set an ID number for the data being saved. Using the number keys (0 to 9), the "-" (minus) key, and the "." (period) key, enter a number of up to eight characters and then press ⇨.

```
[SYS-B.UP-SAVE]AREA1
make identification
 ID=_
Effective key[0→9-.]
```

10) A confirmation message appears.
To save the data, press F1 (yes).
To cancel, press F2 (no).

```
[SYS-B.UP-SAVE]AREA1
save OK ?
 ID=00.07.01
1yes 2no
```

11) This screen is displayed while the data is being saved.

```
[SYS-B.UP-SAVE]AREA1
saving ...
```

12) When saving is finished, the screen returns to step 4.

```
[SYS-B.UP]
select menu

1SAVE2LOAD3FMT 4ID
```

⚠ *CAUTION*

Never eject the memory card during saving of data.
Do not leave the memory card inserted into the TPB when not in use. This shortens the backup battery life.

10-6-2 Loading data from a memory card

1) Insert the memory card into the TPB.

2) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

3) Next, press F2 (B.UP).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

4) Press F2 (LOAD).

```
[SYS-B.UP]
select menu

1SAVE2LOAD3FMT 4ID
```

5) Specify the load area in the memory card.
   Enter the load area with the number keys and press ⇨.

```
[SYS-B.UP-LOAD]
select card AREA
AREA _         [0-2]
1ID
```

6) The saved status of data on the memory card can be checked by pressing F1 (ID) in step 5.
   To check the saved status in AREA 3 onward, press STEP UP or STEP DOWN to scroll the screen. To return to the screen in step 5, press ESC.

```
[SYS-B.UP-ID]
AREA 0 : 00.06.01
AREA 1 : 00.06.10
AREA 2 :
```

7) When the load area was selected in step 5, the data load screen appears. Select the data to be loaded.

   To load the program data, press F1 (PGM).

   To load the point data, press F2 (PNT).

   To load the parameter data, press F3 (PRM).

   To load all of the program, point and parameter data, press F4 (ALL).

```
[SYS-B.UP-LOAD]AREA3
Select menu

1PGM 2PNT 3PRM 4ALL
```

8) When F1 (PGM) or F2 (PNT) was selected in step 7, a confirmation message appears asking whether to overwrite the data.

   Pressing F1 (yes) overwrites the data only with the same program numbers or point numbers. (The previous data remains if its program or point number differs from the program or point number to be loaded.)

   Pressing F2 (no) loads the data after initializing the data in the DRCX controller.

   When F4 (ALL) was selected in step 7, all data in the DRCX controller will be initialized and then loaded.

```
[SYS-B.UP-LOAD]AREA3
program data
overwrite OK ?
1yes 2no
```

9) A confirmation message appears asking whether to load the data.

   To load the data, press F1 (yes)

   To cancel, press F2 (no).

```
[SYS-B.UP-LOAD]AREA3
program data
load OK ?
1yes 2no
```

10) This screen is displayed while data is being loaded.

```
[SYS-B.UP-LOAD]AREA3
loading ...
```

11) When data loading is complete, the screen returns to step 7.

```
[SYS-B.UP-LOAD]AREA3
Select menu

1PGM 2PNT 3PRM 4ALL
```

CAUTION

Never eject the memory card during loading of data.
Do not leave the memory card inserted into the TPB when not in use. This shortens the backup battery life.

10-6-3 Formatting a memory card

1) Insert the memory card into the TPB.

2) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

3) Next, press F2 (B.UP).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

4) Press F3 (FMT).

```
[SYS-B.UP]
select menu

1SAVE2LOAD3FMT 4ID
```

5) A confirmation message appears.
   To format the memory card, press F1 (yes).
   To cancel, press F2 (no).

```
[SYS-B.UP]
format OK ?

1yes 2no
```

6) Select the format type.
   To perform the DRCX standard formatting, press F1 (normal).
   To perform the DRC/DRCA compatible formatting, the press F2 (compati).

```
[SYS-B.UP-SAVE]
select format type

1normal 2compati
```

7) This screen is displayed while the memory card is being formatted.

```
[SYS-B.UP]
formatting
```

8) When formatting is complete, the screen returns to step 4.

```
[SYS-B.UP]
select menu


1SAVE2LOAD3FMT 4ID
```

CAUTION

Never eject the memory card during formatting.
Do not leave the memory card inserted into the TPB when not in use. This shortens the backup battery life.

10-6-4 Viewing the ID number for memory card data

1) Insert the memory card into the TPB.

2) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

3) Next, press F2 (B.UP).

```
[SYS]
select menu

1PRM 2B.UP3INIT
```

4) Press F4 (ID).

```
[SYS-B.UP]
select menu

1SAVE2LOAD3FMT 4ID
```

5) The ID number of each area is displayed on the screen.
   To check the saved status in AREA 3 onward, press STEP UP or STEP DOWN to scroll the screen. To return to the screen in step 4, press ESC.

```
[SYS-B.UP-ID]
AREA 0 : 00.06.01
AREA 1 : 00-06-10
AREA 2 : 00.07.11
```

10-7 Duty (load factor) monitor

The DRCX controller has a duty (load factor) monitor to allow you to operate the robot under the most optimal conditions. The duty monitor checks the robot's motor load factor and displays it in percent (%) versus the motor rating.

NOTE
The duty monitor function can be used when the controller version is 18.50 or later and the TPB version is 12.60 or later.

An overload error might appear if the duty exceeds 100% during robot operation. If this happens, either lower the robot acceleration or maximum speed, or increase the robot stop time (lower the duty ratio). On the other hand, if you want to shorten the cycle time even further, when there is currently no overload, you can raise the acceleration or maximum speed, or shorten the robot stop time (raise the duty ratio).

There are the following two methods to measure the duty.

Method 1: On the TPB, select DUTY mode and measure the duty during robot movement with a point movement command (ABS-PT, INC-PT) or a program start command (AUTO-R, STEP-R) via the I/O connector.

Method 2: Specify an interval in a program in which you want to measure the duty and run the program.

[Method 1]

1) Operate the robot with a point movement command (ABS-PT, INC-PT) or a program start command (AUTO-R, STEP-R) via the I/O connector.
2) On the TPB, select DUTY mode.
3) Measure the operation duty.
4) Check the measurement result.

Refer to "10-7-1 Measuring the duty (load factor)" for procedures to start and stop duty measurement and check the result.

NOTE
In method 1, the duty cannot be measured during robot movement by the TPB (RS-232C).

[Method 2]

1) Add the robot language command "DUTY 1" to the beginning of the interval in a program in which you want to measure the duty and also add the robot language command "DUTY 0" to the end of the interval.

```
            :
005:    DO 0,1
006:    WAIT 1,1
007:    DO 0,0
008:    TIMR 100
009:    DUTY 1      ← Start operation duty measurement   ↑
010:    DO 0,0
011:    WAIT 0,1
012:    MOVA 2,100
013:    DO 0,1                                            Operation duty
014:    WAIT 1,1                                          measurement interval
015:    DO 1,0
016:    TIMR 100
017:    DUTY 0      ← Stop operation duty measurement    ↓
018:    DO 0,0
            :
```

2) Run the program including the operation duty measurement interval.
3) Stop (end) the program.
4) On the TPB, select DUTY mode and check the measurement result.

Refer to "10-7-1 Measuring the duty (load factor)" for the procedure to check the measurement result.

10-7-1 Measuring the duty (load factor)

NOTE
The duty monitor function can be used when the controller version is 18.50 or later and the TPB version is 12.60 or later.

1) Press F4 (MON) on the TPB initial screen while moving the robot with a point movement command (ABS-PT, INC-PT) or a program start command (AUTO-R, STEP-R) via the I/O connector.

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F2 (DUTY).

```
[MON]
select menu

1DIO 2DUTY
```

3) Press F1 (RUN) to start measuring the operation duty.

```
[MON-DUTY]
select menu

1RUN 2STOP3RSLT
```

4) Press F2 (STOP) to stop measuring the operation duty.

```
[MON-DUTY]
select menu
measuring ...
1RUN 2STOP3RSLT
```

5) Next, press F3 (RSLT) to display the measurement result on the TPB screen.

```
[MON-DUTY]
select menu

1RUN 2STOP3RSLT
```

6) The operation duty value appears as a percent on the TPB screen.

```
[MON-DUTY-RSLT]
measuring result
X:   12%
Y:   30%
```

NOTE
The operation duty can also be monitored while the program is being executed with a program command. For more information, see "10-7 Duty (load factor) monitor". The method for displaying the measurement result is the same as described above.

10 OTHER OPERATIONS

MEMO

Chapter 11 COMMUNICATION WITH PC

The DRCX controller allows you to edit the program data and point data or control the robot operation using a PC (personal computer) by RS-232C communication instead of using the TPB.
This chapter describes how to set the communication parameters required to communicate between the PC and the DRCX controller, and also explains the communication command specifications.

11-1 Communication Parameter Specifications

The communication parameters on the PC should be set as follows. For the setting procedure, refer to the computer operation manual.

■ Baud rate	9600 bps
■ Data bit length	8 bits
■ Stop bit length	1 bit
■ Parity check	On
■ Parity setting	Odd
■ Control method (X parameter)	XON/XOFF software control (Effective)
■ Communication method	Full duplex
■ Sync method	Asynchronous method
■ Return key transmission	CR code
■ CR code reception	For CR/LF reception : Return + line feed For CR reception : Return

* If the above parameter settings are not possible due to your equipment specifications, the robot controller settings can be changed by changing PRM22 (communication parameter settings) from the TPB.

PRM22 settings (default value: 0)

bit	Function	Selection			
15 to 9	Reserved	Always set to 0.			
8	Termination code	0: CR + LF		1: CR	
7 to 4	Transmission speed	0: 9600bps	1: 300bps	2: 600bps	3: 1200bps
		4: 2400bps	5: 4800bps	6: 9600bps	7 to 15: Cannot be set
3	Data bit length	0: 8 bits		1: 7 bits	
2	Stop bit length	0: 1 bit		1: 2 bits	
1 to 0	Parity check	0: Odd	1: Even	2 to 3: Non	

Example: To set the data bit length to "7 bits" and the parity check to "Non", enter "10" for PRM22, which is given by
0000000000001010 (binary) = 10 (decimal)

CAUTION

Be sure to use a cable which conforms to specifications listed in "11-2 Communication Cable Specifications". The settings will be invalid if other cables such as POPCOM communication cables or those having different specifications are used.
After changing the parameters, turn the power off and then turn it on again to enable the settings.
The TPB can be used even if the parameters have been changed.

11-2 Communication Cable Specifications

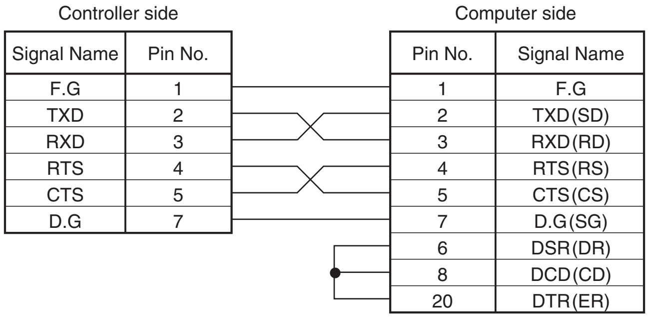
CAUTION

Pins 10, 12, 18 and 21 of the controller's connector are specifically used for TPB connection. To avoid possible accidents do not connect other inputs to these pins.
When using optional POPCOM software, make connections while referring to the POPCOM operation manual since it shows the different connection specifications.
The personal computer may have its own connector specifications, so be sure to check the computer operation manual to ensure the connections are correct.

11-2-1 Connecting to the computer with a 25-pin D-sub connector

Connector model

Mating connector type No.	: XM2A-2501	(OMRON) or equivalent type
Mating connector cover type No.	: XM2S-2511	(OMRON) or equivalent type



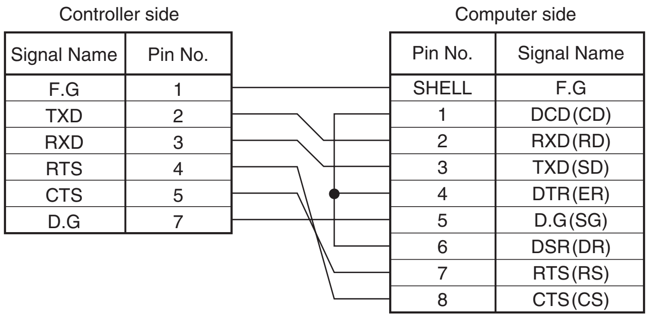
11-2-2 Connecting to the computer with a 9-pin D-sub connector

Connector model (controller side)

Mating connector type No.	: XM2A-2501	(OMRON) or equivalent type
Mating connector cover type No.	: XM2S-2511	(OMRON) or equivalent type

Connector model (computer side)

Mating connector type No.	: XM2D-0901	(OMRON) or equivalent type
Mating connector cover type No.	: XM2S-0913	(OMRON) or equivalent type



The "SHELL" is the metallic casing of the connector.

NOTE

Transmission stops while CTS on the controller side is off. If a robot alarm is issued while CTS is on, the controller keeps sending the message.
RTS on the controller side is always on.

11-3 Communication Command Specifications

On the DRCX controller, a command interface resembling the BASIC programming language is provided as standard, to facilitate easy communication with a PC. Communication commands are divided into the following four categories:

1. Robot movements
2. Data handling
3. Utilities
4. Special codes

Format: (except for special codes)

```
@<operation code> [<operand 1>][,<operand 2>][,<operand 3>]c/r l/f
```

- Basically, all of the commands begin with the start code '@' (=40H) and end with the code c/r (=0DH) l/f (=0AH). These two codes signal the controller that the statements between them constitute one command line. (The special codes are the only ones that do not require a start or an end code.)

- A communication command is basically composed of an operation code and an operand. Depending on the command statement, either no operand is used, or up to three operands are used.
  Items in [ ] (brackets) can be omitted.

- The character codes used in the DRCX series, are the JIS8 unit system codes (ASCII codes with katakana characters added). Input characters can be upper case or lower case.

- One or more space must be inserted between the operation code and the operand.

- Items with the < > marks should be specified by the user. Check the description of each communication command and enter the appropriate data. (Refer to "11-5 Communication Command Description".)

- When two or more operands are entered, insert a comma (,) between them.

An example is shown below.

Transmission example

```
@MOVI 123,100c/r l/f
```

Start code
Opcode (Operation code)
Space
Operand 1
Comma
Operand 2

11-4 Communication Command List

1. Robot movement

No.	Operation code	Operand 1	Operand 2	Operand 3	Command details
1.	ORG ORGN	[axis]			Returns all axes or specified axis to origin
2.	RESET				Resets program
3.	RUN				Starts automatic operation
4.	SRUN				Starts step operation
5.	SRVO	0 1	[axis] [axis]		Turns off the servo of all axes or a specified axis Turns on the servo of all axes or a specified axis
6.	X+/X-				Performs jog movement (inching) along X-axis
7.	Y+/Y-				Performs jog movement (inching) along Y-axis
8.	XINC/XDEC				Performs jog movement along X-axis
9.	YINC/YDEC				Performs jog movement along Y-axis
10.	MOVD	X-axis position (mm)	Y-axis position (mm)	speed	Directly moves to specified position
11.	MOVA	point number	speed		Moves to specified position
12.	MOVI	point number	speed		Moves specified movement amount
13.	MOVF	point number	DI number	0 or 1	Moves in response to general-purpose input
14.	MOVM	pallet work position	speed		Moves to specified pallet work position
15.	DRVD	axis	position (mm)	speed	Moves a specified axis to a specified coordinate position.
16.	DRVA	axis	point number	speed	Moves a specified axis to a specified point data position.
17.	DRVI	axis	point number	speed	Moves a specified axis a distance equal to specified point data.
18.	ACHA	axis	position		Defines an arch motion by setting a position.
19.	ACHI	axis	position		Defines an arch motion by setting a distance.
20.	DO	output number	0 1		Turns off general-purpose output or memory output Turns on general-purpose output or memory output
21.	WAIT	input number	0 or 1		Waits general-purpose input or memory input
22.	TIMR	time			Waits for specified time
23.	MAT	number of rows	number of columns	pallet number	Defines matrix on specified pallet
24.	MSEL	pallet number			Specifies pallet number where to move
25.	P	point number			Defines point variable P
26.	P+				Adds 1 to point variable P
27.	P-				Subtracts 1 from point variable P
28.	CSEL	array element number			Specifies array element of counter array variable C
29.	C	counter value			Defines counter array variable C
30.	C+	[addition value]			Adds specified value to counter array variable C
31.	C-	[subtraction value]			Subtracts specified value from counter array variable C
32.	D	counter value			Defines counter variable D
33.	D+	[addition value]			Adds specified value to counter variable D
34.	D-	[subtraction value]			Subtracts specified value from counter variable D
35.	SHFT	point number			Performs point data shift
36.	MOVL	point number	speed		Executes linear interpolation motion
37.	MOVC	point number	speed	locus type	Executes circular interpolation motion

2. Data handling

No.	Operation code	Operand 1	Operand 2	Operand 3	Command details
1.	?POS	[axis]			Reads current position of all axes or a specified axis
2.	?XPOS				Reads current position of X-axis
3.	?YPOS				Reads current position of Y-axis
4.	?NO				Reads current program number
5.	?SNO				Reads current step number
6.	?TNO				Reads current task number
7.	?PNO				Reads current point number
8.	?STP	program number			Reads total number of steps in specified program
9.	?MEM				Reads number of steps that can be added
10.	?VER				Reads ROM version number
11.	?ROBOT				Reads robot number
12.	?CLOCK				Reads total operation time of controller
13.	?ALM	history number	[display count]		Reads alarm history
14.	?EMG				Confirms emergency stop status
15.	?SRVO	[axis]			Confirms servo status of all axes or a specified axis
16.	?ORG	[axis]			Confirms return-to-origin status of all axes or a specified axis
17.	?XGRDP				Reads X-axis grid position (%)
18.	?YGRDP				Reads Y-axis grid position (%)
19.	?MODE				Confirms operation mode
20.	?MAT	pallet number			Reads matrix definition contents
21.	?MSEL				Reads currently specified matrix number
22.	?PVA				Reads current point variable P
23.	?CSEL				Reads currently specified element number of counter array variable C.
24.	?C	[array element number]			Reads current counter array variable C
25.	?D				Reads current counter variable D
26.	?SHFT				Reads current shift data
27.	?DI	input number			Reads general-purpose input or memory input status
28.	?DO	output number			Reads general-purpose output or memory output status
29.	?PRM	parameter number			Reads specified parameter data
		parameter number	parameter number		Reads specified multiple parameter data
30.	?P	point number			Reads specified point data
		point number	point number		Reads specified multiple point data
31.	READ	program number	step number	number of steps	Reads specified program data
		PGM			Reads all program data
		PNT			Reads all point data
		PRM			Reads all parameter data
		ALL			Batch reads all program, point and parameter data
		DIO			Reads input/output information
		MIO			Reads memory input/output information
		INF			Reads registered program information
32.	WRITE	PGM			Writes program data
		PNT			Writes point data
		PRM			Writes parameter data
		ALL			Batch writes program, point and parameter data
33.	?ERR	history number	[display count]		Reads error history

3. Utility

No.	Operation code	Operand 1	Operand 2	Operand 3	Command details
1.	INIT	PGM			Initializes program data
		PNT			Initializes point data
		PRM	dual-axis robot type		Initializes dual-axis robot parameters
		CPRM			Initializes common parameters
		XPRM	single-axis robot type		Initializes X-axis robot parameters
		YPRM	single-axis robot type		Initializes Y-axis robot parameters
		CLOCK			Initializes timer that measures total operation time
		ALM			Initializes alarm history
		ERR			Initializes error history
2.	SWI	program number			Switches program number to be run
3.	SWITSK	task number			Switches task number to be run
4.	SINS	program number	step number		Inserts one program step
5.	SDEL	program number	step number		Deletes one program step
6.	SMOD	program number	step number		Modifies one program step
7.	COPY	program number (copy source)	program number (copy destination)		Copies program
8.	DEL	program number			Deletes specified program
9.	PDEL	point number	number of points		Deletes point data

4. Special codes

No.	Code	Command details
1.	^C (=03H)	Interrupts RUN, SRUN, ORG, etc.
2.	^Z (=1AH)	Ends data transmission

11-5 Communication Command Description

11-5-1 Robot movements

(1) @ORG [axis]
@ORGN [axis]

This command performs return-to-origin on all axes or a specified axis, or checks whether return-to-origin is complete.
When the search method is selected as the origin detection method, this command performs return-to-origin and outputs the machine reference value after completing the return-to-origin. When the mark method is selected, this command checks whether return-to-origin is complete and outputs the machine reference value if complete, but issues an error if not complete.

Axis : Specify the axis. "0" for all axes, "1" for the X-axis, and "2" for the Y-axis.
If this setting is omitted, all axes are selected.

Transmission example : @ORG c/r l/f .... All axes return to their origin.

Response example 1 : OK c/r l/f
48% c/r l/f
52% c/r l/f
OK c/r l/f

Response example 2 : NG c/r l/f .... The robot is running.
31: running c/r l/f Execute the command again after stopping the robot.

Response example 3 : NG c/r I/f .... Return-to-origin is not complete on the axis using the
32: origin incomplete c/r l/f mark method.

NOTE
When the mark method is used for one axis and the search method for the other axis, then return-to-origin on the axis using the mark method must first be completed before performing return-to-origin on the axis using the search method. Use the TPB to perform return-to-origin on the axis using the mark method.

NOTE
Once return-to-origin is performed after the robot cable and absolute battery are connected, there is no need to repeat it even when the controller is turned off. (As an exception, return-to-origin becomes incomplete if the absolute backup function is disabled or a parameter relating to the origin is changed. Return-to-origin must be performed again in that case.)

CAUTION
When performing return-to-origin by the stroke-end detection method, do not interrupt the return-to-origin operation while detecting the origin (while contacting the mechanical limit). Otherwise, the operation will stop due to a controller overload alarm and the power will need to be turned on again.

CAUTION
If you must repeat return-to-origin using the stroke-end detection method, wait at least 5 seconds before repeating it.

(2) @RESET

This returns the program execution step to the first step of the program selected with the '@SWI' statement, and turns all general-purpose outputs (DO0 to DO12) and memory output off. The "current position in the program" used as a reference for the relative movement command (MOVI) is initialized to the current position of the robot, and the point variable P is also cleared to 0.

Transmission example	: @RESET c/r l/f		
Response example 1	: OK c/r l/f		
Response example 2	: NG c/r l/f		The robot is running.
	31: running	c/r l/f	Execute the command again after stopping the robot.

* When PRM2 ("operation at return-to-origin complete" parameter) is set to 1 or 3, DO5 does not turn off even if the @RESET command is executed. Likewise, when PRM21 (servo status output parameter) is set to 1, DO7 does not turn off even if the @RESET command is executed.

(3) @RUN

This executes a program all the way to the last step.
In the case of a multi-task program, all tasks are executed.

Transmission example	: @RUN c/r l/f		
Response example 1	: STOP c/r l/f		The last step of the program
	60: program end	c/r l/f	has been executed.
Response example 2	: NG c/r l/f		Return-to-origin has not been
	32: origin incomplete	c/r l/f	performed. Execute the command again after performing return-to-origin

CAUTION

When using an endless program (program that unconditionally returns to the head of the program at the last step), there will be no response.

(4) @SRUN

This executes only one step of a program.
In the case of a multi-task program, the selected task is executed.

Transmission example	: @SRUN c/r l/f		
Response example 1	: OK c/r l/f		
Response example 2	: STOP c/r l/f		The last step of the program
	60: program end	c/r l/f	has been executed.
Response example 3	: NG c/r l/f		Return-to-origin has not been
	32: origin incomplete	c/r l/f	performed. Execute the command again after performing return-to-origin.

(5) @SRVO <servo status> [,<axis>]

Controls the servo on/off operation of all axes or a specified axis.

Servo status	: Specify 1 to turn the servo on or 0 to turn it off.
Axis	: "0" for all axes, "1" for the X-axis, and "2" for the Y-axis. If this setting is omitted, all axes are selected.
Transmission example	: @SRVO 0 c/r l/f .... Turns off the servo of all axes.
Response example	: OK c/r l/f

(6) @X+, (@X-)

@X+ moves the robot to the + side and @X- to the - side based on the following equation.

Movement distance = 1 × (PRM12/100) (mm)   PRM12: Teaching movement data (%)

CAUTION

If the robot uses a rotary axis, the distance units are in degrees.

(7) @Y+, (@Y-)

This moves the Y-axis to the + or - side, based on the following equation.

Movement distance = 1 × (PRM12/100) (mm)   PRM12: Teaching movement data (%)

CAUTION

If the robot uses a rotary axis, the distance units are in degrees.

(8) @XINC, (@XDEC)

@XINC moves the robot to the + side and @XDEC to the - side at a speed calculated by the equation below. The robot continues moving until the ^C code is input or the robot reaches the soft limit.

Movement speed = 100 × (PRM12/100) (mm/sec.)   PRM12: Teaching movement data (%)

CAUTION

If the robot uses a rotary axis, then speed units are in deg/sec.

CAUTION

The soft limit will not work unless return-to-origin has been performed.

(9) @YINC, (@YDEC)

@YINC moves the Y-axis to the + side and @YDEC to the - side, at a speed calculated by the equation below. The robot continues moving until the ^C code is input or the robot arrives at the soft limit.

Movement speed = 100 × (PRM12/100) (mm/sec)   PRM12: Teaching movement data (%)

CAUTION

If the robot is a rotary axis, the speed units are in deg/sec.

CAUTION

The soft limit will not function unless return-to-origin has been performed.

(10) @MOVD <X-axis position (mm)> , <Y-axis position (mm)> , <speed>

Moves the robot to a specified coordinate position.

X(Y)-axis position	:	Directly specify the target position to move the robot to. If the robot uses a rotary axis, the coordinate position is expressed in deg. (degrees).
Speed	:	The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).
Transmission example	:	@MOVD 50.37,45.55,100 c/r l/f ..... Moves the robot to the position at X=50.37, Y=45.55, at 100% speed.
Response example 1	:	OK c/r l/f
Response example 2	:	NG c/r l/f .... The target position exceeds the soft limit. Change the target position or soft limit parameter. 30: soft limit over c/r l/f

(11) @MOVA <point number>,<speed>

Moves the robot to a position specified by a point number at a specified speed.

Point number	:	This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.
Speed	:	The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).
Transmission example	:	@MOVA 123,100 c/r l/f .... Moves the robot to point 123 at 100% speed.
Response example 1	:	OK c/r l/f
Response example 2	:	NG c/r l/f .... The target position exceeds the soft limit. Change the point data or soft limit parameter. 30: soft limit over c/r l/f

(12) @MOVI <point number>,<speed>

Moves the robot a distance specified by a point number from the current position, at a specified speed.

Point number	:	This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.
Speed	:	The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).
Transmission example	:	@MOVI 123,100 c/r l/f .... Moves the robot a distance defined by point 123, at 100% speed.
Response example 1	:	OK c/r l/f
Response example 2	:	NG c/r l/f .... The target position exceeds the soft limit. Change the point data or soft limit parameter. 30: soft limit over c/r l/f

CAUTION

When movement is interrupted with a stop (^C) statement, the current position in the program stays unchanged so that the movement can be resumed by executing the @MOVI command again. However, if the command is reset, the current position in the program is initialized to the actual robot position.

(13) @MOVF <point number>,<DI number>,<DI status>

This command moves the robot toward a position specified by a point number until a specified DI input condition is met. When the DI condition is met, the robot stops and the command terminates. Even if the DI condition is not met, the command terminates when the target point is reached.

Point number	:	This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.
DI number	:	Specify one of the general-purpose inputs DI0 to DI15.
DI status	:	Specify 1 (ON) or 0 (OFF) as the input condition.
Transmission example	:	@MOVF 2,10,1 c/r l/f .... Moves to point 2 until DI10 becomes 1 (ON).
Response example	:	OK c/r l/f

CAUTION

The movement speed is set with PRM4 (MOVF speed) and independent of the PRM17 setting (Maximum program speed).

(14)@MOVM <pallet work position>,<speed>

Moves the robot to a specified pallet work position at a specified speed.

Pallet work position : The pallet work position is a number used to identify each point on a matrix, and can be from 1 to 65025 (=255 × 255). The counter array variable C or counter variable D can also be used.

Speed : The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).

Transmission example : @MOVM 5,100 c/r l/f .... When a 4 × 3 matrix is defined, the robot moves to the point at "row 2, column 2" at 100% speed.

Response example 1 : OK c/r l/f

Response example 2 : NG c/r l/f .... Data error. The specified pallet work position is outside the matrix.
23: data error c/r l/f

CAUTION

Because calculation is performed on the assumption that the robot operates on a Cartesian coordinate system, the desired motion cannot be obtained if a SCARA robot is used.

(15)@DRVD <axis>, <position (mm)>, <speed>

Moves a specified axis to a specified coordinate position.

Axis : Specify the axis to be moved. "1" for the X-axis and "2" for the Y-axis.

Position : The position to be moved to is directly specified. If the robot uses a rotary axis, the movement position units are in degrees.

Speed : The speed can be set to any level between 1 and 100. When PRM17 (program execution speed) is set to 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).

Transmission example : @DRVD 1, 150.55, 100 c/r l/f ..... Moves the X-axis to the position at X=150.55 at 100% speed.

Response example 1 : OK c/r l/f

Response example 2 : NG c/r l/f .... The target position exceeds the soft limit. Change the point data or soft limit parameter.
30: soft limit over c/r l/f

(16)@DRVA <axis>, <point number>, <speed>

Moves a specified axis to a position (absolute position with respect to the origin) specified by a point number.

Axis	:	Specify the axis to be moved. "1" for the X-axis and "2" for the Y-axis.
Point number	:	This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.
Speed	:	The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).
Transmission example	:	@DRVA 1, 123, 100 c/r l/f .... Moves the X-axis to point 123 at 100% speed.
Response example 1	:	OK c/r l/f
Response example 2	:	NG c/r l/f .... The target position exceeds the 30: soft limit over c/r l/f soft limit. Change the point data or soft limit parameter.

(17)@DRVI <axis>, <point number>, <speed>

Moves a specified axis a distance equal to specified point data from the current position.

Axis	:	Specify the axis to be moved. "1" for the X-axis and "2" for the Y-axis.
Point number	:	This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.
Speed	:	The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 3000 rpm (when PRM64 and PRM104=3000).
Transmission example	:	@DRVI 2, 123, 100 c/r l/f .... Moves the Y-axis a distance equal to point 123 from the current position at 100% speed.
Response example 1	:	OK c/r l/f
Response example 2	:	NG c/r l/f .... The target position exceeds the 30: soft limit over c/r l/f soft limit. Change the point data or soft limit parameter.

⚠ ***CAUTION***

When movement is interrupted with a stop (^C) statement, the current position in the program stays unchanged so that the movement can be resumed by re-executing the @DRVI command. However, if the command is reset, the current position in the program is initialized to the robot position.

11

COMMUNICATION WITH PC

(18)@ACHA <axis>, <position>

Defines an arch motion by setting a position.

Axis : Specify the axis that performs an arch motion. "1" for the X-axis and "2" for the Y-axis.

Position : This is the position (absolute position with respect to the origin) the arch motion axis moves to. The setting range is from -9999 to 9999 mm.

Transmission example : @ACHA 2, 10 c/r l/f .... Defines an arch motion in which the Y-axis temporarily returns to Y=10.00.

Response example : OK c/r l/f

(19)@ACHI <axis>, <distance>

Defines an arch motion by setting a distance.

Axis : Specify the axis that performs an arch motion. "1" for the X-axis and "2" for the Y-axis.

Distance : This is the distance (relative distance with respect to the current position) the arch motion axis moves. The setting range is from -9999 to 9999 mm.

Transmission example : @ACHI 2, -100 c/r l/f .... Defines an arch motion in which the Y-axis temporarily returns a distance of -100.00.

Response example : OK c/r l/f

⚠ ***CAUTION***

When movement is interrupted with a stop (^C) statement, the current position in the program stays unchanged so that the movement can be resumed by re-executing the movement command. However, if the command is reset, the current position in the program is initialized to the robot position.

(20)@DO <general-purpose output or memory output number>,<output status>

Turns a general-purpose output or memory output on or off.

Output number : Specify one of the general-purpose outputs from 0 to 12 (13 points) or one of the memory outputs from 100 to 131 (32 points).

Output status : Specify 1 (ON) or 0 (OFF).

Transmission example : @DO 3,1 c/r l/f .... Turns on general-purpose output 3.

Response example : OK c/r l/f

(21)@WAIT <general-purpose input or memory input number>,<input status>

Waits until a specified general-purpose input or memory input is switched to a specified status.

Input number	:	Specify one of the general-purpose inputs from 0 to 15 (16 points) or one of the memory inputs from 100 to 147 (48 points).
Input status	:	Specify 1 (ON) or 0 (OFF).
Transmission example	:	@WAIT l,l c/r l/f .... Waits until DI1 becomes 1 (ON).
Response example	:	OK c/r l/f

(22)@TIMR <time>

Waits a specified amount of time.

Time	:	Set the time between 1 and 65535 in units of 10ms.
Transmission example	:	@TIMR 100 c/r l/f .... Waits one second.
Response example	:	OK c/r l/f

(23)@MAT <number of rows>,<number of columns>,<pallet number>

Defines a matrix.

Number of rows	:	Set the number of rows from 1 to 255.
Number of columns	:	Set the number of columns from 1 to 255.
Pallet number	:	The pallet number is a number used to identify each matrix (pallet) and can be from 0 to 31.
Transmission example	:	@MAT 5,2,1 c/r l/f .... Defines a matrix of 5 × 2 on pallet number 1.
Response example	:	OK c/r l/f

(24)@MSEL <pallet number>

Specifies a matrix where the robot moves with a MOVM statement.

Pallet number	:	The pallet number is a number used to identify each matrix (pallet) and can be from 0 to 31.
Transmission example	:	@MSEL 0 c/r l/f .... Specifies pallet number 0.
Response example	:	OK c/r l/f

(25) @P <point number>

Sets the point variable P.

Point number	: This can be any value from 0 to 999.
Transmission example	: @P 100 c/r l/f .... Set the point variable P to 100.
Response example	: OK c/r l/f

⚠ *CAUTION*

The contents of the point variable P are held even when the DRCX is turned off. However, when the program is reset or when the program reset is applied for example by switching the execution program, the point variable P will be initialized to 0.

(26) @P+

Adds 1 to the point variable P.

Transmission example	: @P+ c/r l/f .... Increments the point variable P. (P ← P+1)
Response example	: OK c/r l/f

(27) @P-

Subtracts 1 from the point variable P.

Transmission example	: @P- c/r l/f .... Decrements the point variable P. (P ← P-1)
Response example	: OK c/r l/f

(28) @CSEL <array element number>

Specifies an array element for the counter array variable C to be used.

Array element number	: This is a number used to designate an array element for the counter array variable C, and can be from 0 to 31. The counter variable D can also be specified here as the array element.
Transmission example	: @CSEL 1 c/r l/f .... Uses the counter array variable C of element number 1 in the subsequent steps.
Response example	: OK c/r l/f

(29) @C <counter value>

Sets a specified value in the counter array variable C specified with the CSEL statement.

Counter value	: This can be any value from 0 to 65535.
Transmission example	: @C 100 c/r l/f .... Sets the counter array variable C to 100.
Response example	: OK c/r l/f

(30) @C+ [<addition value>]

Adds a specified value to the counter array variable C.

Addition value : This can be any value from 1 to 65535. If this value is omitted, then 1 is added to the counter array variable.

Transmission example : @C+ c/r l/f .... Increments the counter array variable C. (C ← C+1)

Response example : OK c/r l/f

(31) @C- [<subtraction value>]

Subtracts a specified value from the counter array variable C.

Subtraction value : This can be any value from 1 to 65535. If this value is omitted, then 1 is subtracted from the counter array variable.

Transmission example : @C- c/r l/f .... Decrements the counter array variable C. (C ← C-1)

Response example : OK c/r l/f

(32) @D <counter value>

Sets a specified value in the counter variable D.

Counter value : This can be any value from 0 to 65535.

Transmission example : @D 100 c/r l/f .... Sets the counter variable D to 100.

Response example : OK c/r l/f

(33) @D+ [<addition value>]

Adds a specified value to the counter variable D.

Addition value : This can be any value from 1 to 65535. If this value is omitted, then 1 is added to the counter variable.

Transmission example : @D+ c/r l/f .... Increments the counter variable D. (D ← D+1)

Response example : OK c/r l/f

(34) @D- [<subtraction value>]

Subtracts a specified value from the counter variable D.

Subtraction value : This can be any value from 1 to 65535. If this value is omitted, then 1 is subtracted from the counter variable.

Transmission example : @D- c/r l/f .... Decrements the counter variable D. (D ← D-1)

Response example : OK c/r l/f

(35)@SHFT <point number>

Shifts the position data by an amount equal to the distance defined by a specified point number. The shifted data is valid until the SHFT statement is executed again or until the program is reset.

Point number : This is a number used to identify each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable P can also be used.

Transmission example : @SHFT 1 c/r l/f .... Shifts the point data by an amount defined by point number 1 and the shifted data is used with the subsequent movement commands.

Response example : OK c/r l/f

CAUTION

When the program is reset, the shift data will be initialized to (0.00, 0.00).
The movement commands affected by the execution of the SHFT command are MOVA, MOVF, MOVM, DRVA, ACHA, MOVL and MOVC. MOVD, DRVD, MOVI, DRVI and ACHI are not affected.

(36)@MOVL <point number>,<speed>

Moves to the position specified by a point number in a linear interpolation motion.

Point number : This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.

Speed : The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 1000mm/s (when PRM64 and PRM104 are set to 3000 and the ball screw lead is 20mm).

Transmission example : @MOL 123,100 c/r l/f .... Moves to point 123 in a linear interpolation motion at 100% speed.

Response example 1 : OK c/r l/f

Response example 2 : NG c/r l/f .... The target position exceeds the soft limit. Change the point data or soft limit parameter.
30: soft limit over c/r l/f

CAUTION

Interpolation movement cannot resume even if it is not yet finished. So, if interpolation movement is stopped by an interlock signal, the program should be reset and restarted from the beginning.
Linear interpolation movement cannot be performed on SCARA type robots which do not operate on a rectangular coordinate system.

(37) @MOVC <point number>, <speed>, <locus type>

Performs a circular interpolation motion passing through the position specified by a point number. For example, when the circular segment is selected as the locus type and the point number is specified as n, the robot moves from the current position along a circular segment locus with the end point of point n+1, passing through point n. In the case of a circular locus, the robot moves from the current position along a circular locus with the same start and end points, passing through points n and n+1.

Point number : This is a number assigned to each point (position data) and can be from 0 to 999 (a total of 1,000 points). Data for the point numbers can be edited with the @WRITE PNT statement. The point variable "P" can also be used.

Speed : The speed can be set to any level between 1 and 100. If PRM17 (Maximum program speed) is 100, then 100 will be equal to 1000mm/s (when PRM64 and PRM104 are set to 3000 and the ball screw lead is 20mm).

Locus type : Select the locus type. "0" for the circular segment locus and "1" for the circular locus.

Transmission example : @MOVC 123,100,0 c/r l/f .... Moves from the current position along a circular segment locus with point 124 specified as the end point, passing through point 123 at 100% speed.

Response example 1 : OK c/r l/f

Response example 2 : NG c/r l/f .... The target position exceeds the soft limit. Change the point data or soft limit parameter.
30: soft limit over c/r l/f

CAUTION

The radius that allows interpolation motion is 1000mm maximum and 2mm minimum.
Interpolation motion cannot resume if interrupted before completion. So, if interpolation motion is stopped by an interlock signal, the program must be reset and restarted from the beginning.
SCARA robots cannot move in a circular interpolation motion because they do not operate on a Cartesian coordinate system.

11-5-2 Data handling

(1) @?POS [<axis>]

Reads the current position of all axes or a specified axis.

Axis : "0" for all axes, "1" for the X-axis, and "2" for the Y-axis.
If this setting is omitted, X-axis is selected.

Transmission example : @?POS 0 c/r l/f

Response example : 321.05, 100.15 c/r l/f
OK c/r l/f

(2) @XPOS

This reads the current position of the X-axis.

Transmission example : @?XPOS c/r l/f

Response example : 321.05 c/r l/f
OK c/r l/f

(3) @YPOS

This reads the current position of the Y-axis.

Transmission example : @?YPOS c/r l/f

Response example : 100.15 c/r l/f
OK c/r l/f

(4) @?NO

Reads the current program number. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example : @?NO c/r l/f

Response example 1 : 31 c/r l/f .... Program No.31 is being executed.
OK c/r l/f

Response example 2 : 10/l c/r l/f .... Program No.1 is the lead program (program selected with @SWI statement), and program No.10 is currently being executed with the JMP or CALL statement, etc.
OK c/r l/f

(5) @?SNO

Reads the current step number. The @RUN and @SRUN commands are executed from the step read here. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example : @?SNO c/r l/f

Response example : 170 c/r l/f
OK c/r l/f

(6) @?TNO

Reads the current task number.

Transmission example : @?TNO c/r l/f

Response example : 0 c/r l/f .... Task 0 (main task) is currently selected.
OK c/r l/f

(7) @?PNO

Reads the currently selected point number. This is used to find which point data is being used for movement, or to find the point that caused an error if it occurs. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example : @? PNO c/r l/f

Response example : 57 c/r l/f
OK c/r l/f

(8) @?STP <program number>

Reads the total number of steps in the specified program.

Program number : This is a number used to identify each program and can be 0 to 99 (a total of 100).

Transmission example : @?STP 10 c/r l/f .... Reads the total number of steps for program No. 10.

Response example : 140 c/r l/f
OK c/r l/f

(9) @?MEM

Reads the number of steps that can be added.

Transmission example : @?MEM c/r l/f

Response example : 1001 c/r l/f
OK c/r l/f

CAUTION

In addition to the number of existing steps, the steps equivalent to the number of programs are used internally as the program control steps. For example, if one program with 50 steps is registered, the number of the available remaining steps will be 2949 steps (3000 - 1- 50 = 2949).

(10) @?VER

Reads the ROM version in the DRCX controller.

Transmission example	:	@?VER c/r l/f	
Response example	:	18.11 c/r l/f OK c/r l/f	

(11) @?ROBOT

Reads the type of the robot currently specified.

Transmission example	:	@?ROBOT c/r l/f	
Response example 1	:	SXYt c/r l/f .... OK c/r l/f	In the case of dual-axis robot setting, the robot name is displayed.
Response example 2	:	20/21 c/r l/f .... OK c/r l/f	In the case of Multi-Flip setting, the robot number of each axis is displayed.

(12) @?CLOCK

Reads the total operation time of the DRCX controller.

Transmission example	:	@?CLOCK c/r l/f	
Response example	:	00101,05:11:12 c/r l/f .... OK c/r l/f	Indicates that the total operation time is 101 days, 5 hours, 11 minutes and 12 seconds.

(13) @?ALM <history number>[,<display count>]

Displays a specified number of past alarms, starting from a specified history number.
A maximum of 100 past alarms can be displayed.
This alarm history shows the time (total elapsed time from controller start-up) that each alarm occurred and a description of the alarm.

History number	:	This number is assigned to each alarm sequentially from 0 to 99 in the order the alarms occurred. History number 0 indicates the most recent alarm that occurred. A larger history number indicates it is an older alarm.	
Display count	:	Specify the number of alarms you want to display from 1 to 100. If this entry is omitted, only one alarm is displayed.	
Transmission example	:	@?ALM 0,2 c/r l/f	Displays the two most recent alarms that occurred.
Response example	:	00101,05:11:12,X04: POWER DOWN c/r l/f 00096,18:10:02,X04: POWER DOWN c/r l/f OK c/r l/f	The most recent alarm that occurred was a voltage drop alarm occurring 101 days, 5 hours, 11 minutes and 12 seconds after the DRCX controller has started. The next most recent alarm was a voltage drop alarm occurring 96 days, 18 hours, 10 minutes and 2 seconds after the DRCX controller has started.

(14)@?EMG

Reads the emergency stop status.

Transmission example	: @?EMG c/r l/f	
Response example 1	: 0 c/r l/f .... OK c/r l/f	Emergency stop is off.
Response example 2	: 1 c/r l/f .... OK c/r l/f	Emergency stop is on.

(15)@?SRVO [<axis>]

Reads the servo state of all axes or a specified axis.
When all axes are specified, the response results in "1" only when the servo of all axes has been on.

Axis	: "0" for all axes, "1" for the X-axis, and "2" for the Y-axis. If this setting is omitted, all axes are selected.	
Transmission example	: @?SRVO c/r l/f	
Response example 1	: 0 c/r l/f .... OK c/r l/f	Servo is off.
Response example 2	: 1 c/r l/f .... OK c/r l/f	Servo is on.

(16)@?ORG [<axis>]

Reads whether or not return-to-origin has been completed on all axes or specified axis.
When all axes are specified, the response results in "1" only when return-to-origin has been completed on all axes.

Axis	: "0" for all axes, "1" for the X-axis, and "2" for the Y-axis. If this setting is omitted, all axes are selected.	
Transmission example	: @?ORG c/r l/f	
Response example 1	: 0 c/r l/f .... OK c/r l/f	Return-to-origin not completed.
Response example 2	: 1 c/r l/f .... OK c/r l/f	Return-to-origin completed.

(17)@?XGRDP

Reads the X-axis grid position (machine reference when return-to-origin is completed).

Transmission example	: @?XGRDP c/r 1/f
Response example	: 50% c/r 1/f OK c/r l/f

CAUTION

The response value will be meaningless if return-to-origin is not completed. Always transmit the command after return-to-origin has been completed.

(18)@?YGRDP

Reads the Y-axis grid position (machine reference when return-to-origin is completed).

Transmission example : @?YGRDP c/r 1/f

Response example : 52% c/r 1/f
OK c/r l/f

CAUTION
The response value will be meaningless if return-to-origin is not completed. Always transmit the command after return-to-origin has been completed.

(19)@?MODE

Reads the robot status.

Transmission example : @?MODE c/r l/f

Response example 1 : 0 c/r l/f .... Robot is stopped.
OK c/r l/f

Response example 2 : 1 c/r l/f .... Program is being executed from TPB or PC.
OK c/r l/f

Response example 3 : 2 c/r l/f .... Program is being executed by I/O command.

(20)@?MAT <pallet number>

Reads the matrix data on a specified pallet.

Pallet number : This is a number used to identify each matrix (pallet) and can be from 0 to 31.

Transmission example : @?MAT 1 c/r l/f .... Reads the matrix data on pallet number 1.

Response example : 20,30 c/r l/f
OK c/r l/f

(21)@?MSEL

Reads the pallet number for the currently specified matrix. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example : @?MSEL c/r l/f

Response example : 0 c/r l/f
OK c/r l/f

(22) @?PVA

Reads the point variable P. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example	:	@?PVA c/r l/f
Response example	:	0 c/r l/f OK c/r l/f

CAUTION

The contents of the point variable P are held even when the DRCX is turned off. However, when the program is reset or when the program reset is applied for example by switching the execution program, the point variable P will be initialized to 0.

(23) @?CSEL

Reads the currently specified element number of the counter array variable C. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example	:	@?CSEL c/r l/f
Response example	:	0 c/r l/f OK c/r l/f

(24) @?C [<array element number>]

Reads the value in the counter array variable C of the specified element number.

Element number	:	This is a number used to specify each array element and can be from 0 to 31. If this entry is omitted, the element number selected with the @CSEL command is used.
Transmission example	:	@?C c/r l/f OK c/r l/f
Response example	:	21202 c/r l/f OK c/r l/f

(25) @?D

Reads the counter variable D.

Transmission example	:	@?D c/r l/f
Response example	:	21202 c/r l/f OK c/r l/f

(26) @?SHFT

Reads the shift data currently set. In multi-task operation, this command reads the program information on the task currently selected.

Transmission example	:	@?SHFT c/r l/f
Response example	:	150.00, 250.00 c/r l/f OK c/r l/f

(27) @?DI <general-purpose input or memory input number>

Reads the status of a general-purpose input or memory input.

Input number	: Specify one of the general-purpose inputs 0 to 15 (16 points) or one of the memory inputs 100 to 147 (48 points).
Transmission example	: @?DI 1 c/r l/f
Response example 1	: 0 c/r l/f .... Input status is off. OK c/r l/f
Response example 2	: 1 c/r l/f .... Input status is on. OK c/r l/f

(28) @?DO <general-purpose output or memory output number>

Reads the status of a general-purpose output or memory output.

Output number	: Specify one of the general-purpose outputs 0 to 12 (13 points) or one of the memory outputs 100 to 131 (32 points).
Transmission example	: @?DO 2 c/r l/f
Response example 1	: 0 c/r l/f .... Output status is off. OK c/r l/f
Response example 2	: 1 c/r l/f .... Output status is on. OK c/r l/f

(29-1) @?PRM <parameter number>

Reads the data from a specified parameter.

Parameter number	: This is a number used to identify each parameter and can be from 0 to 127.
Transmission example	: @?PRM48 c/r l/f .... Reads the data from PRM48 (parameter 48).
Response example 1	: 350 c/r l/f OK c/r l/f
Response example 2	: c/r l/f .... No data is registered in PRM1 (parameter 48). OK c/r l/f

(29-2) @?PRM <parameter number>,<parameter number>

Reads multiple parameter data from the first parameter number to the second parameter number. If unregistered parameters exist, they will be skipped.

Parameter number	: This is a number used to identify each parameter and can be from 0 to 127.
Transmission example	: @?PRM48,52 c/r l/f .... Reads the data from PRM48 to PRM52 (parameters 48 to 52).
Response example	: PRM48=100 c/r l/f PRM49=3 c/r l/f PRM50=10 c/r l/f PRM51=100 c/r l/f PRM52=1 c/r l/f OK c/r l/f

(30-1) @?P <point number>

Reads the data of a specified point.

Point number	: This is a number used to identify each point data and can be from 0 to 999.	
Transmission example	: @?P 254 c/r l/f	Reads the data of point 254.
Response example 1	: -0.05,0.01 c/r l/f OK c/r l/f	
Response example 2	: c/r l/f .... OK c/r l/f	No data is registered in the specified point.

(30-2) @?P <point number>,<point number>

Reads multiple point data from the first point number to the second point number. If unregistered points exist, they will be skipped.

Point number	: This is a number used to identify each point data and can be from 0 to 999.	
Transmission example	: @?P15,22 c/r l/f	Reads the data from points 15 to 22.
Response example	: P15=100.00,200.00 c/r l/f P16=32.11,50.10 c/r l/f P20=220.00,250.27 c/r l/f P22=0.50,2.11 c/r l/f OK c/r l/f	

(31-1) @READ <program number>,<step number>,<number of steps>

Reads a specified number of step data from a specified step in a program. If the number of steps from the specified step to the final step is less than the number of steps specified here, the command execution will end when the final step is read out.

Program number	: This is a number used to identify each program and can be from 0 to 99.	
Step number	: This is a number assigned to each step and can be from 1 to 255.	
Number of steps	: Any number between 1 and 255 can be specified.	
Transmission example	: @READ 3,50,1 c/r l/f	Reads one step of data from step 50 in program No. 3.
Response example 1	: MOVA 29,100 c/r l/f ^Z(=1AH) OK c/r l/f	
Response example 2	: NG c/r l/f .... 42: cannot find step c/r l/f	The specified step number is not registered

(31-2) @READ PGM

Reads all of the program data.

Transmission example : @READ PGM c/r l/f

Response example : NO0 c/r l/f
MOVA 0,100 c/r l/f
JMPF 0,31,13 c/r l/f
NO31 c/r l/f
STOP c/r l/f
^Z (=1AH)
OK c/r l/f

(31-3) @READ PNT

Reads all point data.

Transmission example : @READ PNT c/r l/f

Response example : P0=0.00,0.00 c/r l/f
P1=350.00,100.00 c/r l/f
P2=196.47,201.15 c/r l/f
P254=-0.27,10.01 c/r l/f
^Z(=1AH)
OK c/r l/f

(31-4) @READ PRM

Reads all parameter data.

Transmission example : @READ PRM c/r l/f

Response example : PRM0=4 c/r l/f
PRM1=0 c/r l/f
:
:
PRM119=8 c/r l/f
^Z(=1AH)
OK c/r l/f

(31-5) @READ ALL

Reads all data (parameters, programs, points) at one time. Each data group (parameters, programs, points) is separated by an empty line (a carriage return only).

Transmission example : @READ ALL c/r l/f

Response example : PRM0=4 c/r l/f
PRM1=1 c/r l/f
:
:
PRM117=0 c/r l/f
c/r l/f
NO0 c/r l/f
MOVA 0,100 c/r l/f
MOVA 1,100 c/r l/f
NO10 c/r l/f
CALL 0, 10 c/r l/f
STOP c/r l/f
c/r l/f
P0=0.00,0.00 c/r l/f
P1=550.00,350.00 c/r l/f
^Z (=1AH)
OK c/r l/f

(31-6) @READ DIO

Reads the on/off status of DIO. Refer to "4-3-4 DIO monitor screen".

Transmission example : @READ DIO c/r l/f

Response example : D I 00000000 00000000 c/r l/f
10000000 c/r l/f
DO 00000000 11000000 c/r l/f
XO:0 YO:1 XS:1 YS:1 c/r l/f
OK c/r l/f

(31-7) @READ MIO

Reads the on/off status of memory I/O. From the left, the top line shows MIO numbers from 115 to 100, the middle line from 131 to 116, and the bottom line from 147 to 132.

Transmission example : @READ MIO c/r l/f

Response example : M 00000000 00000000 c/r l/f
00000000 00000000 c/r l/f
00000000 00000001 c/r l/f
OK c/r l/f

(31-8) @READ INF

Reads the status of the registered programs. The registered program numbers and number of steps are displayed.

Transmission example : @READ INF c/r l/f

Response example : NO0- 43 steps c/r l/f
NO1- 52 steps c/r l/f
NO31- 21 steps c/r l/f
^Z (=1AH)
OK c/r l/f

(32-1) @WRITE PGM

Writes the program data. The controller will transmit READY when this command is received. Confirm that READY is received and then transmit the program data. Always transmit ^Z (=1AH) at the end of the data.

Transmission example	Send	Receive
	@WRITE PGM c/r l/f	
		READY c/r l/f
	NO0 c/r l/f	
	MOVA 0,100 c/r l/f	
	JMPF 0,31,12 c/r l/f	
	NO31 c/r l/f	
	STOP c/r l/f	
	^Z(=1AH)	
		OK c/r l/f

⚠ ***CAUTION***

When @WRITE PGM is executed, the previous data of the same program number is overwritten. (The previous data remains as long as its program number differs from the program number to be written.)

(32-2) @WRITE PNT

Writes the point data. The controller will transmit READY when this command is received. Confirm that READY is received and then transmit the point data. Always transmit ^Z (=1AH) at the end of the data.

Transmission example	Send	Receive
	@WRITE PNT c/r l/f	
		READY c/r l/f
	P0=0.00,0.00 c/r l/f	
	P1=350.00,100.00 c/r l/f	
	P254=-0.27,10.01 c/r l/f	
	^Z(=1AH)	
		OK c/r l/f

⚠ ***CAUTION***

When @WRITE PNT is executed, the previous data of the same point number is overwritten. (The previous data remains as long as its point number differs from the point number to be written.)

(32-3) @WRITE PRM

Writes the parameter data. The controller will transmit READY when this command is received. Confirm that READY is received and then transmit the parameter data. Always transmit ^Z (=1AH) at the end of the data.

Transmission example	Send	Receive
	@WRITE PRM c/r l/f	
		READY c/r l/f
	PRM48=550 c/r l/f	
	PRM49=10 c/r l/f	
	^Z(=1AH)	
		OK c/r l/f

⚠ ***CAUTION***

Loading unsuitable robot data to the DRCX can inhibit the robot controller performance, possibly resulting in failures, malfunctions, and errors.

(32-4) @WRITE ALL

Writes all data (parameters, programs and points) at one time. The controller will transmit READY when this command is received. Confirm that READY is received and then transmit all data. Always transmit ^Z (=1AH) at the end of the data.

Transmission example :

Send	Receive
@WRITE ALL c/r l/f	
	READY c/r l/f
PRM0=8 c/r l/f	
PRM1=0 c/r l/f	
c/r l/f	
NO10 c/r l/f	
CALL 0, 20 c/r l/f	
STOP c/r l/f	
c/r l/f	
P1=550.00,300.00 c/r l/f	
^Z(=1AH)	
	OK c/r l/f

CAUTION

- ***Always place one or more empty line to separate between each data group (parameters, programs, points).***
- ***There is no specific rule in the data group sequence. There can be data groups that are not written in.***
- ***When @WRITE ALL is executed, the previous data of the same program number or point number is overwritten. (The previous data remains as long as its program number or point number differs from the program number or point number to be written.)***
- ***Loading unsuitable robot data to the DRCX can inhibit the robot controller performance, possibly resulting in failures, malfunctions, and errors.***

(33) @?ERR <history number>[,<display count>]

Displays a specified number of past errors, starting from a specified history number.
A maximum of 100 past errors can be displayed.
This error history shows the time (total elapsed time from controller start-up) that each error occurred and a description of the error.
This command is available with controller version 18.50 or later.

History number : This number is assigned to each error sequentially from 0 to 99 in the order the errors occurred. History number 0 indicates the most recent error that occurred. A larger history number indicates it is an older error.

Display count : Specify the number of errors you want to display from 1 to 100. If this entry is omitted, only one error is displayed.

Transmission example : @?ERR 0,2 c/r l/f .... Displays the two most recent errors that occurred.

Response example :
00:00101,05:11:12,PIO,52 : NO POINT DATA c/r l/f
01:00096,18:10:02,CMU,30: SOFT LIMIT OVER c/r l/f
OK c/r l/f .... The most recent error that occurred was a "no point data" error in a parallel I/O command occurring 101 days, 5 hours, 11 minutes and 12 seconds after the DRCX controller has started. The next most recent error was a "soft limit over" error during TPB or RS-232C operation occurring 96 days, 18 hours, 10 minutes and 2 seconds after the DRCX controller has started.

11-5-3 Utilities

(1-1) @INIT PGM

Initializes all program data.

Transmission example : @INIT PGM c/r l/f

Response example : OK c/r l/f

(1-2) @INIT PNT

Initializes all point data.

Transmission example : @INIT PNT c/r l/f

Response example : OK c/r l/f

(1-3) @INIT PRM <dual-axis robot type>

Initializes the parameter data to match the specified dual-axis robot.
For robot numbers, refer to "15-1-2 Robot number list".

Transmission example : @INIT PRM 110 c/r l/f .... Parameter data is initialized to match the model SXYx robot.

Response example : OK c/r l/f

⚠ *CAUTION*

After initialization, change the lead length parameters (X-axis: PRM60, Y-axis: PRM100) to match the robot lead length.

⚠ *CAUTION*

When a Cartesian robot (XY robot) such as SXYx and MXYx is specified, always enter the Y-axis stroke (in mm) after initialization. (See below.)

Transmit/receive example: (When Y-axis stroke is 250 mm)

Transmit	Receive
@WRITE PRMc/r l/f	
	READY c/r l/f
PRM58=250c/r l/f	
^Z (=1AH)	
	OK c/r l/f

(1-4) @INIT CPRM

Initializes the common parameters to the default value.

Transmission example : @INIT CPRM c/r l/f

Response example : OK c/r l/f

(1-5) @INIT XPRM <single-axis robot number>

Initializes the X-axis parameter data to match the specified single-axis robot.
For robot numbers, refer to "15-1-2 Robot number list".

Transmission example : @INIT XPRM 20 c/r l/f .... Parameters are initialized to match the F14 robot.

Response example : OK c/r l/f

CAUTION
After initialization, change the lead length parameter (PRM60) to match the robot lead length.

(1-6) @INIT YPRM <single-axis robot number>

Initializes the Y-axis parameter data to match the specified single-axis robot.
For robot numbers, refer to "15-1-2 Robot number list".

Transmission example : @INIT YPRM 21 c/r l/f .... Parameters are initialized to match the F14 robot (vertical use).

Response example : OK c/r l/f

CAUTION
After initialization, change the lead length parameter (PRM100) to match the robot lead length.

(1-7) @INIT CLOCK

Initializes the timer to 0, which is used to measure the total operation time of the DRCX controller. The alarm history and error history are also initialized at this point.

Transmission example : @INIT CLOCK c/r l/f

Response example : OK c/r l/f

(1-8) @INIT ALM

Initializes the alarm history.

Transmission example : @INIT ALM c/r l/f

Response example : OK c/r l/f

(1-9) @INIT ERR

Initializes the error history.
This command is available with controller version 18.50 or later.

Transmission example : @INIT ERR c/r l/f

Response example : OK c/r l/f

(2) @SWI <program number>

This command switches the execution program number. When a program is reset, program execution will always return to the first step of the program selected here. The program is reset when the @SWI command is executed.

Program number : This is a number used to identify each program and can be from 0 to 99.

Transmission example : @SWI 31 c/r l/f

Response example : OK c/r l/f

(3) @SWITSK <task number>

Switches the task number to be executed. In the subsequent step run, the program of the task selected here is executed. When the command such as @?NO or @?SNO is issued, the contents of this task replies to it.

Task number : This is a number used to identify each task and can be from 0 to 3.

Transmission example : @SWITSK 1 c/r l/f

Response example 1 : OK c/r l/f

Response example 2 : NG c/r l/f .... The specified task was not found.
72: not execute task c/r l/f

(4) @SINS <program number>,<step number>

Inserts data in a specified step of a specified program. All data below the inserted data will shift down one line. If the step following the last step is specified, a new step will be added. If the first step of a program that does not exist is specified, a new program will be created. The DRCX controller will transmit READY when this command is received. Confirm that READY is received and then transmit the insertion data.

Program number : This is a number used to identify each program and can be from 0 to 99.

Step number : This is a number used to identify each step and can be from 1 to 255.

Transmission example 1 :

Send	Receive
@SINS 19,4 c/r l/f	
	READY c/r l/f
TIMR 50 c/r l/f	
	OK c/r l/f

Transmission example 2 :

Send	Receive
@SINS 19,4 c/r l/f	
	NG c/r l/f 43: cannot find PGM c/r l/f

(5) @SDEL <program number>,<step number>

Deletes a specified step.

Program number	: This is a number used to identify each program and can be from 0 to 99.	
Step number	: This is a number used to identify each step and can be from 1 to 255.	
Transmission example	: @SDEL 31,99 c/r l/f	Deletes step 99 of program No. 31.
Response example 1	: OK c/r l/f	
Response example 2	: NG c/r l/f .... 42: cannot find step c/r l/f	The specified step number is not registered.

(6) @SMOD <program number>,<step number>

Modifies data in a specified step. The DRCX controller will transmit READY when this command is received. Confirm that READY is received and then transmit the modification data.

Program number	: This is a number used to identify each program and can be from 0 to 99.	
Step number	: This is a number used to identify each step and can be from 1 to 255.	
Transmission example 1	: Send	Receive
	@SMOD 0,5 c/r l/f	
		READY c/r l/f
	TIMR 50 c/r l/f	
		OK c/r l/f
Transmission example 2	: Send	Receive
	@SMOD 0,5 c/r l/f	
		NG c/r l/f 43: cannot find PGM c/r l/f

(7) @COPY <program number (copy source)>,<program number (copy destination)>

Copies a program. If a program exists in the copy destination, the program will be rewritten.

Program number	: This is a number used to identify each program and can be from 0 to 99.	
Transmission example	: @COPY 0,1 c/r l/f	Copies program No. 0 to program No. 1.
Response example 1	: OK c/r l/f	
Response example 2	: NG c/r l/f .... 43: cannot find PGM c/r l/f	The program to be copied is not registered.

(8) @DEL <program number>

Deletes a program.

Program number	: This is a number used to identify each program and can be from 0 to 99.	
Transmission example	: @DEL 10 c/r l/f	Deletes program No. 10.
Response example 1	: OK c/r l/f	
Response example 2	: NG c/r l/f .... 43: cannot find PGM c/r l/f	The program to be deleted is not registered.

(9) @PDEL <point number>,<number of points>

Deletes point data. Deletes the specified number of points starting with the point number specified here.

Point number	: This is a number assigned to each point and can be from 0 to 999.	
Number of points	: Any number between 1 and 999 can be specified.	
Transmission example	: @PDEL 16,10 c/r l/f	Deletes 10 points starting from point 16 (up to point 25).
Response example	: OK c/r l/f	

MEMO

Chapter 12 MESSAGE TABLES

This section lists all of the messages that are displayed on the TPB or sent to the PC (personal computer) to inform the operator of an error in operation or a current status. For a list of the alarm messages displayed if any trouble occurs, refer to "13-2 Alarm and Countermeasures".

12-1 Error Messages

12-1-1 Error message specifications

The error message transmission format is as follows.

<Error No.> : <Error message> c/r l/f

The length of the <error message> character string is 17 characters. (Spaces are added until the message contains 17 characters.) Thus, the character string length containing the c/r l/f will be 22 characters.

12-1-2 Command error message

Error No. 20	Message	no start code
	Cause	The start code (@) has not been added at the beginning of the command.
	Action	Always make sure the command begins with a start code (@).
Error No. 21	Message	illegal type
	Cause	The command is erroneous.
	Action	Use the correct command.
Error No. 22	Message	line buf overflow
	Cause	The number of characters in one line exceeds 80.
	Action	Limit the number of characters per line to 80 or less.
Error No. 23	Message	data error
	Cause	There is an error in numeric data.
	Action	Correct the data.
Error No. 24	Message	cannot access
	Cause	Execution is limited by the password or access level (operation level).
	Action	Cancel the limit.

12-1-3 Operation error message

Error No. 30	Message	soft limit over
	Cause	Executing the command will move the robot to a position that exceeds the soft limit set by parameter.
	Action	Review the point data or soft limit parameter.
Error No. 31	Message	running
	Cause	Another command is already being executed, so the command cannot be accepted.
	Action	Wait until execution of the current command finishes before inputting another command.
Error No. 32	Message	origin incomplete
	Cause	The command cannot be executed because a return to origin has not yet been completed.
	Action	Complete a return to origin first. Charge the absolute battery if not charged.
Error No. 33	Message	emergency stop
	Cause	The command cannot be executed because an emergency stop is triggered.
	Action	Cancel the emergency stop.
Error No. 34	Message	servo off
	Cause	The command cannot be executed because the servo is off.
	Action	Turn servo on.
Error No. 35	Message	system error 2
	Cause	An error interruption occurred due to noise or an unknown cause, so the status changed to servo off.
	Action	Turn servo on.
Error No. 36	Message	cannot restart
	Cause	Restart of the interpolation operation program was attempted.
	Action	Reset the program.
Error No. 37	Message	SVCE-port changed
	Cause	Execution was forcibly terminated because the SERVICE mode input state was changed.
	Action	Restart execution.
Error No. 38	Message	net link error
	Cause	The connection was forcibly disconnected because an error occurred in the network connection.
	Action	Remedy the network connection error, and then restart.

12-1-4 Program error message

Error No. 40	Message	stack overflow
	Cause	① Seven or more successive CALL statements were used within a CALL statement. ② In the program called as a subroutine by a CALL statement, a jump was made to another program by a JMP or JMPF statement.
	Action	① Reduce the number of CALL statements used in a CALL statement to 6 or less. ② Review the program.
Error No. 41	Message	cannot find label
	Cause	The specified label cannot be found.
	Action	Create the required label.
Error No. 42	Message	cannot find step
	Cause	The specified step cannot be found.
	Action	Check whether the step number is correct.
Error No. 43	Message	cannot find PGM
	Cause	The specified program cannot be found.
	Action	Check whether the program number is correct.
Error No. 44	Message	PGM memory full
	Cause	The total number of steps in all programs has exceeded 3000.
	Action	Delete unnecessary programs or steps.
Error No. 45	Message	step over
	Cause	The total number of steps in one program has exceeded 255.
	Action	Delete unnecessary steps or divide the program into two parts.
Error No. 46	Message	bad radius
	Cause	A radius exceeding the allowable range is specified with the MOVC statement.
	Action	Check whether the point data is correct.

12-1-5 System error message

Error No. 50	Message	system error
	Cause	An unexpected error occurred.
	Action	Contact YAMAHA and describe the problem.
Error No. 51	Message	illegal opecode
	Cause	There is an error in a registered program.
	Action	Check the program.
Error No. 52	Message	no point data
	Cause	No data has been registered for the specified point number.
	Action	Register the point data.
Error No. 53	Message	PRM0 data error
	Cause	The number of conditional input points is set to something other than 1 to 8.
	Action	Correct the setting for the PRM0 parameter.
Error No. 54	Message	PRM8 data error
	Cause	This error will not occur in the DRCX controller.
	Action	
Error No. 59	Message	robot type error
	Cause	Unsuitable parameter data was transmitted to the controller.
	Action	Initialize the parameters. Transmit the correct parameter data.

12-1-6 Multi-task error message

Error No. 70	Message	task running
	Cause	An attempt was made to start the task which is already in progress.
	Action	Check the program.
Error No. 71	Message	can't select task
	Cause	An attempt was made by a task to finish itself. An attempt was made to switch a task which is suspended.
	Action	Check the program. Check the task status.
Error No. 72	Message	not execute task
	Cause	An attempt was made to switch a task which has not started.
	Action	Check the task status.

12-2 TPB Error Messages

	Message	SIO error
	Cause	1. Parity error in data received from controller. 2. TPB was connected when dedicated command input was on.
	Action	1. Contact YAMAHA for consultation. 2. Turn all dedicated command inputs off before connecting the TPB.
	Message	bad format
	Cause	The memory card is not formatted.
	Action	Format the memory card.
	Message	save error
	Cause	Error in writing to the memory card.
	Action	Replace the memory card.
	Message	load error
	Cause	The memory card data is damaged.
	Action	Format or replace the memory card.
	Message	checksum error
	Cause	The memory card data is damaged.
	Action	Format or replace the memory card.
	Message	battery error
	Cause	The memory card battery voltage dropped.
	Action	Replace the memory card battery.
	Message	printer busy!!
	Cause	The printer is not ready.
	Remedy	Set the printer to print-ready state.

12-3 Stop Messages

12-3-1 Message specifications

The stop message transmission format is as follows.

<Message No.> : <Stop message> c/r l/f

The length of the <stop message> character string is 17 characters. (Spaces are added until the message contains 17 characters.) Thus, the character string length containing the c/r l/f will be 22 characters.

12-3-2 Stop messages

No. 60	Message	program end
	Meaning	Execution has stopped because the program has ended.
No. 61	Message	stop key
	Meaning	Execution has stopped because the Stop key on the TPB was pressed.
No. 62	Message	interlock
	Meaning	Execution has stopped because an I/O interlock was applied.
No. 63	Message	stop command
	Meaning	Execution has stopped because the STOP command was carried out.
No. 64	Message	key release
	Meaning	Execution has stopped by the hold-to-run function.

12-4 Displaying the Error History

A history of past errors can be displayed. Up to 100 errors can be stored in the controller.
This function is available when the controller version is 18.50 or later and the TPB version is 12.18 or later.

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu

1EDIT2OPRT3SYS 4MON
```

2) Next, press F4 (next) to change the menu display and then press F3 (UTL).

```
[SYS]
select menu

1SAFE2OPT 3UTL 4next
```

3) Press F2 (REC).

```
[SYS-UTL]
select menu

1HDPR2REC
```

4) Press F2 (ERR).

```
[SYS-UTL-REC]
select menu

1ALM 2ERR
```

5) History numbers, time that errors occurred (total elapsed time from controller start-up) and error descriptions are displayed. One screen displays the past 4 errors in the order from the most recent error.
Pressing the [X/Z−] and [X/Z+] keys displays the hidden items.
Press the [STEP UP] and [STEP DOWN] keys to sequentially scroll through the error list.

```
00:00101,05:11:12,CM
01:00096,18:10:02,PI
02:00080,10:07:33,CM
03:00015,20:35:45,CM
```

00 : 00101,05:11:12, CMU, 62:Interlock
↓ ↓ ↓ ↓
① ② ③ ④

① History number

② Time the error occurred

(The above example means that the error occurred 101 days, 5 hours, 11 minutes and 12 seconds after controller start-up.)

③ Movement command control mode immediately before the error occurred.

CMU: TPB or RS-232C control

PIO: Parallel I/O control

SIO: Serial I/O control

WIO: Remote command (CC-Link) control

④ Error description

(See "12-1 Error Messages" and "12-3 Stop Messages".)

6) To return to the previous screen, press (ESC).

```
[SYS-UTL-REC]
select menu

1ALM 2ERR
```

12

MESSAGE TABLES

MEMO

Chapter 13 TROUBLESHOOTING

This chapter explains how to take corrective action when a problem or breakdown occurs, by categorizing it into one of two cases depending on whether or not an alarm is output from the controller.

13-1 If A Trouble Occurs

If trouble or breakdown occurs, contact YAMAHA or your YAMAHA dealer, providing us with the following information in as much detail as possible.

Item	Description (example)
What you were using	・Controller model name : DRCX + Regenerative unit ・Robot model name : Multi (F14-20-350-F10-20-450) Dual-axis (SXYx-A-5545) ・Controller version : Ver. 18.11 ・Power : AC 100V ・I/O 24V power supply : Internal power supply and external power supply were used.
When	・When purchased ・How long used, how often used ・Problem happened at power on? One hour after power was turned on.
Under what conditions	・During automatic operation ・While writing a program ・When the robot was at a specific position
What happened	・Servo does not lock. ・Alarm (No. and message) is issued. ・Motor makes an unusual sound. ・A program was lost.
How often	・Always occurs. ・Occurs once an hour. ・Cannot be made to occur again.

13-2 Alarm and Countermeasures

If the READY signal is turned off except in cases of emergency stop, then an alarm has probably been issued. The status LED on the front panel of the controller lights up in red.

13-2-1 Alarm specifications

■ If an alarm is issued:

If an alarm is issued, keep the power turned on and connect the TPB or set the POPCOM on-line to check the contents of the alarm. An alarm message appears on the screen.

The transmission format for alarm messages is as follows.

```
<alarm generating axis> <alarm number> : <alarm message> c/r l/f
```

If an alarm is issued on the X-axis, X is transmitted as <alarm generating axis>. If an alarm is issued on the Y-axis, Y is transmitted.
The <alarm number> is displayed in two digits, so a one-digit number is prefixed with 0 like 01.
The <alarm message> is displayed in a 17 character string length. (Spaces are added until the message contains 17 characters.) Therefore, an message including c/r and l/f consists of 23 characters.

■ To cancel the alarm:

To cancel the alarm, turn the power off and after first eliminating the problem, turn it back on again.
If an alarm is still issued while the power is turned on, then try turning the power on while the robot is in emergency stop. No alarm detection is performed with this method, so that the data can be checked, corrected or initialized. Normal alarm detection is performed when the servo is turned on after cancelling emergency stop.

13-2-2 Alarm message list

Alarm No.	Alarm Message	Meaning	Possible Cause	Action
01	OVER LOAD	Excessive load on motor	① Improper operation ② Motor failure ③ Parameter error ④ Electromagnetic brake failure or wire broken ⑤ Wrong power supply voltage setting ⑥ Insufficient power supply capacity ⑦ Excessive friction in robot	① Lower the operation duty on the robot or reduce the acceleration parameter, or correct the payload parameter. ② If the motor armature resistance is too low or the motor movement is sluggish when turned by hand, then replace the motor assembly. ③ Initialize the parameters and check the robot type setting. ④ Supply 24V to the brake wire to check whether the brake is released. ⑤ Check the voltage setting (100V/200V). ⑥ Check the power supply capacity. If too low, use a power supply of larger capacity. ⑦ Check whether the robot moving parts work sluggishly. If sluggish, then adjust the mechanical alignment.
02	OVER CURRENT	Motor drawing too much current	① Motor wire shorted ② Motor failure ③ Controller failure ④ Parameter error	① Check the motor wires for electrical continuity, and replace the motor assembly if abnormality is found. ② Replace the motor if internally shorted. ③ If the resistance between motor terminals U and V or V and W is less than 1 kilo-ohms, the output transistor is defective, so replace the DRCX controller. *The resistance between U and W is about 24Ω, but this is a normal value. ④ Initialize the parameters and check the robot type setting.
03	OVER HEAT	Transistor has heated to 90°C or more.	① Rise in ambient temperature (above 40°C) ② Excessive load on motor ③ Defective transistor	① Correct the ambient environmental conditions. (Install a cooling fan.) ② Lower the operational duty on the robot. ③ If the controller is being used correctly, the transistor is probably defective, so replace the DRCX controller.
04	POWER DOWN	Power supply voltage has dropped to less than 85% of rated value.	① Insufficient power supply capacity ② Wrong power supply voltage being used	① Check the power supply capacity. If insufficient, use a power supply having larger capacity. (Power is consumed mostly during return-to-origin of stroke end detection, robot start-up and acceleration/deceleration.) ② Check the voltage specifications indicated on the front panel of the controller.
05	BATT.LOW-VOLTAGE	System backup battery voltage is low.	① Battery worn out. ② Battery failure	①② Replace the battery. (If not possible to replace the battery immediately, then temporarily set bit 3 of PRM20 to "1").
06	24V POWER OFF	24V power is not supplied.	① 24V power supply is not connected to A13 and B13 of the I/O connector. ② Fuse has blown due to short-circuit or excessive current flow in the 24V circuit.	① Check the 24V power supply. ② Check for short-circuit using a multi-meter or recheck the I/O connections.

Alarm No.	Alarm Message	Meaning	Possible Cause	Action
07	P.E. COUNTER OVER	Overflow in position deviation counter	① Mechanical lock ② Motor wire is broken or connected wrong. ③ Electromagnetic brake failure or wire broken ④ Parameter error	① Check whether robot moving parts are locked. ② Check the motor wire and resolver signal wire connections. ③ Supply 24V to the brake wire to check whether the brake is released. ④ Initialize the parameters.
08	PNT DATA DESTROY	Point data has been corrupted.	① Backup circuit failure ② Power was turned off while writing data. ③ Data was destroyed by external noise.	①② In emergency stop, turn power on and check point data. If part of the data is defective, correct the data. If all data are defective, initialize the point data and then reload the data. If there is no problem with the data, perform rewriting on any data. ③ Check the surrounding environment for noise.
09	PRM DATA DESTROY	Parameter data has been corrupted.	① Backup circuit failure ② Power was turned off while writing data. ③ Data was destroyed by external noise.	①② In emergency stop, turn power on and initialize the parameters. ③ Check the surrounding environment for noise.
10	PGM DATA DESTROY	Program data has been corrupted.	① Backup circuit failure ② Power was turned off while writing data. ③ Data was destroyed by external noise.	①② In emergency stop, turn power on and check program data. If part of the program is defective, correct the data. If all data are defective, initialize the program data and then reload the data. If there is no problem with the data, perform rewriting on any data. ③ Check the surrounding environment for noise.
11	SYSTEM FAULT	Software problem	① External noise or other factors have disrupted software program. ② Overflow in receiving buffer . When communicating with a PC, the XON/ XOFF control communication parameter was not selected on the PC.	① Check the surrounding environment for noise. ② Select the XON/XOFF control.
12	BAD ORG-SENSOR	Origin sensor failure	① Origin sensor wire breakage ② Origin sensor failure ③ Poor origin dog adjustment	①② Replace the origin sensor ass'y. ③ Readjust.
13	Not used			
14	FEEDBACK ERROR 1	Incorrect parameter setting	① Parameter error ② Motor is miswired. ③ Resolver signal wire is misconnected.	① Initialize the parameters. ② Check the motor wire connection. ③ Check the resolver signal wire connection.
15	FEEDBACK ERROR 2	Resolver signal discontinuity	① Resolver signal wire is broken.	① Check the resolver signal wire connection.

Alarm No.	Alarm Message	Meaning	Possible Cause	Action
16	ABNORMAL VOLTAGE	Excessive voltage (higher than 420V) generated	① Rise in regenerative absorption resistor temperature above 120°C). ② No regenerative unit is connected. ③ Defective regenerative unit cables ④ Incorrect power supply voltage ⑤ Regenerative unit defect	① Lower the operation duty on the robot, or install a cooling fan. ② Connect the regenerative unit. ③ Check the cables with a multimeter. ④ Check the voltage setting (100V/200V). ⑤ If there is no problem with the usage, the regenerative unit is probably defective. Replace it.
17	SYSTEM FAULT 2	Controller's internal LSI error	① Internal LSI failure or malfunction	① If the error occurs frequently, then the LSI is probably defective, so replace the DRCX controller.
18	FEEDBACK ERROR 3	Motor cable is disconnected, improperly wired or overload	① Motor wire is broken or misconnected. ② The robot slider struck on an obstacle or mechanical damper. ③ Defective or disconnected electromagnetic brake ④ Parameter error ⑤ Wrong power supply voltage setting ⑥ Insufficient power supply capacity ⑦ Drop in voltage at stopper origin	① Check the motor wire connection. ② Remove the obstacle or correct the point data or origin position. ③ Apply 24V to the brake line and check brake release. ④ Initialize the parameters. ⑤ Check the voltage setting (100V/200V). ⑥⑦ Check the power supply capacity and increase if necessary.
19	SYSTEM FAULT 3	CPU error	① External noise or other factors have disrupted software program. ② CPU failure or malfunction	① Check the environment for noise. ② If the error occurs frequently, then the CPU is defective. Replace the DRCX controller.
20	Not used			
21	Not used			
22	VERSION MISMATCH	Wrong combination of PB and controller	① The PB used does not match the controller.	① Replace the PB.
23	ABS.BAT.L-VOLTAGE	The absolute battery voltage is low.	① The absolute battery voltage is low. ② The absolute battery is not connected. ③ End of the absolute battery service life. ④ The absolute battery is defective.	① Charge the absolute battery. ② Connect the absolute battery. (If using the controller with the absolute battery disconnected, set bit 4 of PRM 20 to "0".) ③ When the backup time becomes short even after fully charging the battery, this is probably the end of service life. Replace the battery. ④ When this alarm does not disappear even after fully charging the battery, the battery is probably defective. Replace the battery.
24	ABS.DATA ERROR	Absolute data error was detected.	① Movement amount has exceeded the limit (approx. ±4000 turns) that can be retained during power off. ② The absolute battery was disconnected during power off. ③ Discontinuity of absolute battery wire ④ Discontinuity or misconnection of resolver signal line ⑤ Circuit defect or malfunction	① Limit the movement range during power off. ② Do not disconnect the absolute battery when the position data is backed up. ③ Check the absolute line connection. ④ Check the resolver signal line connection. ⑤ If the fault occurs frequently, replace the controller.
25	ABS.DATA ERROR 2	Abnormal reset	① Malfunction due external noise, etc. ② Circuit defect or malfunction	① Check the environment for noise. ② If the fault occurs frequently, replace the controller.
26	FEEDBACK ERROR 4	Motor wire breakage or misconnection	① Motor wire is broken or connected wrong. ② Parameter error ③ Wrong power supply voltage setting ④ Insufficient power supply capacity	① Check the motor wire connection. ② Initialize the parameters. ③ Check the voltage setting (100V/200V). ④ Check the power supply capacity. If too low, use a power supply of larger capacity.

13-3 Troubleshooting for Specific Symptom

If any problems develop while the controller is being used, check the items below for the appropriate way to handle them. If the problem cannot be corrected using the steps listed below, please contact our sales office or sales representative right away.

13-3-1 Relating to the robot movement

No.	Symptom	Possible Cause	Items to Check	Action
1	Servo of robot does not lock even after power is turned on.	1) Power is not being supplied.	• Check that the status LED on the front panel of the controller lights up or flashes.	• Check the voltage on the power input terminal block. If the voltage is correct, replace the DRCX controller.
		2) Emergency stop is activated.	• If the READY signal of the I/O connector is off and no alarm has been issued, an emergency stop is in effect. • Check whether the status LED is flashing.	• Check whether the emergency stop button of the TPB or the I/O emergency stop input (between EMG1 and EMG2) is on.
		3) The servo is off.	• Check whether the servo has been turned off in the program, and whether the TPB has been plugged or unplugged without pressing the ESC switch. • Check whether the status LED is flashing.	• Turn the servo on with the I/O servo recovery input or from the TPB operation.
		4) An alarm has occurred.	• Connect the TPB and check whether an alarm is displayed. • Check whether the status LED is lit in red.	• Take corrective action according to the instructions in the alarm message list.
2	Program does not run correctly.	1) Misprogramming	• Run step operations to check whether the program is correct.	• Correct the program if necessary.
		2) A different program is selected.	• Reset the program and check whether the desired program is selected.	• Change the program to select the desired program.
		3) The selected program No. was switched when the program was loaded into the controller from the memory card.	• Reset the program and check whether the desired program is selected.	• Change the program to select the desired program.
3	Origin incomplete error occurs even if using a robot with absolute specifications. Return-to-origin is required to start robot operation.	1) A parameter relating to the origin was changed.	• Check whether a parameter has been changed from the TPB or POPCOM or whether a parameter has been loaded into the controller from the memory card or whether parameters have been initialized.	• If an origin-related parameter has been changed, perform return-to-origin again.
		2) Feedback error 2 or an alarm relating to absolute control occurred.	• Check the cable for electrical discontinuity or check whether the absolute battery is sufficiently charged.	• Remove the cause of alarm and then perform return-to-origin.
		3) Absolute backup function is disabled by parameter setting.	• Check whether bit 4 of PRM20 is set to "1".	• Set bit 4 of PRM20 to "1".

No.	Symptom	Possible Cause	Items to Check	Action
4	Abnormal noise or vibration occurs.	1) Coupling is not securely tightened.	• Check the coupling bolts.	• Tighten if loose.
		2) A screw on the cover is loose.	• Check the screws used to secure the cover.	• Tighten if loose.
		3) Robot installation surface is not flat or even.	• Measure the degree of leveling.	• Correct the leveling if outside the tolerance limit.
		4) Linear guide abnormality	• Check for debris intrusion, damage or deformation.	• Clean or replace the linear guide. • Check to make sure the linear guide is being used properly.
		5) Ball screw abnormality	• Check for debris intrusion, damage or deformation.	• Clean or replace the ball screw. • Check to make sure the ball screw is being used properly.
		6) Bearing abnormality	• Check for noise or vibration around the axes.	• Correct the assembled condition.
		7) Motor failure	• Try replacing the motor with another one.	• If another motor works normally, then the currently used motor is defective so replace it.
		8) Improper grounding of motor case	• Measure to see if the resistance between the motor case and the controller's FG terminal is 1 ohm or less.	• If the resistance is too high, find and repair the poor connection.
		9) Parameter setting error	• Check the parameter data.	• Initialize the parameters.
		10) Controller failure	• Try using another controller if available.	• If another controller operates normally, then the currently used controller is defective, so replace it. • Use the correct controller and robot combination.
5	Position deviation or offset occurs. If this occurs, leave the power on and perform return-to-origin. Depending on the results of the return-to-origin, there are two possible causes of the problem: • If position offset is not corrected by the return-to-origin: Mechanical offset - See causes 1) to 4). • If position offset is corrected by the return-to-origin: Electrical offset - See causes 5) to 8).	1) Coupling or pulley is not securely tightened.	• Check the coupling or pulley bolts.	• Tighten if loose.
		2) Ball screw is loose.	• Check the ball screw.	• Replace the ball screw if necessary.
		3) Belt is not properly meshed.	• Check the acceleration. • Check the amount of belt slack.	• Correct the parameter setting. • Adjust the belt tension.
		4) Robot is not securely installed.	• Make sure there is no loose parts where the robot is installed.	• Reinstall the robot securely.
		5) Malfunction caused by noise	• Check whether the motor case is properly grounded. • Check that the resistance between the motor case and the controller's FG terminal is 1 ohm or less, and also that the controller is properly grounded.	• If the controller is used near a unit that generates noise such as welding machines and electric discharge machines, move it as far away as possible. If the entire unit cannot be moved, then at least move the power supply away. It might be necessary to use a noise filter or isolating transformer depending on the trouble.
		6) The robot was moved at high speeds during power off. (higher than 3000rpm)		• Do not move the robot at high speeds while the position data is retained.
		7) Controller failure	• Try using another controller if available.	• If another controller operates normally, then the currently used controller is defective, so replace it.
		8) Motor failure	• Try replacing the motor with another one.	• If another motor works normally, then the currently used motor is defective so replace it.
6	During return-to-origin, the robot stops due to alarm after striking on the stroke end (overload).	1) Wrong robot type number setting	• Connect the TPB and check the robot type number.	
		2) Origin sensor failure	• Remove the cover and check the sensor operation by looking at the LED. Origin position: LED is ON. Other positions: LED is OFF. • Check whether the origin sensor signal turns ON/OFF in DIO monitor of TPB.	• If the origin sensor does not work correctly, replace it.
		3) Parameter setting error	• Check the parameter for the origin detection method. (X-axis: PRM55) (Y-axis: PRM95)	• When the parameter setting is "1" (stroke-end detection method), "3" (mark method), initialize the parameter. • When the parameter setting is "0" (sensor method), set the parameter to "1".
		4) The origin position is inappropriate so the robot slider makes contact with the damper when at the origin.	• Use the TPB to check whether the alarm occurs before or after return-to-origin is complete. If the alarm occurs after return-to-origin is complete, the damper position is inappropriate.	• Adjust the origin position.

No.	Symptom	Possible Cause	Items to Check	Action
7	Robot starts moving at high speed when the power is turned on. * The DRCX controller has a safety circuit to detect wire breakage, but check the points listed on the right anyway.	1) Motor and/or resolver are miswired.	• Check the motor wire and resolver signal wire connections.	• Correct the connections.
		2) Parameter error		• Try Initialize the parameters.
8	Robot speed is abnormally fast or slow.	1) Parameter setting error	• Check whether the robot setting displayed on the TPB matches the robot actually used.	• If they do not match, initialize the parameters.
		2) Speed setting was changed.	• Check the speed parameter (PRM17).	• Correct the parameter.

13-3-2 Relating to the I/O

No.	Symptom	Possible Cause	Items to Check	Action
1	Output signal cannot be controlled. * In cases other than cause 2, the output signal cannot be controlled even with the manual instruction of TPB general-purpose output.	1) Wiring to external devices is incorrect.	• Check the wiring. • Check the operation with the manual instruction of the TPB general-purpose output. (Refer to "7-4 Manual Control of General-Purpose Output".)	• Make the correct wiring by referring to the connection diagram.
		2) Misprogramming	• Connect the TPB and check the program.	• Change the program.
		3) Output transistor is defective.	• Measure the voltage at the PLC input terminal. ON: 0.5V or less OFF: +IN COM (+24V)	• Replace the DRCX controller if the output transistor is defective.
2	Robot will not move even with dedicated command input.	1) Return-to-origin has not yet been completed.	• Connect the TPB and check the operation.	• Reperform return-to-origin.
		2) Program cannot be run.	• Connect the TPB and check the operation.	• Eliminate the cause of error by referring to the error message.
		3) Signal pulse width is too narrow.	• Check that the signal pulse width is 50ms or more.	• Increase the signal pulse width ("on" duration).
		4) Time interval before inputting a dedicated command after canceling emergency stop is too short.	• After canceling emergency stop, allow at least 200ms before inputting a dedicated command before inputting a dedicated command.	• Increase the delay time.
		5) Interlock signal remains off.	• Check the signal by using the TPB DIO monitor.	• Switch on the interlock signal.
		6) Another dedicated command input is on.	• Check the signal input (by using a PLC monitor, etc.).	• Turn off the dedicated command input.

13-3-3 Other

No.	Symptom	Possible Cause	Items to Check	Action
1	An error occurs when the TPB is connected. The TPB cannot be used.	1) A dedicated I/O command input is on.	• Check the signal input (by using a PLC monitor, etc.)	• Always turn off dedicated command input signals when connecting the TPB to the controller.
		2) The TPB cable is broken.	• Check the cable wiring. • Try connecting another TPB if available.	• Replace the TPB if defective.
2	Program can be input only up to No. 31, or point data can be specified only up to P254.	1) An old version DPB was used as the teaching box.	• Check that the DPB version is 1.50 or later.	• Replace the ROM to upgrade the version.
		2) Communication cable specifications are wrong.	• Check whether the wrong cable (POPCOM cable, etc.) is being used.	• Use the specified communication cable. (POPCOM cable is different from the communication cable.) As an alternative, transmit "@DPBVER 210" in advance.
		3) POPCOM/WIN version is obsolete.	• Check that POPCOM/WIN version is 1.3 or later.	• Upgrade the POPCOM/WIN version.
		4) POPCOM/DOS is used.		• Use POPCOM/WIN.
3	Return-to-origin cannot be performed by the mark method.	1) The TPB version is obsolete.	• Check whether the TPB version is 2.10 or later.	• Replace the ROM.
		2) The DPB version is obsolete.	• Check whether the DPB version is 1.60 or later.	• Replace the ROM.
		3) The POPCOM/WIN version is obsolete.	• Check whether the POPCOM/WIN version is 1.8 or later.	• Upgrade the version.
		4) POPCOM/DOS is being used.		• Use POPCOM/WIN.

13-4 Displaying the Alarm History

A history of past alarms can be displayed. Up to 100 alarms can be stored in the controller. This function is available with TPB version 12.18 or later.

1) On the initial screen, press F3 (SYS).

```
[MENU]
select menu


1EDIT2OPRT3SYS 4MON
```

2) Next, press F4 (next) to change the menu display and then press F3 (UTL).

```
[SYS]
select menu


1SAFE2OPT 3UTL 4next
```

3) Press F2 (REC).

```
[SYS-UTL]
select menu


1HDPR2REC
```

4) Press F1 (ALM).

```
[SYS-UTL-REC]
select menu


1ALM 2ERR
```

5) History numbers, time that alarms occurred (total elapsed time from controller start-up) and alarm descriptions are displayed. One screen displays the past 4 alarms in the order from the most recent alarm.
Pressing the [X- Z-] and [X+ Z+] keys displays the hidden items.
Press the [STEP UP] and [STEP DOWN] keys to sequentially scroll through the alarm list.

```
00:00101,05:11:12,X0
01:00096,18:10:02,X0
02:00080,10:07:33,X0
03:00015,20:35:45,X0
```

00 : 00101,05:11:12, X04: POWER DOWN
↓ ↓ ↓
① ② ③

① History number
② Time the alarm occurred
(The above example means that the alarm occurred 101 days, 5 hours, 11 minutes and 12 seconds after controller start-up.)
③ Alarm description
(See "13-2-2 Alarm message list".)

6) To return to the previous screen, press (ESC).

```
[SYS-UTL-REC]
select menu

1ALM 2ERR
```

Chapter 14 MAINTENANCE AND WARRANTY

For safety purposes, always turn the power off before starting robot maintenance, cleaning or repairs, etc.

14-1 Warranty

For information on the product warranty, please contact your local agent where you purchased your product.

14-2 Replacing the System Backup Battery

If an alarm is issued indicating that the system backup battery voltage is low, replace the battery using the procedure listed below.

(1) First, make a backup copy of all necessary data using a memory card or POPCOM software, because that data in the controller might be lost or destroyed during battery replacement.

(2) Unplug all connectors from the controller and then remove the top cover.

(3) You can now see the control board. Remove it from the controller.

(4) The lithium battery is soldered to the control board. Use a desoldering tool or similar tool to remove the solder and then remove the battery from the control board.

(5) Solder the new battery to the control board.

Battery product number: CR2450THE (Toshiba Battery)

(6) Install the control board back in its original position.

(7) Reattach the top cover.

(8) Initialize all data and then return the data you backed up into the controller.

> Please note that the state of California USA has legal restrictions on the handling of manganese dioxide lithium batteries. See the following website for more information:
> *http://www.dtsc.ca.gov/hazardouswaste/perchlorate*

14-3 Replacing the Absolute Battery

The absolute battery will wear down and must be replaced. Replace the battery when its service life has expired or when problems with backing up data occur even when the battery charge time was long enough.
Though battery wear depends on the number of charges and the ambient temperature, both B1 type and B2 type batteries should generally be replaced one and a half years after being connected to the controller. Replace both batteries at the same time.
Always charge the new battery after it is installed. The battery is automatically charged while the controller is turned on. Keep the battery charged for longer than the time listed in the table below. Since the battery charging time does not affect robot operation, the controller can be used to perform teaching, program editing and robot operation while the battery is still being charged.

	Hours until full charge *1)	Backup time *2)
Type B1 (3.6V/700mAh)	15h	120h
Type B2 (3.6V/2000mAh)	48h	340h

*1) At ambient temperature of 20°C
*2) After power is off with absolute battery fully charged.

* When the absolute battery is disconnected from the controller, an alarm (24:ABS. DATA ERROR) is issued. So an alarm is always issued when the absolute battery is replaced, but this is not an error. (An alarm "23:ABS. BAT.L-VOLTAGE" may occur in some cases.)

> Absolute batteries are recycled items. Please contact our sales office for proper disposal of used batteries.

14-4 Updating the System

YAMAHA may request, on occasion, that you update the system in your equipment. The following steps describe how to update the system.
Before updating the system, you must set up a system that allows communications between the controller and a PC (personal computer). Use a communication cable which conforms to the specifications listed in "11-2 Communication Cable Specifications".

(1) First, make a backup copy of the necessary data using a memory card or POPCOM software, because the data in the controller might be lost or destroyed while updating the system.

(2) With the controller started up, type "@SETUP" and press the Return (Enter) key.

(3) When a response "OK" is returned from the controller, turn off the power to the controller.

(4) Unplug the I/O connector from the controller.

(5) With the I/O connector still unplugged, turn on the power to the controller again.

(6) The controller enters the system setup mode and the YAMAHA copyright message appears on the PC screen.

(7) Type "@UPDATA" and press the Return (Enter) key.

(8) The controller then returns READY message, so transfer the new system data. (It will take about 5 minutes to transfer all the data.)

(9) An "OK" response is returned when the system transfer is complete. Now turn off the power to the controller.

(10) Plug in the I/O connector.

(11) Turn on the power to the controller again. Type "@?VER" and press the Return (Enter) key. Then make sure that the controller version is updated.

(12) Initialize all data and then return the data you backed up into the controller.

CAUTION

- ***The controller must remain in emergency stop until updating of the system is finished. (Specifically, terminals A24 (EMG1) and B24 (EMG2) of the I/O connector should be left open.)***
- ***Before starting the system updating, we strongly recommend for safety reasons that the robot cable be disconnected from the controller.***

MEMO

SPECIFICATIONS

15-1 DRCX sereis

15-1-1 Basic specifications

Specification item	Model	DRCX
Basic specifi-cations	Applicable motor capacitance	Total power 1200W max. *1)
	Max. power consumption	1600VA
	External dimensions	W100 H250 D157mm
	Weight	2.1kg
	Power supply voltage	0505, 0510, 0520, 1005, 1010, 2005 drivers Single phase AC100 to 115V / 200 to 230V, ±10%, 50/60Hz 1020, 2010, 2020 drivers Single phase AC200 to 230V, ±10%, 50/60Hz
Axis control	No. of controllable axes	2 axes
	Control method	AC full digital servo PTP, CP*2), ARC*2)
	Position detection method	Resolver with multi-turn data backup function
	Speed setting	100-step setting possible per program step
	Acceleration setting	Automatically set according to robot type and payload. 100-step setting is also possible with acceleration parameter.
	Servo adjustment	Handled with parameters (special). Servo gain, current limit, etc.
	No. of pulses	16384P/R
	Lead length	Lead length is selectable during initialization or by parameter setting (special function)
Memory	ROM	256K bytes (with built-in CPU)
	RAM	128K bytes with 64K lithium battery backup (5-year life)
	No. of program steps	3000 steps or less in total, 255 steps/program
	No. of programs	100
	No. of points	1000 points
	No. of multi tasks	4
	Teaching method	MDI (coordinate value input), teaching playback, direct teaching
	Auxiliary memory unit	IC memory card is available as TPB option
I/O	I/O input	General-purpose: 16 points, dedicated input: 8 points
	I/O output	General-purpose: 13 points, dedicated output: 3 points, open collector output (0.5A/24V maximum per output)
	External drive power supply	DC24V/900mA (when not using brake)
	Brake output	Relay output (for 24V/300mA brake), two channel, built-in power supply (24V)
	Origin sensor input	Connectable to a DC24V B-contact sensor
	Emergency stop input	Normally closed contact input (origin return not required after emergency stop is released)
	Serial interface	One RS-232C channel (for communication with TPB or PC)
	Network (option)	CC-Link, DeviceNet, Ethernet, PROFIBUS
General specification	Ambient temperature	0 to 40°C
	Storage temperature	-10 to 65°C
	Ambient humidity	35 to 85%RH (no condensation)
	Noise immunity	Conforms to IEC61000-4-4 Level 2

*1) A regenerative unit (RGU-2) is required when operating robot models specified by YAMAHA or a robot handling a load with large inertia.
*2) Only for Cartesian robot.

⚠ ***CAUTION***
Specifications and external appearance are subject to change without prior notice.

15
SPECIFICATIONS

15-1-2 Robot number list

Each robot model has an identification number as listed in the table below. After you initialize the parameters, enter the correct robot number that matches the robot model actually connected to the controller.

Single-axis robot

	T6	T7	F10	F14	T9	B10	B14	B14H	F14H
Standard (horizontal installation model)	14	10	28	20	18	25	26	27 (827*1)	32
-BK (vertical installation model)	15	11	72	21	19				76

	T9H	F17	F17L	N15	N18	F20	F20N
Standard (horizontal installation model)	78	30 (830*2)	36	38	39	40 (840*2)	42 (842*2)
-BK (vertical installation model)	79	31	37			41	

	C6	C14	C14H	C17	C20
Standard (horizontal installation model)	14	20	32	30	40
-BK (vertical installation model)	15	21	76	31	41

FROP	
R5	16
R10	17
R20	33

*1: When maximum rotational speed is 4500 rpm. (RGU-2 required)
*2: When maximum rotational speed is 3600 rpm. (Ball screw lead length 20mm; RGU-2 required)

Cartesian robot

Arm type	PXYx		TXYx		FXYx		FXYBx	
	No.	Display	No.	Display	No.	Display	No.	Display
-A	305	PXYx	300	TXYx	310	FXYx	320	FXYBx
-M								
-P								
-G								
-F								

Arm type	SXYx		SXYBx	
	No.	Display	No.*	Display
-A	110	SXYx	120 (125)	SXYBx
-M	112	SXYx-m		
-P	113	SXYx-p		
-G				
-F	116	XZx-ZF	121 (126)	XZBx-ZF
	117	XZx-ZS	122 (127)	XZBx-ZS
			128	XZBx-ZFH
	119	XZx-ZFL	129	XZBx-ZFL

*: When maximum rotational speed of X-axis is 4500 rpm, use the numbers in parentheses. (RGU-2 required)

Arm type	MXYx		HXYx		HXYLx	
	No.	Display	No.	Display	No.	Display
-A	130	MXYx	150	HXYx	160	HXYLx
-M	132	MXYx-m	152	HXYx-m	162	HXYLxm
-P	133	MXYx-p	153	HXYx-p	163	HXYLxp
-G	134	MXYx-g	154	HXYx-g	164	HXYLxg
-F			156	XZx-ZH		
			157	XZx-ZL		
	138	XZMx-ZFH				
	139	XZMx-ZFL				

P&P robot

	YP320x		YP220Bx	
No.(Display)	211	YP320x	213	YP220Bx

15 SPECIFICATIONS

15-1-3 LED display

The table below shows the specifications of the operation status LED on the front panel of the controller.

LED display	Robot or controller operation status
Not lit	The power is off or fuse is blown.
Lit in green	Servo motors of all axes are on. (All axes are ready for operation)
Lit in red	Error occurred. (Alarm is being issued.)
Flashes green (0.5 sec.) and red (0.5 sec.)	Emergency stop
Flashes green (1.5 sec.) and red (0.5 sec.)	Emergency stop is canceled. (One or more axes are servo-off.)

15-1-4 Absolute Battery Unit

The absolute battery unit has the following basic specifications. (Either B1 or B2 type is supplied depending on the request by the user.)

Specification item \ Type		B1	B2
Basic specifications	Battery type	Ni-Cd battery	
	Charge method	Trickle charge	
	Battery capacity	3.6V/700mAh	3.6V/2000mAh
	Dimensions	W47 × H52 × D15mm	L152 × ϕ29mm
	Weight	80g	280g
Characteristics	Data backup time *1)	120h	340h
	Hours until full charge *2)	15h	48h
	Service life	About 1.5 years	
General	Cable length	300mm (standard)	
	Accessories	Batteries (2 pieces), cable ties, fastening straps (4 pieces each)	

*1) After power is off with the absolute battery fully charged.
*2) At ambient temperature of 20°C

15

SPECIFICATIONS

15-2 TPB

15-2-1 Basic specifications

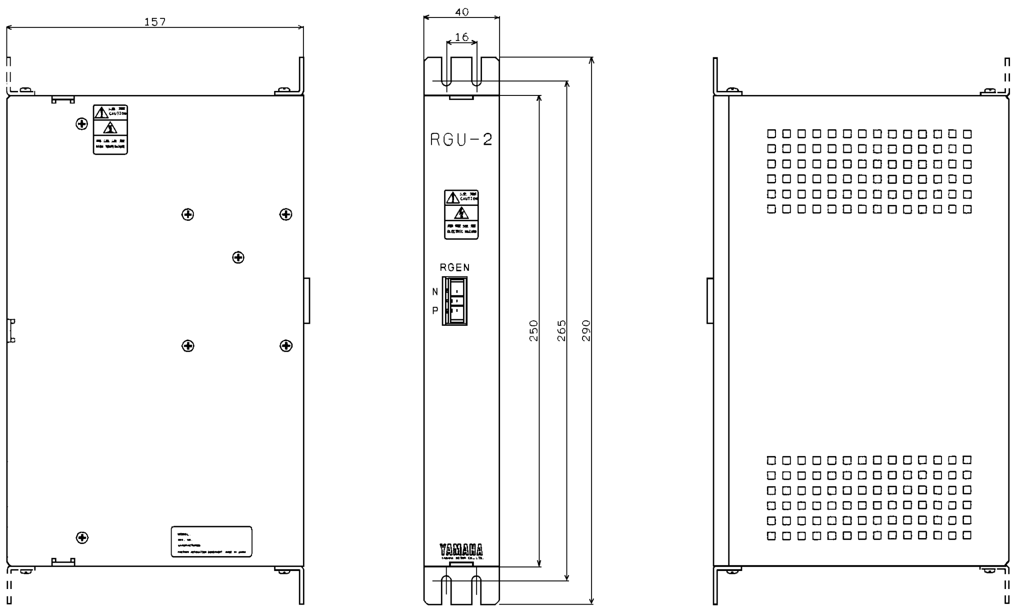
Specification item	Model	TPB
Basic specifications	External dimensions	W107 H235 D47mm
	Weight	590g
	Power consumption	5 V, 200 mA max.
	Power supply	DC12V (supplied form the controller)
	Cable length	Standard 3.5m
I/O	Serial interface	RS-232C, one channel, for communications with controller
	Display	Liquid crystal, 20 characters 4 lines
	Keyboard	29 keys, membrane switch + emergency stop button
	Emergency stop button	Normally-closed contact (with lock function)
	Auxiliary memory device	IC memory card
General specification	Ambient temperature	0℃ to 40℃
	Storage temperature	-10℃ to 65℃
	Ambient humidity	35 to 85% RH (no condensation)
	Noise immunity	Conforms to IEC61000-4-4 Level 2

15-3 Regenerative Unit (RGU-2)

15-3-1 Basic specifications

Specification item \ Model		RGU-2
Basic specifications	External dimensions	W40 × H250 × D157mm
	Weight	1.1kg
	Cable length	300mm
Special specifications	Regenerative voltage	Approx. 380V or more
	Regenerative stop voltage	Approx. 360V or less
General specifications	Ambient temperature	0℃ to 40℃
	Storage temperature	-10℃ to 65℃
	Ambient humidity	35 to 85% RH (no condensation)
	Noise immunity	Conforms to IEC61000-4-4 Level 2

15-3-2 Dimensions



Chapter 16 APPENDIX

16-1 Operation When Not Using Absolute Function

An absolute backup function is standard on the DRCX controllers. This means return-to-origin is unnecessary each time the controller is turned on.
In some cases, however, the absolute function is not needed because the customer is accustomed to always performing return-to-origin when the power is turned on. In other cases the absolute function may not be needed because the current sequence system can be used to automatically perform return-to-origin when the power is turned on.
In such kind of cases where the absolute function is not needed, use the following procedure to operate the controller with the absolute battery removed.

(1) Set bit 4 to 0 in PRM20 (system mode selection parameter) to disable the absolute backup function.

Bit 4 is set to 1 as the default value so that PRM20 = 16. If bits other than bit 4 can stay the default value without causing problems, just enter 0 in PRM20.

(See "PRM20: System mode selection")

(2) Turn off power to the controller and remove the absolute battery from the controller.

* Once the absolute backup function has been disabled, always be sure to perform return-to-origin when the controller power is turned on.

* Even if the absolute battery is removed, the various data such as the program, point data, and parameters stored inside the controller will not be lost.

(The various data such as the program, point data, and parameters are retained by a system backup battery inside the controller. The absolute battery is used to hold the robot position data when the controller power is off.)

16-2 How to Handle Options

16-2-1 Memory card

A memory card (option) can be used with the TPB to back up the DRCX controller data.

■ Using the memory card

1. Insert the memory card into the TPB as shown in Fig. 16-1.
2. Back up the data by referring to section "10-6 Using a Memory Card" in Chapter 10.

Fig. 16-1 Inserting the memory card

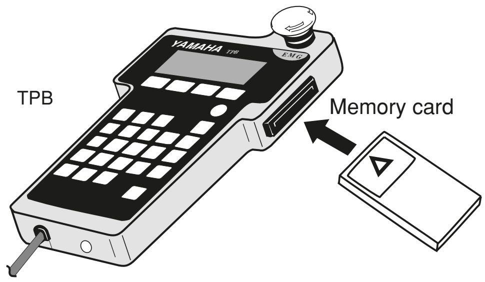
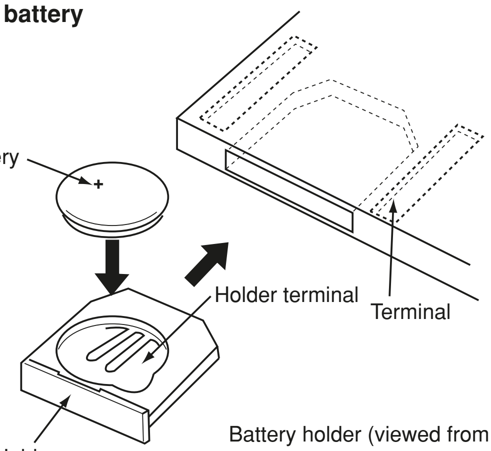


Fig. 16-2 Replacing the battery



■ Precautions when using the memory card

1. Insert the memory card all the way inwards until you feel it makes contact.
2. Be careful not to insert the memory card facing the wrong direction. The mark "Δ" should be facing upward. (A pin for preventing reverse insertion is provided.)
3. Insert or pull out the memory card only when the power is supplied to the TPB.
4. Never eject the memory card while backing up data.
5. The memory card should be used under the following environmental conditions:
   Ambient temperature range : -10 to 40°C
   Ambient humidity range : Below 85% RH
   Storage temperature range : -20 to 60°C
6. Do not leave the memory card inserted in the TPB when not in use, since this will shorten the battery life.
   The battery life is about 5 years (at ambient temperature of 25°C).
   If the battery voltage drops, an alert message appears on the TPB, so replace the battery by referring to Fig. 16-2.
   Battery product number : Panasonic BR2325 or CR2325 (64KB card) or equivalent type

16
APPENDIX

■ Data size that can be saved

Data size that can be saved on one memory card is as follows:

Memory card capacity	Save format	DPB Ver. 1.52 or later	TPB Ver. 2.04 to 2.18	TPB Ver. 12.12 or later
8KB	Standard	Cannot be used.		
	Compatible*			
64KB	Standard	Up to 3 units of DRCX	Up to 3 units of DRCX	Up to 3 units of DRCX
	Compatible*	Up to 4 units of DRCX	Up to 4 units of DRCX	Up to 4 units of DRCX
1024KB (1MB)	Standard	Up to 3 units of DRCX	Up to 3 units of DRCX	Up to 48 units of DRCX
	Compatible*	Up to 4 units of DRCX	Up to 4 units of DRCX	Up to 160 units of DRCX

* DRCX data can be shared with DRC and DRCA by making the save format compatible with each other. Please note, however, that there are the following limitations.

- Number of steps in total: 3000 steps
- Maximum number of points: 255 points (P0 to P254)
- Maximum number of programs: 32 programs (program No. 0 to No. 31)

16

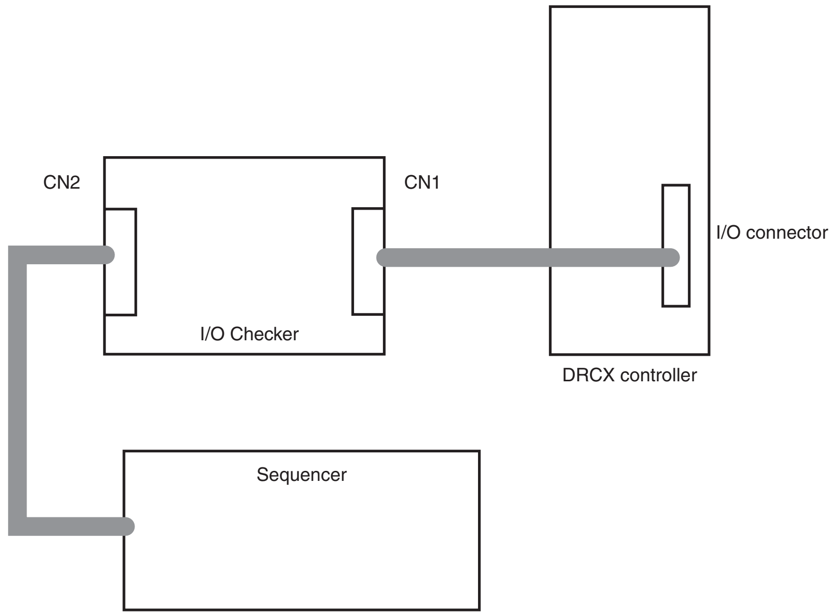
APPENDIX

16-2-2 Handling the I/O Checker

This device connects to the I/O connector of the DRCX controller and is used for pseudo-input (emulation) by means of switches and for input/output monitoring by LED display.

■ Connecting the I/O checker

1. Plug the connector marked "ROBOT I/O" into the I/O connector of the DRCX controller.
2. Plug the connector marked "JAE 50P" into connector CN1 on the right of the I/O Checker and make sure it locks.
3. Plug the connector, which is normally connected to the I/O connector of the DRCX controller, into connector CN2 on the left of the I/O Checker.



■ Operation method

1. The LED monitor turns on (lights) and off (goes out) in conjunction with the input and output.
2. The pseudo-input switch is on when set to the upper side and off when set to the lower side.

   However, the INTERLOCK and EMG inputs are opposite; they are on when set to the lower side and off when set to the upper side.

   Thus, if all of the switches are set to the lower side at first, the unit can be used for pseudo-input and as an I/O monitor.
3. The input changeover switch should be set to the EXTERNAL (upper) side to receive external input from a PLC or a similar unit.

   If the switch is set to the INTERNAL (lower) side, the switch signals on the I/O checker are input to the controller. In either case, the input monitor is handled by means of LEDs.

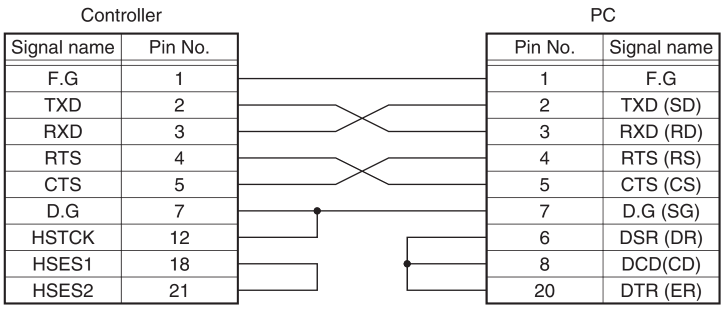
16-2-3 POPCOM communication cable

This cable is used to operate the DRCX controller from POPCOM software which runs on a PC and allows easy and efficient robot programming and operation.
This POPCOM cable is different from typical communication cables, so do not use it for other purpose.

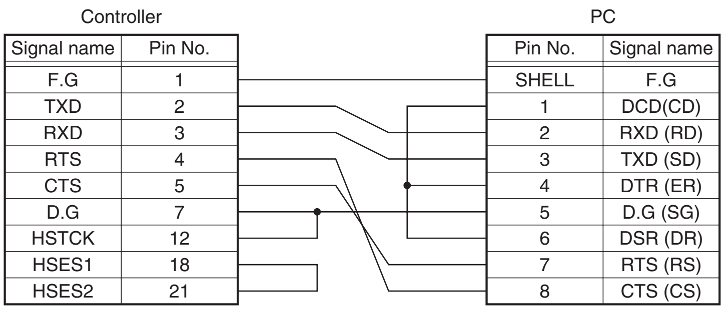
Pins 18 and 21 on the DRCX controller are used for emergency stop input. Install a normally closed (contact B) switch of at least 50mA capacity between these pins when using emergency stop from the PC. Emergency stop is triggered when the switch opens the contact between pins 18 and 21.

Input response: 5ms or less
Input current: 33.3mA (DC24V)

■ When the PC has a D-sub 25-pin connector:



■ When the PC has a D-sub 9-pin connector:



The SHELL means a metallic casing of the connector.

⚠ *CAUTION*
Pin 10 of the connector on the controller is used exclusively for connecting to the TPB. To prevent problems, do not attempt to wire anything to pin 10.

16
APPENDIX

MEMO

Revision record

Manual version	Issue date	Description
Ver. 5.03	Jul. 2007	Addition of cautions about loading the data from the controller. Clerical error corrections, etc.
Ver. 5.04	Oct. 2007	Addition of error message and caustion about loading of unsuitable robot data. Addition of customize function for END output timing in execution of dedicated I/O command. Expansion of I/O assignment function (addition of movement point zone output function, etc.). Addition of robot numbers. Clerical error corrections, etc.
	Jun. 2011	The manual's version number was changed to match that for the Japanese manual.
Ver. 6.08	Jun. 2011	The description regarding "Warranty" was changed.

User's Manual

YAMAHA DRCX
Robot Controller

Jun. 2011
Ver. 6.08
This manual is based on Ver. 6.08 of Japanese manual.