ROBOSTAR ROBOT

N1 Series

USER'S MANUAL

☐ INSTRUCTION MANUAL

√ OPERATION MANUAL

☐ PROGRAMMING MANUAL

☐ UNI-HOST MANUAL

☐ GAIN SETUP MANUAL

☐ ALARM CODE MANUAL

www.robostar.co.kr

Robostar Co., Ltd.

About Product Warranty

Products of Robostar Co., Ltd. are manufactured under the strict quality control. All the Robostar products' warranty period is one year from the date of manufacture. In this period, Robostar is only responsible for the mechanical failures due to negligence of Robostar, or the problems on design and manufacture occurring during normal use, in which the service is free of charge. However, such free service is not possible in the cases below.

(1) after the warranty period has expired
(2) failures arising due to improper repair, alteration, redeployment, or other mishandling, when, under the instruction of you or any third party
(3) failures as a result of using parts, grease, etc. which have not been designated by Robostar.
(4) failures caused by accidents, such as fire, disaster, earthquake, storms, or other natural disasters
(5) failures caused in manure, flooding, or other environment
(6) failure caused by the consumption of consumable parts
(7) failures arising when not being operated under the instructions listed in the user or instruction manual and the maintenance manual
(8) damages in cost other than the cost of robot repairing

Address and contact points of Robostar Robostar Co., Ltd.

- Head Office & Factory
  119-38, Sasa-dong, Sangnok-gu,
  Ansan-City, Gyeonggi-do, Republic of Korea (426-220)

- 2nd Factory
  960, Gosaek-dong, Gwonseon-gu,
  Suwon-City, Gyeonggi-do, Republic of Korea (441-813)

- Request for service and Inquiry of products
  - Information on Marketing
    TEL. 031-400-3600
    FAX. 031-419-4249
  - Customer Service
    TEL. 1588-4428

Robostar

www.robostar.co.kr

Constitution of User Manual

The user manual for this product is constituted as follows. When using this product first, please fully read all the manuals before use.

- **Instruction Manual**

  A controller is generally explained. Overview of the controller, installation, and interfacing to peripherals are explained.

- **Manipulation & Operation Manual**

  As well as general use of the controller, parameter setup, JOB program editing, robot operation, etc. are explained.

- **Programming Manual**

  RRL (Robostar Robot Language) which is the robot program of Robostar and how to write a robot program by using RRL are explained.

- **Unihost Manual**

  'Unihost' which is the on-line PC program of Robostar is explained.

- **Gain Setup Manual**

  How to set up the gain necessary for trial run, and the motor response performance according to change in the gain value are explained.

- **Alarm Code Manual**

  Reasons for and countermeasures against the alarms which can occur while the controller is operating is explained.

CONTENTS

N1 Controller Version

N1 controller is divided into two types of application software depending on robot type, each application software to be confirmed on the Teach Pendant screen when turning power to N1 controller ON.

Robostar
NewRo N1-Series
Ver: 03.02.03-SB
(RO 120727)

Press ENTER Key

Scara Robot, Cartesian Robot, DeskTop, Synchronous Robot Application Software (RO)

Robot Type	
SCARA	RSA-40A, RSA-50A, RSA-60A, RSA-70A, RSA-70B, RGA-80A, RGA-80B, RHA-80C, RBSA-90A, RBSA-90B, RGA-A0A, RGA-A0B, RGA-A1A, RGA-A1B, SC-TEST
XYZW	XY__, X_Z_, XYZ_, XZY_, XY_W, X_ZW, XYZW, XZYW, XYZH, XZYR, XYXY, XYZXY_, XYZXYZ, XY-TEST
DESK	RDS2_, RDS3_, RDS4_, RDT4_F, RDT4_A
DUAL	D2S, D2O, D3S, D3O, D4S, D4O, DDSO, DDOS, DDOO, DUAL
BGT	BGT

Robostar
NewRo N1-Series
Ver: 03.02.03-SB
(TR 120727)

Press ENTER Key

Robot application software for transferring semiconductor (TR)

Transfer Robot Type	
TR	TML__, TMS__, CML__, ROSEP300, ROSEP200
BGT	BGT

Caution
▶ Before running a robot, be sure to check application software and robot type.
▶ Normal operation is possible only when the robot in use and application software on N1 controller are consistent.

How to mark axis by robot type configuration

A minimum of 2 axes to maximum of 6 axes are displayed on the Teach Pendant screen according to robot type configuration.

```
<RSA80B : GEAR>
      GEAR   RATIO
1 axis: 1            2 axes: 1
3 axes: 1            4 axes: 1.25
5 axes: 1            6 axes: 1
```

The number of axis and axis names displayed on the Teach Pendant screen are changed according to the robot type.

```
<XYZW : JONT>
      JONT MOTION:   X
Mv: 3000             At: 0.80
Jv: 500              Ip: 10
It: 10               Jk: 50

1axis    2axis       3axis    4axis
5axis    6axis
```

F1

F6

Able to select a maximum of 6 axes, respectively, according to robot type.

F1: 1axis F2: 2 axis F3: 3 axis

F4: 4 axis F5: 5 axis F6: 6 axis

Caution ▶ The name of an axis is subject to change depending on robot type configuration on N1 controller and an axis may not be displayed.

● Teach Pendant

N1 controller supports Teach Pendant with two types of model (TPM, TP6). If wrongly selecting a Teach Pendant used by robot type or having a different configuration, it may cause a maloperation.

TPM(T/P for 2~4axis)	TP6(T/P for 2~6axis)
	

Caution ▶ When the Teach Pendant in use and the one set to N1 controller have different models, it may cause maloperation or a failure in Key input.

N1 Controller PARA Changes

Parameter functions are changed according to N1 controller's application software and part of initial configuration is changed. Parameters "IRNG",UIRNG" functions are changed along with changes to default values such as "Robot Type Configuration", "TP Configuration" , "AUTO SERVO ON Configuration", "MAX CH Configuration".

<table>
<tr><td colspan="8">RO & TR ROBOT COMPARE CHART</td></tr>
<tr><td colspan="5">PARA</td><td>RO</td><td>TR</td><td rowspan="2">Confrence</td></tr>
<tr><td rowspan="9">N1</td><td colspan="4">Version</td><td>(RO 130114)</td><td>(TR 130114)</td></tr>
<tr><td rowspan="3">LOCAL</td><td rowspan="2">PARA(1)</td><td colspan="2" rowspan="2">1.ROBOT CONF</td><td>5. IRNG</td><td>5. IRNG</td><td>1.2.5.5 IRNG</td></tr>
<tr><td>6. UIRNG</td><td>6. UIRNG</td><td>1.2.5.6 UIRNG</td></tr>
<tr><td>PARA(2)</td><td colspan="2">1. SETUP</td><td>1. SCARA
2. XYZW
3. DeskTop
4. Dual
5. BGT</td><td>1. TR
5. BGT</td><td>1.2.8 SETUP</td></tr>
<tr><td rowspan="4">PUB</td><td rowspan="2">HW CONF(0)</td><td rowspan="2">4. TP</td><td>2. DEADMAN S/W</td><td>DIS</td><td>ENB</td><td rowspan="2">1.3.1.4 TP</td></tr>
<tr><td>3. MODEL</td><td>TPM</td><td>TP6</td></tr>
<tr><td>HW CONF(0)</td><td colspan="2">5. SVON</td><td>ENB</td><td>DIS</td><td>1.3.1.5 SVON</td></tr>
<tr><td>HW CONF(1)</td><td colspan="2">2. MAX CH</td><td>2</td><td>3</td><td>1.3.1.8 MAX CH</td></tr>
</table>

For further details or how to configure, refer to each parameter item related.

Chapter 1 Parameter Mode

☞ Parameter setup is very important in that 50% or more causes to the error occurring in robot operation result from miss-setup of the parameter value.
☞ For the basic parameter values, refer to the contents of a label attached to the robot.
☞ Basic parameter setup is completed when manufactured in a factory, but the parameter value needs to be inevitably modified in the actual installation or operation site depending on the environmental conditions. At this time, please port the customer support team of Robostar.

1.1 Overview

■ This mode is for setup the configuration of the robot and the use environment.
■ Before operating the robot (mechanism + controller), be sure to confirm the Set-up values, and re-setup is needed, if necessary.
- Data related to the mechanical and electrical specification of the robot
- Basic operation method of the robot, such as Origin searching, special functions, etc.
- Computer and data communication protocol

■ At the time of shipment of the robot (mechanism + controller), the basic parameters have been set up.
■ When the setup is completed, be sure to record, or backup in a computer, the value for preparation for the change due to wrong use, etc.

■ Kinds of Parameter Groups

Group	Details	Remarks
BODY	Sets the configuration of the robot mechanism.	PARA(0)
MOTION	Sets robot operation related parameters.	PARA(0)
GAIN	Sets the gain values of each axis.	PARA(0)
PROTECT	Sets the protection related parameters.	PARA(0)
ROBOT CONF	Sets the environment of use of the robot.	PARA(1)
FIX JOB	Sets the initial operation job when SYSTEM MODE is executed.	PARA(1)
SETUP	Initializes the robot parameters.	PARA(2)
PUB	Sets the public parameters.	PUB
ORG	Sets the velocity, sequence, sensor, etc. of Origin searching execution.	

1.1.1 Parameter Mode Flowchart

1. Parameter includes 4 Groups.

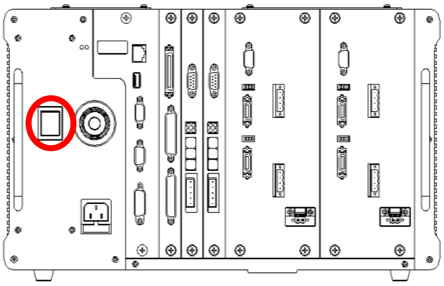
- PARAMETER(0) : Sets the configuration of the robot, motor gain, and protection system parameters.
- PARAMETER(1) : Sets how to operate ROBOT CONF and FIX/SEL JOB.
- PARAMETER(2) : Initializes the robot parameters.
- PUB PARAMETER : Sets the common setup, robot information, communication, external interface card, external I/O ports, pallet work, necessary data, and password.

2. Parameter Menu

- 4 PARA
  - F1 SEL (ROBOT)
    - PARA(0)
      - 1 BODY — CONF, LENG, RANG, OFFSET, JNT [TYPE, GEAR, PTCH, ZW], TOOL, MOTOR [ENCTY, WATT, MTYPE, HALL, M-TURN]
      - 2 MOTION(0) — ENC, JONT, LINR, DISP, MOVE, SENS, LIMIT
      - 2 MOTION(1) — INIT_V, DEC_ZONE, STOP_T
      - 3 GAIN — PVG_1st, PVG_2nd, FILTER
      - 4 PROTECT(0) — SET, FOW, OVS, OVT, OVA, OVL, IPA, IPE
      - 4 PROTECT(1) — TRQ, TOL, OND, DB [ALARM, NORMAL]
    - PARA(1)
      - 1 ROBOT CONF — RENB, USAX, XENB, P/R, IRNG, UIRNG, SYNC [ENB, AXIS, DIR, ALARM]
      - 2 FIX JOB — ENB, SELJOB
    - PARA(2)
      - 1 SETUP
  - F1 SEL (BGT)
    - 1 ENABLE
    - 9 SETUP
  - F2 INFO
  - F3 PUB
    - 1 HW CONF(0)
      - 1 TMR
      - 2 COMM — RS232C, FIELD BUS [CARD, USER I/O, PROFIBUS ENDIAN, MAP EXTENTION]
      - 3 I/O
      - 4 TP — EMG TYPE, DEADMAN S/W, MODEL
      - 5 SVON
      - 5 A I/O
    - 2 PALLET
      - F1 NO — DATA, CNT, OUT, PATTERN
    - 3 PLC
      - 1 PNO
      - 2 AUTO
      - 3 IO ENB
      - 4 BCD
      - 5 OCL
    - 4 ETC
      - 1 W.TIMER
      - 2 CINF
      - 3 PASS — MODE, VALUE
      - 4 NAME
      - 5 BRK PASS
  - F4 EXIT

3. Procedure of Opening Parameter Group Screen

Step 1. **During AUTO RUN**



Supply the power.
(If error occurs, settle it according to the error code.)

Step 2. **Move to MAIN**

```
        SYSTEM   MODE

CH   MODE   J_NUM     STATE
 1   AUTO NONE     IDL
 2   AUTO NONE     IDL


 1R    2R            EXIT
```

F4

EXIT SYSTEM MODE.

```
         Robostar
      NewRo N1-Series
    Version: 02.00.ST,05
       DATE: 111201

      Press ENTER Key
```

ENTER

Open Software Version.

```
<MAIN MENU>
1. JOB              2. RUN
3. HOST             4. PARA
5. ORIGIN           6. I/O
7. SYSTEM           8. GPNT
9. INT/FLT          A. ALARM

                       SELECT #
```

Open MAIN MENU SCREEN.

Step 3. **PARAMETER**

```
<MAIN MENU>
1. JOB              2. RUN
3. HOST             4. PARA
5. ORIGIN           6. I/O
7. SYSTEM           8. GPNT
9. INT/FLT          A. ALARM

                       SELECT #
```

4
L

4: Select PARA

Step 4. **Enter PASSWORD**

```
Password = * * * *
```

1 Q

9 I

① Enter Password (deault: "1111").
(When for initial setup of the password)

```
Password = * * * *


Invalid!!
```

If the wrong password is entered, "Invalid" message appears.

Step 5. **Select PARAMETER CHANNEL**

```
<PARAMETER>
 NO              TYPE
*CH1             XYZW
 CH2             XY_TEST


SEL    INFO    PUB    EXIT
```

Using "robot selection" ▲▼ key, select the robot.

```
<PARAMETER>
 NO              TYPE
*CH1             XYZW
 CH2             XY_TEST


SEL    INFO    PUB    EXIT
```

F1

Select "*" robot and "F1" Key

Step 6. **Change to PARAMETER PAGE**

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT


                 group #
```

Open PARAMETER(0) screen.

```
< XYZW : PARA(1)>
 1: ROBOT CONF
 2: FIXJOB



                group #
```

Pg Dn

```
< XYZW : PARA(2)>
1: SETUP



                group #
```

Open PARAMETER(2) screen.

■ Comment

▷ Password can be change as follows.

① In an initial menu screen, select 4:PARA → PUB(F3) → 4:ETC →3:PASS. Then, enter 4 digits of password and ESC → ENTER. Change has been completed.
(If the entered PASSWORD value is "0", the above STEP 4 may be omitted.)

1.1.2 Parameter Setup

(1) Storing the changed parameter

Step 1. **Change PARAMETER**

```
<XYZW : GEAR>
        GEAR RATIO
X: 1              Y : 1
Z: 1              W: 1.25
```

ENTER

After changing the Set-up values in a parameter screen, select “ENTER”.

Step 2. **Store PARAMETER**

```
<XYZW : GEAR>
        GEAR RATIO
X: 1              Y : 1
Z: 1              W: 1.25


Update OK? (ENT/ESC)
```

ESC

Press “ESC” to finish editing.

Update OK?(ENT/ESC) message is displayed.

```
<XYZW : GEAR>
        GEAR RATIO
X: 1              Y : 1
Z: 1              W: 1.25


Update OK? (ENT/ESC)
```

ENTER

Press “ENTER” to complete the update.

```
<XYZW : JNT>
      GROUP : JOINT
1: TYPE           2: GEAR
3: PTCH           4: ZW


                item #
```

“GEAR” Parameter update is completed.

■ Comment

▷ If any key other than ENTER key is pressed after changing the value, the previous value is recovered.

▷ If the power is turned off before finally saving the value, PARAMETER values may not be updated. So, be sure to save the values before powering OFF.

(2) When not storing the changed parameter:

Step 1. **Changed PARAMETER**

```
<XYZW : GEAR>
        GEAR RATIO
A: 1              Y : 1
Z: 1              W: 1.25
```

ENTER

After changing the Set-up values in Parameter screen, press "ENTER".

Step 2. **Cancel PARAMETER change**

```
<XYZW : GEAR>
        GEAR RATIO
A: 1              Y : 1
Z: 1              W: 1.25


Update OK? (ENT/ESC)
```

ESC

Press "ESC" to finish changing.

Update OK?(ENT/ESC) message is displayed.

```
<XYZW : GEAR>
        GEAR RATIO
A: 1              Y : 1
Z: 1              W: 1.25


Update OK? (ENT/ESC)
```

ESC

Press "ESC" to cancel the change.

```
<XYZW : JNT>
        GROUP : JOINT
1: TYPE           2: GEAR
3: PTCH           4: ZW


                    item #
```

Cancelling the change is completed.

1.2 ROBOT PARAMETER

1.2.1 BODY

1. Setup Items

BODY(0)	Details	Range	Set-up value
1 : CONF	Sets up the configuration of a robot mechanism.	See the explanation	
2 : LENG	Sets up, in SCARA robot, the arm length of A and B axis.	See the explanation	
3 : RANG	Sets up the work area and motion range of the mechanism.	-99999 to 99999	
4 : OFFS	Difference value between the coordinate origin of the mechanism and that of the controller.	-99999 to 99999	
5 : JNT	Sets up the types of mechanism per axis, reduction ratio, pitch, ZW synchronization.	See the explanation	
6 : TOOL	Sets up the offset value of a tool attached to the end part of Z axis of the mechanism.	-1000 to 1000	
7 : MOTOR	Sets up the motor capacity of each axis of the mechanism.	See the explanation	

2. Setup Sequence

Step 1. **Move to PARAMETER(0) screen**

```
<XYZW : PARA(0)>
1: BODY              2: MOTION
3: GAIN              4: PROTECT



                      group #
```

Open PARAMETER(0) Group screen.

Step 2. **Move to BODY(0) screen**

```
<XYZW : PARA(0)>
1: BODY              2: MOTION
3: GAIN              4: PROTECT



                      group #
```

1
Q

Select 1: BODY.

```
<XYZW : BODY(0)>
1: CONF              2: LENG
3: RANG              4: OFFS
5: JNT               6: TOOL
7: MOTOR

                      item #
```

Move to BODY(0) screen.

1.2.1.1 CONF --- Display the configuration of the robot mechanism.

1. Setup Sequence

Step 1. **Move to BODY(0) screen**

```
<XYZW : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

            item #
```

Open BODY(0) screen.

Step 2. **CONF setup screen**

```
<XYZW : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

            item #
```

1 Q

Select 1: CONF.

```
<XYZW : CONF>
ROBOT   TYPE: XYZW
1: SCARA         2: XYZW
3: DESK          4: DUAL
5: TR
                 8: BGT
            item #
```

Select robot type of the selected channel.

■ Comment

▷ Default axis of a controller is 4 axes. When 2 to 3 axes are used, PARA(1) -> 1: ROBOT CONF -> 3:XENB Set-up values are to be changed to "DIS", per unused axis.

▷ When changing CONF, only the robot type of the selected channel.

▷ Basic PARAMETER Set-up values according to the robot type are adapted when changing the type of PARA(2) -> 1: SETUP.

```
<XYZW : CONF>
ROBOT TYPE: XYZW
1: SCARA         2: XYZW
3: DESK          4: DUAL
5: TR
                 8: BGT
```

Currently set robot type

```
<XYZW : CONF>
ROBOT TYPE: XYZW
1: XYZW          2: XZYW
3: XYZR          4: XZYR
5: XYXY          6: XY-TEST
```

Cartesian coordinat robot
No.4 axis is used
W: rotational movement
(angle is displayed)
H : linear movement(mm)

1.2.1.2 LENG --- Arm length setup

1. Setup Sequence

Step 1. **Move to BODY(0) screen**

```
<RGA80B : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                 item #
```

Open BODY(0) screen.

Step 2. **LENG setup screen**

```
< RGA80B: BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                 item #
```

2
R

Select 2: LENG. (Applied for SCARA robot only)

```
< RGA80B: LENG>
      ARM LENGTH (mm)
A-ARM: 450
B-ARM: 350
```

Set A- and B-arm length of horizontal articulated multi-joint.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

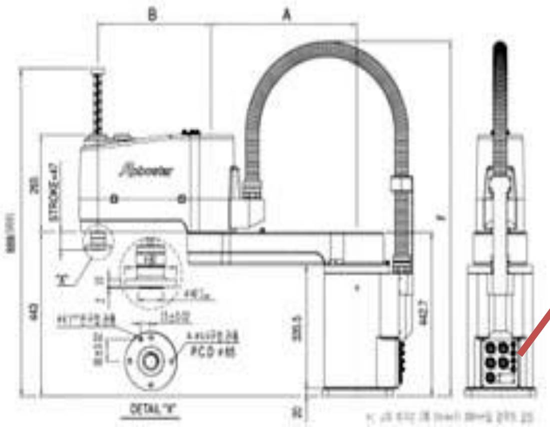
■ Comment

▷ This LENG is applied for a horizontal articulated (SCARA) robot only.

▷ The arm length has been calculated by using a jig at a final inspection process before shipment from the factory. If it is to be re-entered, enter the values written on a name plate at the rear side of the robot body.

▷ For Cartesian coordinate robot, if LENG is selected, "Only use for SCARA!" message appears. Then, press "ENTER" key.

CAUTION	▶ Applied for a horizontal articulated (SCARA) robot only.



NewRo AC ROBOT

SCARA ROBOT

MODEL : RGA80A

LENGTH : A = 450, B = 350

STORKE : Z = 200

CONTROLER : N1– 4 – 4211 –NX

Ser. NO : 0412 - 007

CAUTION

▶ If the arm length is wrong set, interpolation (LMOV, AMOV, CMOV) and palletizing (PMOV) do not perform exactly. That is, the robot moves to a position different from the acual taught positional coordinate.

▶ When first installing the robot, be sure to check the Set-up values of the controller and the name plate of the mechanism.

1.2.1.3 RANG --- Motion range (software limit) setup

1. Setup Sequence

Step 1. **Move to BODY(0) screen**

```
<XYZW : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                    item #
```

Select 3: RANG.

Step 2. **RANG setup screen**

```
<XYZW: BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                    item #
```

3
S

Select 3: RANG.

```
<XYZW: RANG>
    SOFTWARE LIMIT RANGE
X : -10000       10000
Y : -10000       10000
Z : -10000       10000
W: -10000        10000
```

Set RANG according to a robot type.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ SCARA (horizontal articulated) : A-axis: ±110˚ , B-axis: ±145˚, Z-axis: 0 to 300mm, W-axis: ±360˚

▷ Cartesian coordinate robot: The stroke corresponds to the motion range.

Importantt

▶ Since the operation range of the parameter is the software limit, it must be equal to or less than the hardware limit.

▶ If the position coordinate deviating from the parameter’s operation range is taught, "Range Over" is displayed.

▶ Although "Range Over" has not occurred in point teaching, "S/W LIMIT" may occur during interpolation (especially, AMOV, CMOV) or palletizing (PMOV).

▶ Move the workpiece or set the installation location of the robot within the robot’s operation range.

1.2.1.4 OFFS --- Offset setup

1. Setup Sequence

Step 1. **Move to BODY(0) screen**

```
<RGA80B: BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                 item #
```

Open BODY(0) screen.

Step 2. **Move to OFFSET screen**

```
<RGA80B: BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                 item #
```

Select 4: OFFS.

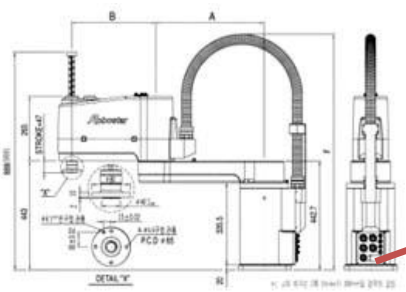
4
L

```
<RGA80B: OFFS>
      OFFSET SETTING



EDIT   CALIB   ACAL   EPOS
```

Offset value changing item

■ Comment

▷ Changing the offset value has to be done when the mechanisms deviates due to the collision or the positions changes after the replacement of a motor.



Axis	P/G	Length	**offset**
A	1/89	449.73	**1.80**
B	1/59	349.83	**6.13**
Z	25/1.25	200	**-3.01**
W	1/30	360	

(1) Changing offset value of EDIT item

1. Setup Sequence

Step 1. **Move to EDIT item**

```
<RGA80B: OFFS>
        OFFSET SETTING



EDIT    CALIB    ACAL    EPOS
```

F1

Move to EDIT screen.

Step 2. **Set up EDIT**

```
<RGA80B: OFFS>
        OFFSET SETTING
A : 0
B : 0
Z : 0
W: 0
```

OFFSET change screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Enter "Offset value" written on a name plate. (Initial Set-up value)

▷ This is used to enter the conventional offset parameter after replacing the controller or initialization of the parameters.

▷ In order to enable the changed offset value when changing the offset value by using EDIT function, be sure to perform a procedure of Software Reboot or Power Reboot.

▷ Correcting the offset value must be performed in the following cases.

Important

- ▶ The conventional points deteriorates due to the positional deviation of the mechanism because of the collision
- ▶ Positional change is generated after replacing the motor
- ▶ The interpolation (AMOV, LMOV) and PMOV do not correctly work
- ▶ The changed offset values are effective after Origin searching is performed.
- ▶ An ABS type motor which does not require the Origin searching, be sure to re-supply the power.

(2) Changing offset value using ZERO CALIB(CALIBRATION) menu

1. Setup Sequence

Step 1. **Move to CALIB item**

```
<RGA80B: OFFS>
        OFFSET SETTING



EDIT     CALIB     ACAL     EPOS
```

F2

Select CALIB.

Step 2. **Move to CALIBRATION screen**

```
<RGA80B: OFFS>
      ZERO CALIBRATION



MDI      CURR
```

F2

Select CURR.

```
<RGA80B: OFFS>
  CURR SET                  B
A: 0              B : 0
Z: 0              W: 0


A        B        Z        W
```

Open CALIB setup screen

Step 3. **Set up CALIBRATION**

```
<RGA80B: OFFS>
  CURR SET                  B
A: 1.23           B : -5.25
Z: 0              W: 0


A        B        Z        W
```

Move a robot to be aligned with the arrow marks on the robot (▲▼).

Step 4. **X-AXIS CALIBRATION**

<RGA80B: OFFS>
CURR SET B
A: 1.23 B : -5.25
Z: 0 W: 0

A B Z W

F1

Store X-AXIS.

<RGA80B: OFFS>
CURR SET B
A: 1.23 B : -5.25
Z: 0 W: 0

Save A? (ENT/ESC)

ENTER

Select whether to save CALIB.

<RGA80B: OFFS>
CURR SET B
A: 0 B : -5.25
Z: 0 W: 0

X Y Z W

Confirm X-AXIS CALIB.

Step 5. **Y-AXIS CALIBRATION**

<RGA80B: OFFS>
CURR SET B
A: 0 B : -5.25
Z: 0 W: 0

A B Z W

F2

Store Y-AXIS.

<RGA80B: OFFS>
CURR SET B
A: 0 B : -5.25
Z: 0 W: 0

Save B? (ENT/ESC)

ENTER

Select whether to save CALIB.

```
<RGA80B: OFFS>
  CURR SET              B
A: 0            B : 0
Z: 0            W: 0


A       B       Z       W
```

Confirm Y-AXIS CALIB.

```
<RGA80B: OFFS>
  CURR SET              B
A: 0            B : 0
Z: 0            W: 0


A       B       Z       W
```

F3 ENTER

F4 ENTER

Set up the remaing axes, using the same procedure.

Step 6. After setup, move to Group screen

```
<RGA80B: OFFS>
  CURR SET              B
A: 0            B : 0
Z: 0            W: 0


Please Reboot!
```

ESC

After CALIBRATION is performed, it is necessary to reboot the controller.

```
<RGA80B: OFFS>
      ZERO CALIBRATION



MDI     CURR
```

ENTER

ZERO CALIBRATION screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ If it is necessary to perform the software based Origin searching after the mechanical Origin searching has been completed, have the current position offset from the origin by using CALIBRATION parameter.

▷ If there are not ▼▲ markings, arms A and B of SCARA robot straight extend.

Important

▶ Before CALIB, be sure to move the mechanical plart on the mechanical origin.

▶ Only an absolute motor is supported.

(3) Changing offset value using ACAL(ABSOULTE CALIBRATION) menu

1. Setup Sequence

Step 1. Move to ACAL item

```
<RGA80B: OFFS>
      OFFSET SETTING



EDIT    CALIB    ACAL    EPOS
```

F3

Select ACAL.

Step 2. Set up A-AXIS ACAL

```
<RGA80B: OFFS>
      ABS CALIBRATION
A: 1.23          B: -5.25
Z: 0.24          W: 1.31


A       B       Z       W
```

F1

Select A-AXIS offset.

```
<RGA80B: OFFS>
     ZERO CALIBRATION
A: 1.23          B : -5.25
Z: 0.24          W: 1.31


Save A?(ENT/ESC)
```

ENTER

Press ENTER to change the offset.

Step 3. Adapt A-AXIS ACAL

```
<RGA80B: OFFS>
      ABS CALIBRATION
A: 4.52          B: -5.25
Z: 0.24          W: 1.31


A       B       Z       W
```

Adapt A-AXIS ABS offset

Step 4. Adapt B, Z, W-AXIS ACAL

```
<RGA80B: OFFS>
      ABS CALIBRATION
A: 4.52          B: -5.25
Z: 0.24          W: 1.31


A       B       Z       W
```

F2	ENTER
F3	ENTER
F4	ENTER

Set up the remaing axes, using the same procedure.

Step 5. After setup, move to the higher menu

```
<RGA80B: OFFS>
        ABS CALIBRATION
A: 4.52            B: -7.25
Z: 0.54            W: 1.87


Please Reboot!
```

ESC

Escape to OFFSET SETTING screen.

```
<RGA80B: OFFS>
        OFFSET SETTING




EDIT     CALIB     ACAL     EPOS
```

Escape to Group screen.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ In the case that an absolute motor is replaced or the positional error occurs due to the external shock, by using ACAL Parameter, the error can be corrected.

Important

- ▶ After the controller is replaced and the parameters are initialized, the conventional OFFS and EPOS value are required.
- ▶ Before performing the Absoulte Calibration, engage the jig for the mechanism with the axis to be performed and move it.
- ▶ After the axis moving is completed, perform ACAL.

CAUTION

- ▶ When performing Z-axis ACAL, W-AXIS must be set up so that the offset value is within the permissible error range.
- ▶ When confirming W-AXIS, it is regarded to be normal if the current position value is within ± 2° from EPOS W-AXIS value after it is moved to the jig position.
- ▶ If ACAL has been abnormally performed, the robot may operate outside the Limit range and thus the accident may occur. Make sure to check it.
- ▶ This is a method to check that ACAL has been normally performed.

① ACAL completed -> ② Controller Reboot -> ③ Move the relevant axis to ACAL position -> ④ Compare with EPOS value and check ACAL performing state.

(4) Changing offset value using EPOS(END POSITION) menu

1. Setup Sequence

Step 1. **Move to EPOS item**

<RGA80B: OFFS>
OFFSET SETTING

EDIT CALIB ACAL EPOS

F4

Select EPOS.

Step 2-1. **MDI Mode**

<RGA80B: OFFS>
END ANGLE SET
A: 13.25 B: -98.73
Z: -13.50 W: 107.60

MDI CURR

F1

Select MDI Mode.

<RGA80B: OFFS>
MDI SET B
A: 13.25 B: -98.73
Z: -13.50 W: 107.60

Open MDI setup screen of EPOS.

<RGA80B: OFFS>
MDI SET B
A: 114.25 B: -98.73
Z: -13.50 W: 107.60

Change EPOS value of the relevant axis.

1 Q	1 Q	4 L	. W
2 R	5 M	ENTER	

<RGA80B: OFFS>
MDI SET B
A: 114.25 B: -98.73
Z: -13.50 W: 107.60

Save ?(ENT/ESC)

ESC

Select ESC to save data.

Select ENTER to save data

```
<RGA80B: OFFS>
  MDI SET              B
A: 114.25      B: -98.73
Z: -13.50      W: 107.60

Save ?(ENT/ESC)
```

ENTER

```
<RGA80B: OFFS>
  MDI SET              B
A: 114.25      B: -98.73
Z: -13.50      W: 107.60

Please Reboot!
```

To adapt the offset changed value, the controller reboot is required.

Step 2-2. CURR Mode

```
<RGA80B: OFFS>
        END ANGLE SET
A: 114.50      B: -97.65
Z: -13.50      W: 107.60

MDI    CURR
```

F2

Select CURR Mode.

```
<RGA80B: OFFS>
  CURR SET             B
A: 114.50      B: -97.65
Z: -13.50      W: 107.60

A      B      Z      W
```

F1

Change A-AXIS EPOS value.

```
<RGA80B: OFFS>
  CURR SET             B
A: 114.50      B: -97.65
Z: -13.50      W: 107.60

Save A?(ENT/ESC)
```

ENTER

Select ENTER to save data.

```
<RGA80B: OFFS>
  CURR SET             B
A: 113.20      B: -97.65
Z: -13.50      W: 107.60

A      B      Z      W
```

F2 ENTER

F3 ENTER

F4 ENTER

Set up the remaing axes, using the same procedure.

Select ESC to save data.

```
<RGA80B: OFFS>
  CURR SET              B
A: 113.20         B: -97.65
Z: -13.50         W: 107.60


Save ?(ENT/ESC)
```

ESC

```
<RGA80B: OFFS>
  CURR SET              B
A: 113.20         B: -97.65
Z: -13.50         W: 107.60


Save ?(ENT/ESC)
```

ENTER

Select ENTER to save data.

```
<RGA80B: OFFS>
  CURR SET              B
A: 113.20         B: -97.65
Z: -13.50         W: 107.60


Please Reboot!
```

After checking that "Please Reboot!" message is displayed, reboot the controller.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ This parameter has the coordinate values of the mechanisms which are references to ACAL execution.

▷ This is entered by using MDI and CURR Mode when replacing the controller or initializing the parameters.

1) MDI Mode

   In MDI Mode, the values written on the name plate is entered.

2) CURR Mode

   This mode is used after moving the relevant axis to the reference position by using the jig of the mechanism.

CAUTION	▶ When setup EPOS value in CURR Mode, the value must be changed only when the offset values to which CALIBRATION is adapted are set up.

(5) Confirmation method for the changed offset value

1. Confirmation Sequence

Step 1. **Move to MAIN screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

Open an initial menu screen.

Step 2. **Move to JOB EDIT screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

1
Q

Select JOB.

```
<JOB MAIN>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST


SEL    INFO            EXIT
```

▲
▼

Select a robot using "Robot Select " ▲▼ key.

```
<JOB MAIN>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST


SEL    INFO            EXIT
```

F1

Select a robot by marking *, and then "F1" key.

Step 3. **Move to CURR Mode**

```
<XYZW : JOB EDIT>




DIR    JEDIT
```

F1

Select DIR.

```
<XYZW : DIR>
 * 0. A      23      STEP
   .__      .__
   .__      .__
   .__      .__
   .__      .__
   .__      .__
 COPY    REN      DEL     EDIT
```

▲ ▼

Select Job using "JOB Select " ▲▼ key.

```
<XYZW : DIR>
 * 0. A      23      STEP
   .__      .__
   .__      .__
   .__      .__
   .__      .__
   .__      .__
 COPY    REN      DEL     EDIT
```

F4

Select EDIT.

```
<XYZW : EDIT>
 JOB NAME= A




 PROG    POINT    PLC
```

F2

Select point edit screen.

```
<XYZW : EDIT>    V:         50
F: A       P: 0    US    B L
X: 0               Y : 0
Z: 0               W: 0


MDI      CURR      EDIT   QUIT
```

F2

Select CURR teaching.

Step 4. **Confirm changed offset value**

```
<XYZW : EDIT>    V:         50
F: A       P: 0    US    B L
X: 0               Y : 0
Z: 0               W: 0


EXCH CORD   PJUMP   FWRD
```

Confirm that the coordinate value of all axes is "0".

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

1.2.1.5 JNT --- Joint setup

TYPE - JOINT TYPE setup

1. Setup Sequence

Step 1. **Move to JNT screen**

```
<RGA80B : JNT>
      GROUP : JOINT
1: TYPE            2: GEAR
3: PTCH            4: ZW


                   item #
```

Open JNT screen.

```
<RGA80B : JNT>
      GROUP : JOINT
1: TYPE            2: GEAR
3: PTCH            4: ZW


                   item #
```

1
Q

Select "TYPE"

Step 2. **Change TYPE value**

```
<RGA80B : TYPE>
          JOINT TYPE
A: REV             B : REV
Z: TRAN            W: REV
```

TRAN : Tanslation joint

REV : Revolution joint

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ TYPE : Select one among Revolution Joint and Translation Joint

Item	Set-up value	Description
TYPE	TRAN	Translation (linear movement)
	REV	Revolution Joint (rotating movement)

GEAR --- Reduction ratio(GEAR RATIO) setup

1. Setup Sequence

Step 1. **Move to GEAR**

```
<RGA80B : JNT>
     GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW

                 item #
```

Open JNT screen.

```
<RGA80B : JNT>
     GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW

                 item #
```

2
R

Select “GEAR”

Step 2. **Enter GEAR RATIO**

```
<RGA80B : GEAR>
     GEAR RATIO
A: 89            B: 59
Z: 1.25          Z: 30
```

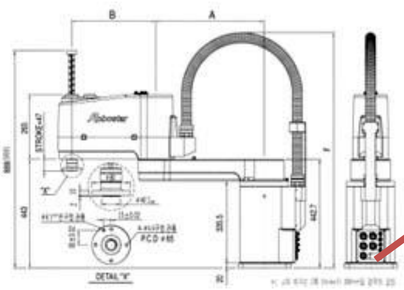
Store the value written on a robot name plate.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Reduction ratio is entered as the value written on a robot name plate. (In the contents of the name plate, enter “G value”.)

▷ Reduction ratio of a motor and a mechanism (including pulley ratio)



Axis	P/G	Length	offset
A	1/**89**	449.73	1.80
B	1/**59**	349.83	6.13
Z	25/**1.25**	200	-3.01
W	1/**30**	360	

PTCH --- PITCH setup

1. Setup Sequence

Step 1. **Move to PITCH**

```
<RGA80B : JNT>
     GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW


                   item #
```

Open JNT screen

```
<RGA80B : JNT>
     GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW


                   item #
```

3
S

Select “PTCH”

Step 2. **Change PTCH**

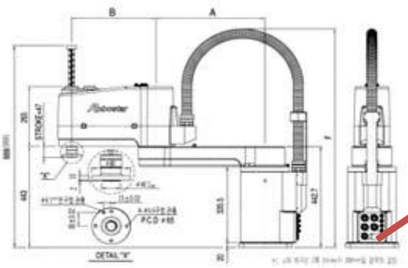
```
<RGA80B : PTCH>
     PITCH RATIO
A: 1             B: 1
Z: 1             Z: 1
```

Store the value written on a robot name plate.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Travelling distance of the mechanism per one revolution of an axis(mm) (In the contents of the name plate, enter “P value”)



Axis	P/G	Length	offset
A	**1**/89	449.73	1.80
B	**1**/59	349.83	6.13
Z	**25**/1.25	200	-3.01
W	**1**/30	360	

ZW --- AXIS synchronization

1. Setup Sequence

Step 1. Move to JNT ZW

```
<RGA80B : JNT>
      GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW

                     item #
```

Open JNT screen

```
<RGA80B : JNT>
      GROUP : JOINT
1: TYPE          2: GEAR
3: PTCH          4: ZW

                     item #
```

4
L

Select “ZW”

Step 2. Change ZW

```
<RGA80B : ZW>
      ZW AXIS SYNC
SYNC:   OFF
```

Refer to SYNC table for setup.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ ZW synchronized Set-up value of SCARA robot is as follows.

① **ZW_SYNC CHART**

	Div.	Description
SYNC	+	Setup the forward synchronization of Z, W-AXIS
	-	Setup the reverse synchronization of Z, W-AXIS
	OFF	Z, W-AXIS synchronization function is not used.

1.2.1.6 TOOL --- Work tool offset setup

1. Setup Sequence

Step 1. **Move to BODY(0)**

```
<RGA80B : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                   item #
```

Open BODY(0) screen.

Step 2. **Move to TOOL screen**

```
<RGA80B : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                   item #
```

6
N

Open TOOL screen

```
<RGA80B : TOOL>
        TOOL OFFSET 0
dX: 0
dY: 0
dZ: 0

T0     T1     T2     T3
```

F1

Move to TOOL OFFSET 0 screen.

```
<RGA80B : TOOL>
        TOOL OFFSET 3
dX: 0
dY: 0
dZ: 0

T0     T1     T2     T3
```

F4

Move to TOOL OFFSET 3 screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ TOOL offset value is used in the following cases.

(1) when a work tool is attached to the end point of Z-axis so that the work tool deviates from the center line of the Z axis
(2) when W-axis (Rotating Axis) performs the interpolation (LMOV, AMOV) and palletizing (PMOV) by using a tool end
(3) when, Cartesian coordinate system of SCARA robot, the travelling deviation (tool offset) is calculated along each direction (X, Y, and Z)

Important

▶ Be cautious that the tool Set-up values are used for basic jog operation.

Method of converting SCARA robot coordinate system (cylindrical coordinate ↔ Cartesian coordinate)

1. Setup Sequence

Step 1. **Move to MAIN screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

Open intial menu screen

Step 2. **Move to JOB EDIT screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

1 Q

JOB Select

```
<JOB MAIN>
 NO             TYPE
*CH1            RGA80B
 CH2            RGAA0B


SEL     INFO          EXIT
```

Select a robot by using "robot select " ▲▼ key.

```
<JOB MAIN>
 NO             TYPE
*CH1            RGA80B
 CH2            RGAA0B


SEL     INFO          EXIT
```

F1

Select a robot by marking * and then "F1" key.

Step 3. **Move to CURR Mode**

```
<RGA80B : JOB EDIT>




DIR     JEDIT
```

F1

Select DIR

```
<RGA80B : DIR>
 * 0. A       23      STEP
   ._         ._
   ._         ._
   ._         ._
   ._         ._
   ._         ._
 COPY    REN      DEL     EDIT
```

▲ ▼ Arbitrarily select one of the previously registered jog as "*".

```
<RGA80B : DIR>
 * 0. A       23      STEP
   ._         ._
   ._         ._
   ._         ._
   ._         ._
   ._         ._
 COPY    REN      DEL     EDIT
```

F4 Select EDIT

```
<RGA80B : EDIT>
 JOB NAME= A



 PROG    POINT    PLC
```

F2 Select POINT editing screen

```
<RGA80B : EDIT>  V :        50
F: A       P: 0       US    B     L
A: 0.26               B : 0.32
Z: 0.23               W: -0.67


MDI      CURR       EDIT      QUIT
```

F2 Select CURR

Step 4. **Coordinate system conversion**

```
<RGA80B : EDIT>  V :        50
F: A       P: 0       US    B     L
A: 0.26               B : 0.32
Z: 0.23               W: -0.67


EXCH    CORD PJUMP FWRD
```

F2 Move to ABZW cylindrical coordinate system screen

```
<RGA80B : EDIT>  V :        50
F: A       P: 0       US    B     L
X: 799.98             Y : 5.31
Z: 0.23               W: -0.67


EXCH    CORD PJUMP FWRD
```

F2 Move to XYZW Cartesian coordinate system screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Use Teach Pendant to move a robot.

▷ When teaching Cartesian coordinate system point, it is easy to teach because the robot moves linearly along X and Y coordinate.

▷ When, in Cartesian coordinate system, JOG key is used to move a robot, it is possible to confirm the orientation of Cartesian coordinate system(X, Y, and Z).

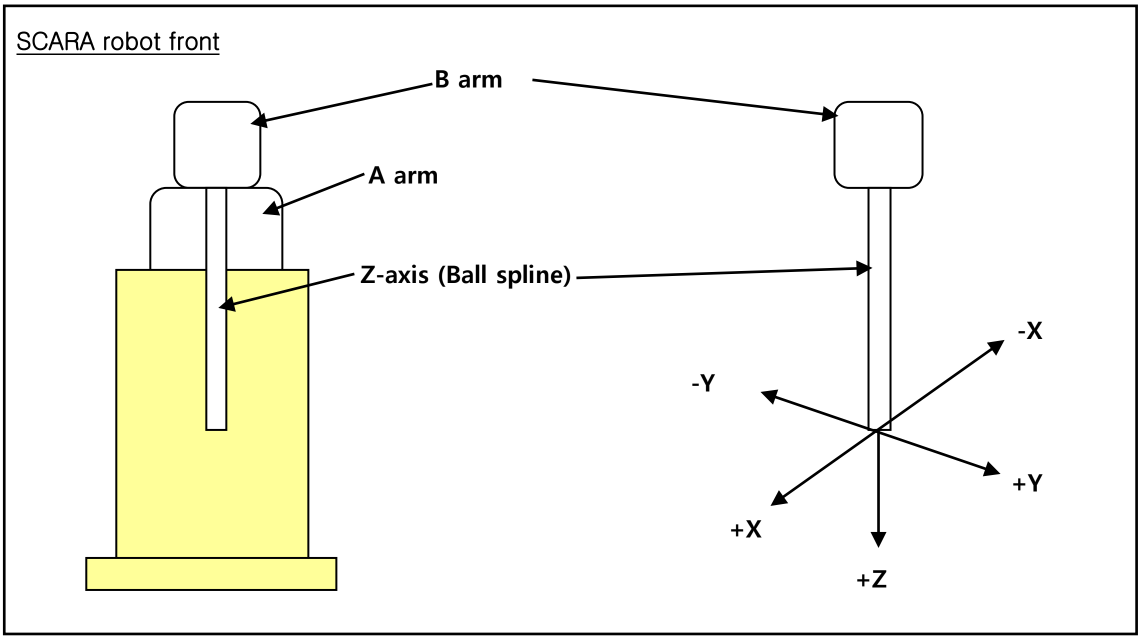
▶ How to check the coordinate system by using Teach Pendant

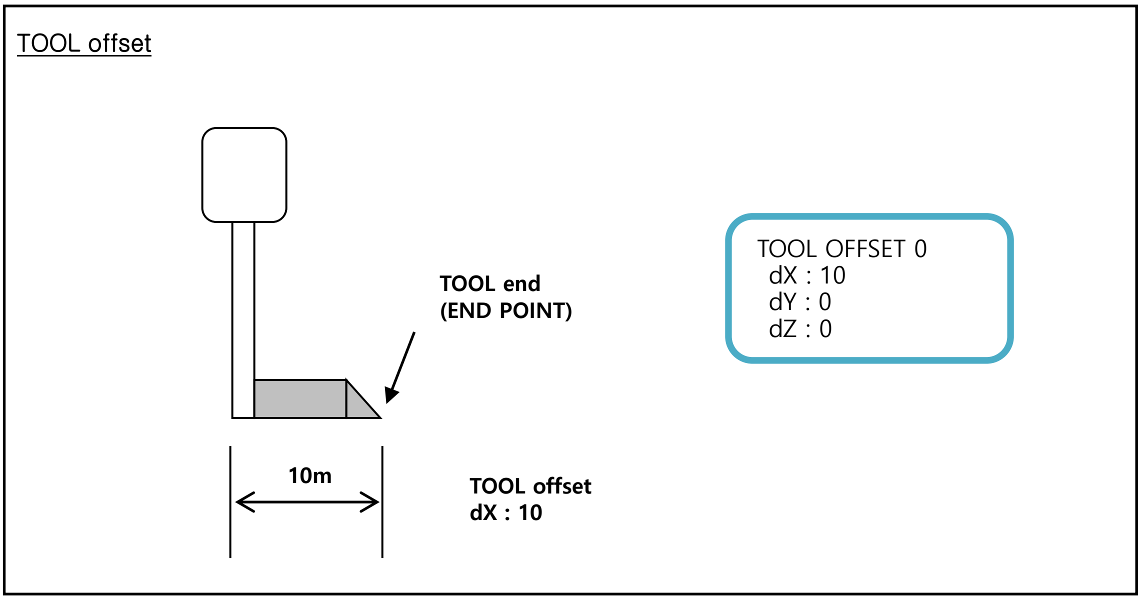
▶ When in point teaching: If converting to Cartesian coordinate system, the robot moves linearly, and thus the the teaching becomes easy.

▶ To move the robot by using a jog key, be sure to press the deadman switch.

▶ If the robot is moved by using a jog key in Cartesian coordinate system, it is possible to confirm the orientation of Cartesian coordinate system(X, Y, and Z).

How to set TOOL offset





Important

- Unless the measuring instrument is used, it is impossible to set up the exact TOOL offset.
- Select TOOL so that there is only one direction dX or dY of TOOL direction as possible.
- When attaching TOOL to Z axis, the offset must be in the one direction.

How to check the correctness of offset value

1. Checking Sequence

Step 1. **Move to POINT screen**

```
<RGA80B : EDIT> V :        50
F: A        P: 0    US    B L
A: 0.26             B : 0.32
Z: 0.23             W: -0.67


MDI    CURR    EDIT   QUIT
```

F2

On POINT screen, select CURR.

Step 2. **Check OFFS**

```
<RGA80B : EDIT> V :        50
F: A        P: 0    US    B L
A: 0.26             B : 0.32
Z: 0.23             W: -0.67


EXCH   CORD   PJUMP FWRD
```

F2

Select CORD. (A cylindrical coordinate system screen appears.)

```
<RGA80B : EDIT> V :        50
F: A        P: 0    US    B L
X: 799.98           Y : 5.31
Z: 0.23             W: -0.67


EXCH   CORD   PJUMP FWRD
```

F2

Select CORD. (Change to Cartesian coordinate system screen)

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

▶ Check that, when a robot is moved by a W-axis jog key in Cartesian coordinate system, the endpoint of a tool rotates around 1 point.
▶ At the time, if the offset is not correct, the endpoint of the tool rotates in a circle or ellipse.

1.2.1.7 MOTOR --- motor setup

1. Items to be set up

MOTOR(0)	Description	Range	Set-up value
1 : ENCTY	Encoder type setup	INC<->ABS	
2 : WATT	Motor capacity setup	See the explanation	
3 : MTYPE	Motor type setup	ROTA<->LINE	
4 : HALL	Hall sensor capacity setup	Refer to Manual	

MOTOR(1)	Description	Range	Set-up value
1 : M-TURN	Multi-turn data clearing	Refer to Manual	

2. Setup Sequence

Step 1. **Move to MOTOR screen**

```
<RGA80B : BODY(0)>
1: CONF          2: LENG
3: RANG          4: OFFS
5: JNT           6: TOOL
7: MOTOR

                 item #
```

7
G

Open BODY(0) screen.

Select 7: MOTOR

Step 2. **MOTOR(0) screen**

```
<RGA80B : MOTOR(0)>
        MOTOR SETUP
1: ENCTY         2: WATT
3: MTYPE         4: HALL

                 item #
```

Open MOTOR(0) screen

Step 3. **MOTOR(1) screen**

```
<RGA80B : MOTOR(1)>
        MOTOR SETUP
1: M-TURN

                 item #
```

Pg Dn

Open MOTOR(1) screen

ENC Type Setup --- motor encoder(ENCODER) setup

1. Setup Sequence

Step 1. **Move to ENCTYPE screen**

```
<RGA80B : MOTOR(0)>
          MOTOR SETUP
1: ENCTY            2: WATT
3: MTYPE            4: HALL



                      item #
```

1 / Q

Open MOTOR(0) screen

Select 1: ENCTYPE

Step 2. **Set up ENCTYPE**

```
<RGA80B : ENCTY>
          ENCODER TYPE
A: ABS              B : ABS
Z: ABS              W: ABS
```

ENTER

① Setup change using "ENTER"

INC ↔ ABS changeable

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ ENCODER Set-up value are as follows..

Item	Set-up value	Description
ENCODER TYPE	ABS	Absolute Type Encoder
	INC	Incremental Type Encoder

▷ In the contents of the name plate, enter "Encoder".

Axis	Motor	Encoder
A	750W/A4	**Abs**.131072
B	400W/A4	**Abs**.131072
Z	400W/A4	**Abs**.131072
W	400W/A4	**Abs**.131072

CAUTION

▶ Reboot a controller after Parameter is changed, in order to adapt the changed Parameter.

Motor Watt Setup --- motor capacity setup

1. Setup Sequence

Step 1. **Move to WATT screen**

<RGA80B : MOTOR(0)>
MOTOR SETUP
1: ENCTY 2: WATT
3: MTYPE 4: HALL

item #

2
R

Open MOTOR(0) screen

Select 2: WATT

Step 2. **Set up WATT**

<RGA80B : MOTOR>
MOTOR [Watt]
A : 756 B : 406
Z : 406 W :206

① Refer to MOTOR TABLE for setup.

Step 3. **Store WATT setup**

<RGA80B : MOTOR>
MOTOR [Watt]
A : 756 B : 406
Z : 406 W :206

Update OK? (ENT/ESC)

ESC

Press ESC to finish the editing.

Update OK?(ENT/ESC) is displayed.

<RGA80B : MOTOR>
MOTOR [Watt]
A : 756 B : 406
Z : 406 W :206

Hall Sensor
Update OK? (ENT/ESC)

ENTER

Press ENTER to save.

Check whether or not a hall sensor proper to the motor capacity has been adapted to.

<RGA80B : MOTOR>
MOTOR [Watt]
A : 756 B : 406
Z : 406 W :206

Hall Sensor
Update OK? (ENT/ESC)

ENTER

Select STORE HALL SENSOR

Changing the motor capacity and the hall sensor has been completed.

<RGA80B : MOTOR>
MOTOR [Watt]
A : 757 B : 407
Z : 407 W :407

Hall Sensor
Please Reboot!

After checking that "Please Reboot!" message is displayed, reboot the controller.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ In the contents of the name plate, enter "Motor value".

▷ In the section of PANASONIC A4 MOTOR, the displayed value is explained.
E.g.) Enter MSMD 400W → 406.

▷ In the section of PANASOINC A5 MOTOR, the displayed value is explained.
E.g.) Enter MSME 400W → 408.

Axis Motor Encoder

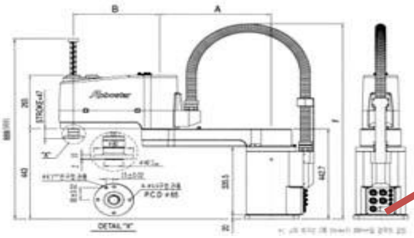
A **750W**/A4 Abs.131072
B **400W**/A4 Abs.131072
Z **400W**/A4 Abs.131072
W **200W**/A4 Abs.131072

① : When setting up the motor capacity, refer to the following table. (★Never enter a number except the following numbers.)

Motor capacity	Panasonic A4			Panasonic A5		
	Type Name	Encoder	Set-up value	Type Name	Encoder	Set-up value
50W	MSMD5A	2500pulse / 17bit	56			
100W	MSMD01	2500pulse / 17bit	106	MSME01	17bit	108
200W	MSMD02	2500pulse / 17bit	206	MSME02	17bit	208
400W	MSMD04	2500pulse / 17bit	406	MSME04	17bit	408
750W	MSMD08	2500pulse / 17bit	756	MSME08	17bit	758
1000W	MSMD10	2500pulse / 17bit	1006	MHMD10	17bit	1008

■ PARAMETER(0)-> BODY(0)-> 7:MOTOR-> 1:ENCTYPE --- motor capacity setup (absolute-related contents)

If Motor Parameter is changed, the controller performs the software rebooting or power rebooting procedure and thereafter the changed Parameter is adapted. Therefore, after Motor Parameter is changed and when it is saved, “Please Reboot!” message is displayed.



Axis	Motor	Encoder
A	**750W/A4**	**Abs.131072**
B	**400W/A4**	**Abs.131072**
Z	**400W/A4**	**Abs.131072**
W	**200W/A4**	**Abs.131072**

MTYPE --- MOTOR TYPE setup

1. Setup Sequence

Step 1. **Move to MTYPE screen**

```
<RGA80B : MOTOR(0)>
        MOTOR SETUP
1: ENCTY        2: WATT
3: MTYPE        4: HALL

                  item #
```

3 S

Open MOTOR(0) screen

Select 3: MTYPE

Step 2. **Set up MTYPE**

```
<RGA80B : MTYPE>
        MOTOR TYPE
A : LINE        B : ROTA
Z : ROTA        W :ROTA
```

ENTER

① Change setup using "ENTER"

▶ROTA ↔ LINE

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ MOTOR TYPE set-up values are as follows.

Item	Set-up value	Description
MOTOR TYPE	LINE	LINEAR MOTOR
	ROTA	ROTARY MOTOR

CAUTION

▶ LINEAR MOTOR is not now supported.

▶ After Parameter is changed, reboot the controller to adapt the changed Parameter.

Hall Sensor Setup --- Hall sensor setup

1. Setup Sequence

Step 1. Move to HALL screen

```
<RGA80B : MOTOR(0)>
          MOTOR SETUP
1: ENCTY            2: WATT
3: MTYPE            4: HALL

                         item #
```

4
L

Open MOTOR(0) screen

Select 4: HALL

Step 2. Set up HALL

```
<RGA80B : HALL SENSOR>
           HALL SENSOR
A: 200              B: 200
Z: 200              W:200
```

① Change HALL SENSOR

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ This parameter is for having a Hall sensor of a servo module be consistent according to the motor capacity.

①: For Hall sensor setup, refer to the table below. (★ Never enter a number, except the following numbers.)

Item	Motor capacity	Set-up value
HALL SENSOR	50W	50
	100W	100
	200W	200
	400W	400
	750W	750
	1000W	1000

CAUTION

▶ When setting up the motor capacity, check whether Hall sensor Parameter is updated.

▶ After Parameter is changed, reboot the controller to adapt the changed Parameter.

M-TURN --- Multi Turn Clear (Clear the multiple revolution data)

1. Setup Sequence

Step 1. **Move to M-TURN screen**

<RGA80B : MOTOR(1)>
MOTOR SETUP
1: M-TURN

item #

1 Q

Open MOTOR(1) screen

Select 1: M-TURN

Step 2. **Set up M-TURN**

<RGA80B : M-TURN>
MULTI TURN CLEAR

A B Z W

F1

F4

Using "F1" to "F4" key, select a motor performing the Multi Turn Clear

<RGA80B : M-TURN>
MULTI TURN CLEAR

Multi-Turn Clear
Motor A? (ENT/ESC)

A B Z W

ENTER

Using "ENTER" key, perform the selected motor performing Multi Turn Clear.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Multi Turn Clear is required for the following cases:

1) When the mechanical origin needs to be set up by clearing the multi-turn information of the motor (If the multi-turn information is deleted, "Encoder Counter Alarm" may occur.)
2) If "Enc Sys Down" and "Enc Over Speed" alarm occurred
(If the alarm occurs when an encoder battery is initially connected, this multi-turn data clear may be performed to clear the alarm.)
3) If the multi-turn alarm (Over-Flow) of the motor has occurred
(Revolution number counter overflow [OVFR] occurs when in +/- 4096 turns or more.)

CAUTION	▶ After performing Multi Turn Clear and when "Please Reboot!" message is displayed, be sure to reboot the controller.

1.2.2 MOTION

1. Setup item

MOTION(0)	Description	Range	Set-up value
1 : ENC	Number of pulses per revolution of encoder	500 to 140000	131072
2 : JONT	Joint motion related parameter	See the explanation	
3 : LINR	Linear motion related parameter	See the explanation	
4 : DISP	Positional coordinate sign setup	+, -	
5 : MOVE	Robot moving direction according to axis moving key (Jog Key)	+, -	
6 : SENS	Limit sensor direction setup	+, -	
7 : LIMIT	Sensor type	NO, NC	

MOTION(1)	Description	Range	Set-up value
1: INIT_V	Override velocity initial value setup	5 to 100	30
2: DEC_ZONE	JOG automatic deceleration zone setup	0 to 20	1
3: STOP_T	STOP deceleration time setup	10 to 100	100

2. Setup Sequence

Step 1. **Move to PARAMETER (0) screen**

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT



            group #
```

Open PARAMETER(0) screen

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT



            group #
```

2
R

Select Motion

Step 2. **Move to MOTION(0) screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                  item #
```

Open MOTION(0) screen

Step 3. **Move to MOTION(1) screen**

```
<XYZW : MOTION(1)>
1: INT_V        2: DEC_ZONE
3: STOP_T

                  item #
```

Pg Dn

Open MOTION(1) screen

1.2.2.1 ENC --- Number of pulse per encoder revolution setup

1. Setup Sequence

Step 1. **Move to ENC screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                item #
```

1
Q

Open MOTION(0) screen

Select 1: ENC

Step 2. **Set up ENCODER**

```
<XYZW : ENC>
     ENCODER(pulse/rev)
X: 131072      Y : 131072
Z: 131072      W: 131072
```

ENCODER setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

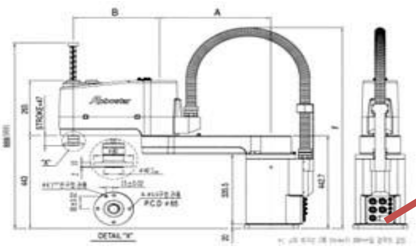
■ Comment

▷ In the contents of the name plate, enter "Encoder value".

In the section of Encoder, the displayed value is explained. E.g.) Abs. 131072P → Absolute 131072 pulses

▷ Set up the number of pulses per one turn of a motor encoder attached to each axis of the mechanism.

▷ Set up the number of pulses per revolution specified in the motor spec.



Axis	Motor	Encoder
A	750W/A4	Abs.**131072**
B	400W/A4	Abs.**131072**
Z	400W/A4	Abs.**131072**
W	200W/A4	Abs.**131072**

CAUTION

▶ Set-up value has to be changed as kinds of the motor.

E.g.) Minas A4 MOTOR → 131072 Pulse/rev

1.2.2.2 JONT --- Joint motion related parameter setup

1. Setup Sequence

Step 1. **Select JONT screen**

```
<XYZW : MOTION(0)>
1: ENC            2: JONT
3: LINR           4: DISP
5: MOVE           6: SENS
7: LIMIT

                        item #
```

2
R

Open MOTION(0) screen

Select 1: JONT

Step 2. **Set up JONT MOTION**

```
<XYZW : JONT>
      JONT MOTION: X
Mv: 3000          At: 0.80
Jv: 500           Ip: 10
It: 10            Jk: 50

X       Y       Z       W
```

F1 F4

F1: X-AXIS F2: Y-AXIS

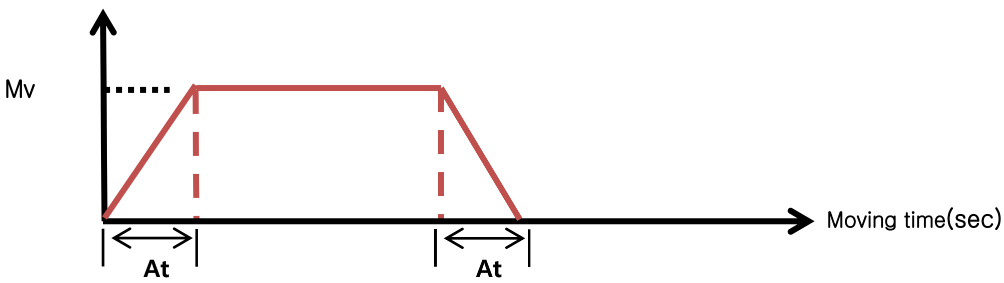
F3: Z-AXIS F4: W-AXIS

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Screen Description

Screen display	Description
<XYZW : JONT> JONT MOTION: W Mv: 3000 At: 0.80 Jv: 500 Ip: 10 It: 10 Jk: 50	● Mv : Maximum moving velocity per axis (rpm) ● At : Acceleration & deceleration time per axis (sec) ● Jv : Maximum velocity of jog based motion (rpm) ● Ip(pulse), It (mm) : Unit moving distance when in inching motion ● Jk(jerk) : Acceleration moving motion per axis



▶ Mv : Maximum moving velocity (rpm) per axis when a robot moves in Run Mode.

▷ Although the moving velocity is set up as the maximum velocity (VEL 1000) in Working Program, the robot may move under the set-up value, depending on the load or moving distance.

▷ When two or more axes move to one point at the same time, the axes have to be synchronized with each other. Thus, the velocity per axis may differ according to the moving distance per axis.

▷ Setup range : 100 to 3000

▷ Standard value : 3000

▶ At : Acceleration & deceleration time (sec) per axis

▷ Adapted to the acceleration time and the deceleration time equally.

▷ The acceleration and deceleration time may become longer than the set-up time, according to the load and mechanical conditions.

▷ Unreasonable setup of the acceleration and deceleration may cause vibration and shaking of the mechanism and may affect the life time of the mechanism.

▷ Adjust the acceleration and deceleration time within the range that does not affect the load capacity and cycle time, if possible.

▷ Setup range : 0.05 to 5

▷ Standard value : 0.2 to 0.8 (vary depending on types)

▶ Jv : Robot moving velocity (rpm) in Point Teaching when the robot is axially moved by a jog key in Teach Pendant

▷ Jog velocity includes Low (Teach Pendant indicates L), Medium(M), and High(H).

▷ Set the value to High(H).

▷ Setup range : 10 to 1000

▷ Standard value : 500

▶ It : Unit moving distance (mm, degree) in inching motion

▷ Only the unit is different from Ip set-up value. If the set-up value is changed, Ip value also changes.

▷ Inching motion includes Minimum (Teach Pendant indicates l), Medium(m), and Maximum (h).

▷ Set the value to Maximum moving distance (h).

▷ Setup range : 0 to 100

▷ Standard value : 10

▶ Ip : Unit moving distance (pulse) in inching motion

▷ Setup range : 0 to 131072

▷ Standard value : 10

▶ Jk : Acceleration moving motion per axis

▷ The higher Jerk value, the lesser the acceleration time and thus the robot moves fast.

▷ Setup range : 1 to 100

▷ Standard value : 50

1.2.2.3 LINR --- Linear motion related parameter setup

1. Setup Sequence

Step 1. **Move to LINR screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                 item #
```

3
S

Open MOTION(0) screen

Select 3: LINR

Step 2. **Set up LINEAR MOTION**

```
<XYZW : LINR>
        LINER MOTION
Mv: 1000        At: 0.20
Jv: 500         Ia: 10
Jk: 50
```

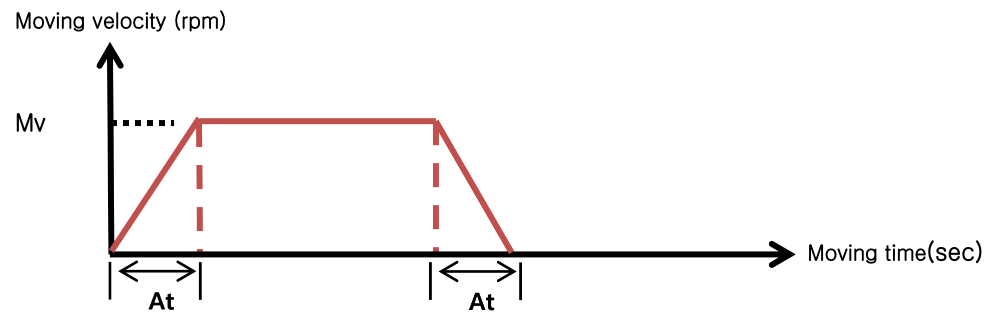
Set up LINEAR MOTION

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Screen & Description

Screen display	Description
LINER MOTION Mv :1000 At : 0.3 Jv : 300 Ia : 10 Jk : 50	Mv : Maximum interpolated motion velocity (mm/s) At : Interpolated motion acceleration/deceleration time (sec) Jv : Maximum interpolated motion velocity when in Jog motion (mm/s) Ia : Unit moving distance when in inching motion (mm) Jk(jerk) : Acceleration moving motion



▶ Mv : Maximum velocity (mm/s) when a robot does interpolated motion in Run Mode.

▷ Although the moving velocity is set up as the maximum velocity (VEL 1000) in Working Program, the robot may move under the set-up value, depending on the load or moving trajectory

▷ Sometimes, the set-up value may not be followed because of the abrupt trajectory change, load, motor gain, etc., when the continuous interpolated motion is performed by using FOS or FIX commands.

▷ Adjust the moving velocity, load, and gain before operating the robot.

▷ Setup range : 0 to 1000

▷ Standard value : 1000

▶ At : Acceleration & deceleration time (s) of the composite axis

▷ Adapted to the acceleration time and the deceleration time equally.

▷ The acceleration and deceleration time may become longer than the set-up time, according to the load and mechanical conditions.

▷ Unreasonable setup of the acceleration and deceleration may cause vibration and shaking of the mechanism and may affect the life time of the mechanism.

▷ Adjust the acceleration and deceleration time within the range that does not affect the load capacity and cycle time, if possible.

▷ Setup range : 0.05 to 5

▷ Standard value : 0.5

▶ Jv : Robot moving velocity (mm/sec) in Point Teaching when the robot is performed in interpolated motion by a jog key in Teach Pendant

▷ Jog velocity includes Low (Teach Pendant indicates L), Medium(M), and High(H).

▷ Set the value to High(H).

▷ When the mechanism is SCARA robot :
When, in Point Teaching, Jog key is used after converting to XY coordinate system and if at this time the velocity has been set up to High (H), the robot may move at the speed exceeding the anticipated velocity, depending on its position. Be careful of this. .

▷ Setup range : 10 to 1000

▷ Standard value : 100

▶ Ia : Unit moving distance (mm) in inching motion (interpolated motion)

▷ Inching motion includes Minimum (Teach Pendant indicates l), Medium(m), and Maximum (h).

▷ Set the value to Maximum moving distance (h).

▷ Setup range : 0 to 500

▷ Standard value : 10

▶ Jk : Acceleration moving motion

▷ The higher Jerk value, the lesser the acceleration time and thus the robot moves fast.

▷ Setup range : 1 to 100

▷ Standard value : 50

1.2.2.4 DISP --- Displayed positional coordinate sign setup

1. Setup Sequence

Step 1. **Move to DISP screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                item #
```

4 / L

Open MOTION(0) screen

Select 4: DISP

Step 2. **Set up DISP DIRECTION**

```
<XYZW : DISP>
    DISPLAY DIRECTION
X: +            Y : +
Z: +            W: -
```

ENTER

"ENTER" to change.

- \+ → -
- \- → +

```
<XYZW : DISP>
    DISPLAY DIRECTION
X: -            Y : +
Z: +            W: -
```

Change DISPLAY DIRECTION

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Standard value has been set when in discharge from the factory.

▷ If the value is set up other than the standard value, W-axis interpolation is not performed in the operation of interpolation or palletizing.

▶ **Change in X-AXIS sign**

Before change

```
F:TEST P:0   NW   B   L
X: 10.0          Y: 10.0
Z: 10.0          W:10.0
```

⇨

After change

```
F:TEST P:0   NW   B   L
X: -10.0         Y: 10.0
Z: 10.0          W:10.0
```

1.2.2.5 MOVE --- Axis moving key(Jog)-based robot moving direction setup

1. Setup Sequence

Step 1. **Move to MOVE screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                item #
```

5 M

Open MOTION(0) screen

Select 5: MOVE

Step 2. **Set up MOVING KEY**

```
<XYZW : MOVE>
   MOVING KEY DIRECTION
X: +            Y : +
Z: +            W: -
```

ENTER

"ENTER" to change.

- \+ → -
- \- → +

```
<XYZW : MOVE>
   MOVING KEY DIRECTION
X: -            Y : +
Z: +            W: -
```

Change MOVING KEY DIRECTION

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Checking after change

- The robot moving direction and the arrow mark direction of the axis moving key(Jog) are to be aligned for definite and convenient teaching during the actual point teaching.
- If the robot moves in the inverse direction of the Jog key, change the sign according to the above procedure.

CAUTION	▶ If setup is different from the standard set-up value, Limit Sensor cannot be detected during Jog key based motion of a robot.

1.2.2.6 SENS --- Each axis' limit sensor(Limit Sensor) direction setup

1. Setup Sequence

Step 1. **Move to SENS screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                item #
```

6
N

Open MOTION(0) screen

Select 6: SENS

Step 2. **Set up SENSOR DIRECTION**

```
<XYZW : SENS>
     SENSOR DIRECITON
X: +            Y : +
Z: +            W: -
```

ENTER

"ENTER" to change.

- \+ → -
- \- → +

```
<XYZW : SENS>
     SENSOR DIRECITON
X: -            Y : +
Z: +            W: -
```

Change X-AXIS Sensor Direction

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Checking after the sign change

1) Change the direction of a limit sensor installed in the robot mechanism.
2) Set up the value of "DISP" and "MOVE", and move the robot to Limit Sensor by using JOG KEY.
3) After checking the Limit Error,
   ① When a moving direction Jog Key is pressed, the mechanism must not move any further.
   ② When a reverse direction Jog Key is pressed, the robot has to be able to escape from the limit sensor.

▷ If the JOG Key does not well operate, check the followings.

1) Whether or not wirings of the robot mechanism limit sensor (+Limit, -Limit) are inverted.
2) When the attachment direction of the rectangular mechanism motor is of parallel type.
3) Set-up values of "DISP" and "MOVE" are set up to the values other than the standard value.

Important

▶ If it is impossible to escape from Limit by using Jog key, toggle the current status of "6:SENS" for the relevant axis by using "ENTER" Key, and then, set up the parameters to be able to escape from Limit Sensor.

1.2.2.7 LIMIT --- Each axis' limit sensor(Limit Sensor) operation setup

1. Setup Sequence

Step 1. **Move to LIMIT screen**

```
<XYZW : MOTION(0)>
1: ENC          2: JONT
3: LINR         4: DISP
5: MOVE         6: SENS
7: LIMIT

                item #
```

7 G

Open MOTION(0) screen

Select 7: LIMIT

Step 2. **Change LIMIT SENSOR**

```
<XYZW : LIMIT>
      LIMIT SERSOR TYPE
X: NC           Y : NC
Z: NC           W: NC
```

ENTER

"ENTER" to change Sensor Type

- NC → NO
- NO → NC

Step 3. **Store the change**

```
<XYZW : LIMIT>
      LIMIT SERSOR TYPE
X: NO           Y : NC
Z: NC           W: NC

Update OK? (ENT/ESC)
```

ESC

"ESC" to select the change

```
<XYZW : LIMIT>
      LIMIT SERSOR TYPE
X: NO           Y : NC
Z: NC           W: NC

Update OK? (ENT/ESC)
```

ENTER

Storing completed

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

Important

▶ LIMIT sensor is generally used as NC type, and ORG sensor as NO type.
For the products by Robostar, it is recommended to use NC type for LIMIT sensor.

1.2.2.8 INIT_V --- initial velocity setup

1. Setup Sequence

Step 1. **Move to INIT_V screen**

```
<XYZW : MOTION(1)>
1: INIT_V            2: DEC_ZONE
3: STOP_T



                     item #
```

1
Q

Open MOTION(1) screen

Select 1: INIT_V

Step 2. **Set up INIT_VEL**

```
<XYZW : INIT VEL>
        INIT VEL(%)
VEL: 30
```

Set up INITIAL VELOCITY

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ In JOB if the velocity value has not been specified by VEL command:

- The value set-up in INIT VEL Parameter operates.

```
*0000   MAIN
 0001   JMOV P0
 0002   JMOV P1
 0003   JMOV P2
 0004   EOP
```

If VEL was not set, the robot is operated by TINIT VEL set-up value.

▷ In JOB if the velocity value has been specified by VEL command:

- The initial driving velocity operates at the percentage ratio of VEL set-up value.

```
*0000   MAIN
 0001   VEL 1000
 0002   JMOV P0
 0003   JMOV P1
 0004   JMOV P2
 0004   EOP
```

VEL 1000 Set-up value is the same as 100 of INIT_VEL.

1.2.2.9 DEC_ZONE --- JOG deceleration zone setup

1. Setup Sequence

Step 1. **Move to DEC_ZONE screen**

```
<XYZW : MOTION(1)>
1: INIT_V           2: DEC_ZONE
3: STOP_T

                        item #
```

2
R

Open MOTION(1) screen

Select 2: DEC_ZONE

Step 2. **Set up DEC_ZONE**

```
<XYZW : DEC ZONE>
      JOG DEC ZONE (%)
ZONE:   1
```

JOG deceleration zone setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ This sets up the deceleration start range in the unit of % ratio, within the Software Limit range when in jog operation.
▷ Setup unit : %
▷ Setup range: 0 to 20

1.2.2.10 STOP_T --- STOP deceleration time setup

1. Setup Sequence

Step 1. **Move to STOP_T screen**

```
<XYZW : MOTION(1)>
1: INIT_V              2: DEC_ZONE
3: STOP_T

                          item #
```

3
S

Open MOTION(1) screen

Select 3: STOP_T

Step 2. **Set up STOP_T**

```
<XYZW : STOP TIME>
         STOP TIME (%)
Rate:     100
```

STOP TIME setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ When performing PARA(0) -> 2.MOTION -> Stop motion performed by inputting "STOP" key on a basis of At set-up value, the stop time can be reduced by changing the set-up value.

▷ Setup range 10 to 100

Item	Set-up value	Description
RATE	10	10% of At set-up value
	100	At set-up value

1.2.3 GAIN

1. Items to be set up

N1 controller has two gains: First gain(PVG_1st) and Second gain(PVG_2nd), where the gain change is possible by using Parameter setup and Conditions.

GAIN	Description	Range	Set-up value
1 : PVG_1st	Position, Velocity, First gain	See the explanation	
2 : PVG_2nd	Position, Velocity, Second gain	See the explanation	
3 : FILTER	Acceleration/Deceleration filter	0 to 7	

▶ Gain value:
- Pay attention to change of the set-up values, because, unlike other parameters, the mechanism vibrates or makes noise even when there is minute difference of the set-up value.
- Do not change the gain values set at the time of shipment since they have been setup optimally for the robot mechanism.

2. Setup Sequence

Step 1. **Move to GAIN screen**

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT



                 group #
```

Open PARAMETER(0) screen

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT



                 group #
```

3
S

Select “3: GAIN”

Step 2. **First & Second GAIN Select screen**

```
<XYZW : GAIN>
1: PVG_1st       2: PVG_2nd
3: FILTER



                 item #
```

First & Second GAIN Select screen

1.2.3.1 PVG_1st--- First Position Velocity Gain setup

1. Setup Sequence

Step 1. **Move to PVG_1st**

```
<XYZW : GAIN>
1: PVG_1st          2: PVG_2nd
3: FILTER



                    item #
```

1 Q

Open GAIN screen

Select 1: PVG_1st

Step 2. **PVG_1st setup screen**

```
<XYZW : PVG>
POS/VEL   GAIN: X
Kpp:60          Kpf: 0
Kvf : 0         Kvi: 16
Kvp: 40         IR : 1
Tcf : 300       Vft: 200

X       Y       Z       W
```

F1

X-AXIS PVG_1st setup screen

F1: Store X-AXIS   F2: Store Y-AXIS

F3: Store Z-AXIS   F4: Store W-AXIS

```
<XYZW : PVG>
POS/VEL   GAIN: W
Kpp:60          Kpf: 0
Kvf : 0         Kvi: 16
Kvp: 40         IR : 1
Tcf : 300       Vft: 200

X       Y       Z       W
```

F4

W-AXIS PVG_1st setup screen

Stored when the axis is moved after the set-up value has been changed.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Kpp : Position Proportional Gain

- Description : Velocity determining parameter for reducing positional deviation
- Unit : 1/s
- Setup range : 0 to 2000
- Setup step : 5
- If Kpp value is increased: The velocity that reduces positional deviation becomes faster, but if too much value the vibration arises.
- If Kpp value is decreased: The velocity that reduces positional deviation becomes slower.
- Standard value : 70 (at no load)

▷ Kpf : Velocity Feedforward Gain

- Description : Parameter that can improve the response-ability by performing predictive control
- Unit : 0.1%
- Setup range : 0 to 2000
- Setup step : 100
- If Kff value is increased : The positional deviation becomes smaller and so the response-ability is improved, but, if it too much increased, the over/undershoot arises.
- Standard value : 0 (at no load)

▷ Kvf : Current Controller Feedfordward Gain

- Not supported
- Setup range : 0.0 to 0.0
- Standard value : 0.0

▷ Kvi : Velocity Integral Gain

- Description : Parameter that determines the velocity which reduces the positional deviation when in stop.
- Unit : ms
- Setup range : 0 to 1000
- Setup step : 5
- Kvi value is increased: The velocity that reduces the positional deviation becomes faster, but the vibration occurs.
- Kvi value is decreased: The vibration becomes less, but the velocity that reduces the positional deviation becomes slower.
- Standard value : 16 (at no load)

-> The smaller the integral time constant, the better the response-ability. However, if 1000 is input, the integral effect disappears.

▷ Kvp : Velocity Proportional Gain

- Description : Parameter that determines the velocity response-ability
- Unit : Hz
- Setup range : 0 to 2000
- Setup step : 5
- Kvp value is increased: Driving torque and dynamic characteristic of a motor are improved but vibration and noise occur.
- Kvp value is decreased: Driving torque of a motor becomes less and the dynamic characteristic becomes slower (staggering).
- Standard value : 40 (at no load)

▷ IR : Load Inertia Ratio

- Description : Parameter that determines the ratio of load inertia to motor inertia
- Unit : **Inertia Ratio : (Load Inertia / Motor Inertia) * 100 [%]**
- Setup range : 0 to 2000
- Setup step : 50
- IR value is increased: The motor torque increases, but vibration and noise occur.
- IR value is decreased: The motor torque becomes lower
- Standard value : 0 (at no load)

-> When the exact load inertia ratio is input, the velocity and the positional gain are normally adapted. However, if the inertia ratio is greater, the velocity and the positional gain are more greatly adapted, and if the inertia ratio is smaller, the velocity and the positional gain are less adapted.

▷ Tcf : Torque Command Filter

- Description : Parameter that results in suppression of the resonance occurring in 200Hz or higher
- Unit : Hz
- Setup range : 0 to 3000
- Setup step : 50
- Tcf value is increased: the greater the value, the better the response-ability, when there are no vibration and noise.

- Tcf value is decreased: The resonance and the noise of the machine decreases, but the response-ability decreases.
- Standard value : 0 (at no load)

-> If 3000 is input, the effect disappears.

▷ Vft: Velocity Feedforward Filter

- Description : Parameter that determines a filter when the over/undershoot occurs in the case that the velocity feedforward value is too great.
- Unit : Hz
- Setup range : 0 to 1000
- Setup step : 50
- Vft value is increased: The over/undershoot occurrence, when the velocity feedforward value is set up, can be reduced.
- Standard value : 0 (at no load)

-> If 1000 is input, this effect disappears. If too little value is input, the velocity forward gain is not adapted.

■ Basic gain set-up value for SCARA Robot

	A(X)	B(Y)	Z(vertical axis)	W(rotating axis)
Item	750W	400W	400W	200W
Kpp	40.0	40.0	70.0	70.0
Kpf	800.0	800.0	800.0	800.0
Kvf	0.0	0.0	0.0	0.0
Kvi	20.0	20.0	16.0	16.0
Kvp	40.0	40.0	40.0	40.0
IR	1500.0	1200.0	1500.0	1000.0
Tcf	300.0	300.0	300.0	300.0
Vft	200.0	200.0	200.0	200.0

■ Basic gain set-up value for Cartesian Robot

	X	Y	Z	W
Item	750W	400W	400W	200W
Kpp	60.0	60.0	60.0	60.0
Kpf	0.0	0.0	0.0	0.0
Kvf	0.0	0.0	0.0	0.0
Kvi	25.0	25.0	25.0	25.0
Kvp	40.0	40.0	40.0	40.0
IR	300.0	300.0	300.0	300.0
Tcf	300.0	300.0	300.0	300.0
Vft	200.0	200.0	200.0	200.0

1.2.3.2 PVG_2nd--- Second Position Velocity Gain setup

1. Setup Sequence

Step 1. **Move to PVG_2nd**

```
<XYZW : GAIN>
1: PVG_1st          2: PVG_2nd
3: FILTER



                  item #
```

2
R

Open GAIN screen

Select 2: PVG_2nd

Step 2. **PVG_2nd setup screen**

```
<XYZW :PVG 2nd>
POS/VEL   GAIN: X
Kpp2: 50         Kvp2: 40
Kvi2 : 30        Tcf2 : 500
Act:: 0          Mode : 0
Dtim : 300       Leve: 200
Hyst: 300        Time: 10
X       Y        Z       W
```

F1

X-AXIS PVG_2nd setup screen

F1: Store X-AXIS F2: Store Y-AXIS

F3: Store Z-AXIS F4: Store W-AXIS

```
<XYZW :PVG 2nd>
POS/VEL   GAIN: X
Kpp2: 50         Kvp2: 40
Kvi2 : 30        Tcf2 : 500
Act:: 0          Mode : 0
Dtim : 300       Leve: 200
Hyst: 300        Time: 10
X       Y        Z       W
```

F4

W-AXIS PVG_2nd setup screen

Stored when the axis is moved after the set-up value has been changed.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▶ <u>How to set up the second gain</u>

▷ For the position loop gain and the velocity loop gain, the second gain is to be set more greatly than the first gain.

▷ Velocity loop time constant is to be set to about 300.

▷ For the velocity detection filter and the torque filter, the first gain and the second gain are to be set equally.

▷ Act: Second Gain Action Setup

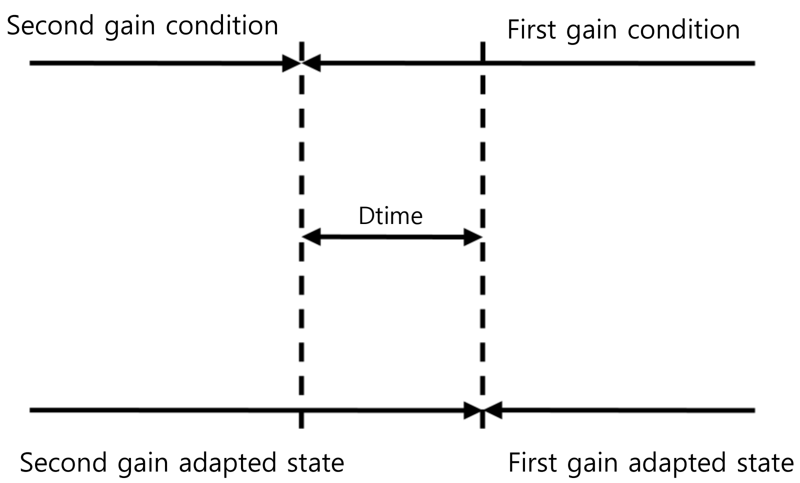
- Description : Parameter that sets up whether or not the first gain and the second gain are switched.
- Setup range : 0 to 1
- If Act value is 0 : Only the first gain is used.
- If Act value is 1 : 1/Second gain switch function is used.
- Standard value : 0

▷ Mode: Gain Switch Mode

- Description : Parameter that determines the gain switch mode on a basis of the amount of positional deviation
- Setup range : 0
- Standard value : 0

▷ Dtime: Gain Switch Delay Time)

- Description : Parameter that determines the delay time when switching from the second gain to the first gain
- Unit : 160us
- Setup range : 0 to 10000
- If Dtime value is increased: the delay time for switching to the first gain becomes longer.
- If Dtime value is decreased: the delay time for switching to the first gain becomes shorter.
- Standard value : 300

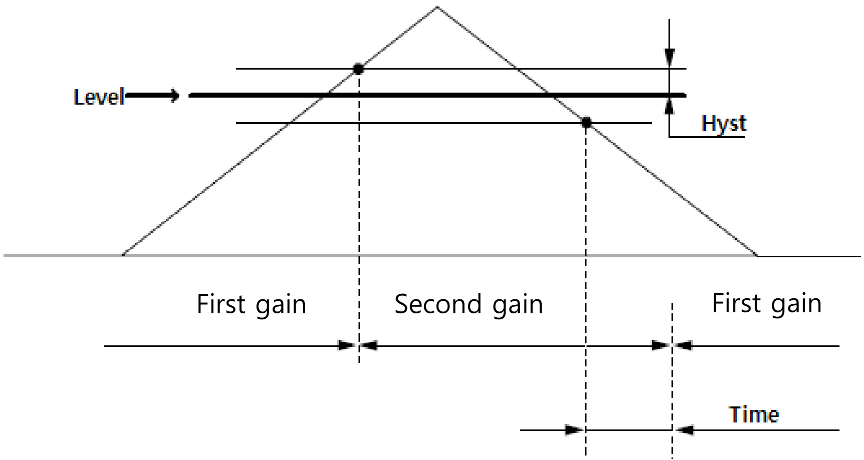


▷ Level: Gain Switch Level

- Description : Parameter that determines the switch level of the first gain and the second gain
- Unit : -
- Setup range : 0 to 10000
- If the level value is increased: the gain switch level increases.
- If the level value is decreased: the gain switch level decreases.
- Standard value : 200

▷ Hyst: Gain Switch Hysteresis

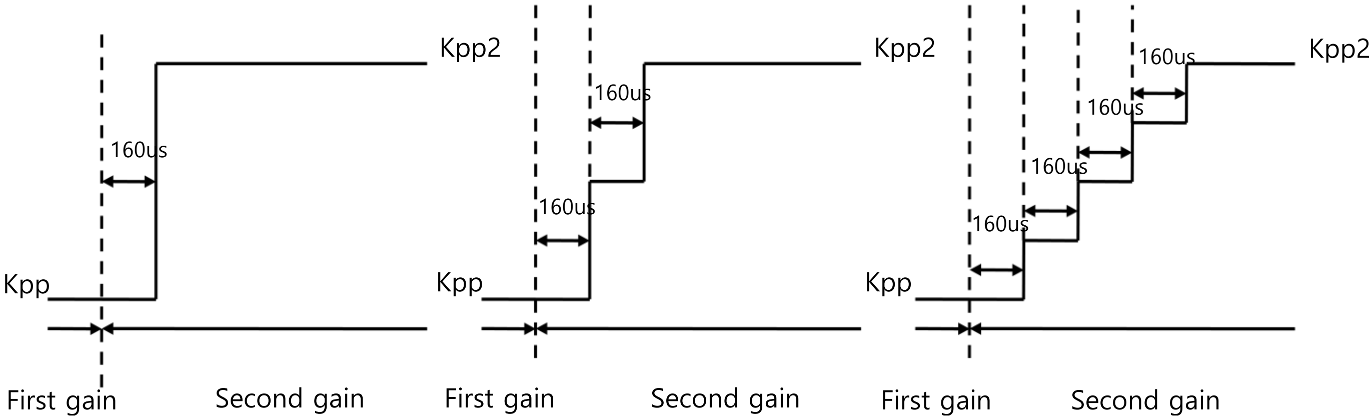
- Description : Parameter that sets up the upper and lower width of the switch level on a basis of the switch level determined by the gain switch level parameter
- Unit : -
- Setup range : 0 to 10000
- If Hyst value is increased: the width increases on a basis of the gain switch level
- If Hyst value is decreased: the width decreases on a basis of the gain switch level
- Standard value : 300



▷ Time: Gain Switch Time

- Description : In the situation that the second gain switch function is used, since the temporary change of the position loop gain may shock the mechanism, the gain is stepwise changed to relieve the shock.
- Unit : (Set-up value + 1) * 160us

  If the parameter set-up value is 0, the switch time is minimum 160us.
- Setup range : 0 to 10000
- If the time value is increased: Since the gain switching steps increase, the shock can be relieved but the gain switching is delayed.
- If the time value is decreased: Since the gain switching steps decrease, the gain switching becomes faster but the degree of shock relieving becomes lower.
- Standard value : 10



1.2.3.3 FILTER--- Position Command Smoothing filter setup

1. Setup Sequence

Step 1. **Move to FILTER**

```
<XYZW : GAIN>
1: PVG_1st          2: PVG_2nd
3: FILTER



                    item #
```

3
S

Open GAIN screen

Select 3: FILTER

Step 2. **Set up FILTER**

```
<XYZW : FILTER>
     SMOOTHING FILTER
X : 0               Y : 0
Z : 0               W: 0
```

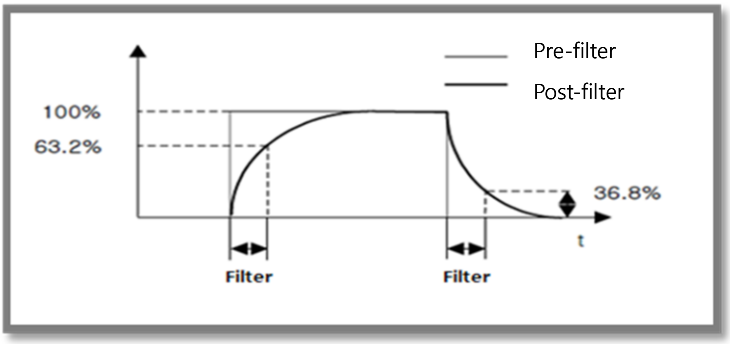
Set up SMOOTHING FILTER value

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ This is used to prevent that a motor stepwise moves when the acceleration and deceleration of the first command filter inserted to the Motor Command Pulse is exceedingly great.

▷ The greater the filter value, the greater the time constant value.



Set-up value	Time constant
[0]	No filtering function
1	Time constant little
to	↓
7	Time constant great

1.2.4 PROTECT

1. Items to be set up

PROTECT(0)	Description	Range	Set-up value
1 : SET	Permissible capacity of the mechanical shaking at the state of In Position	500 to 9999	2000
2 : FOW	Tolerance of the mechanism following in response to the controller command.	1 to 9999	1024
3 : OVS	Maximum velocity permissible to the motor of the mechanism	500 to 6000	3500
4 : OVT	Duration time of the load permissible to the mechanism	10 to 110	25
5 : OVA	Acceleration permissible to the mechanism	500 to 9999	5000
6 : OVL	Load permissible to the mechanism	10 to 110	110
7 : IPA	In Position amount	1 to 999	100
8 : IPE	In Position time	1 to 999	200

PROTECT(1)	Description	Range	Set-up value
1 : TRQ	Torque limit value	1 to 300	300
2 : TOL	Duration time of the torque limit value	1 to 2000	20
3 : OND	Delay time for Brake On signal	0 to 25	10
4 : DB	Alarm status	ENB/DIS	ENB
	Normal status	ENB/DIS	DIS

2. Screen Configuration

Step 1. **Move to PROTECT(0) screen**

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT

                 group #
```

4
L

Open PARAMETER(0) screen

Select 4: PROTECT

Step 2. **PROTECT(0) screen**

```
<XYZW : PROTECT(0)>
1: SET           2: FOW
3: OVS           4: OVT
5: OVA           6: OVL
7: IPA           8: IPE

                 item #
```

Pg Up

PROTECT(0) screen

Step 3. **PROTECT(1) screen**

```
<XYZW : PROTECT(1)>
1: TRQ           2: TOL
3: OND           4: DB

                 item #
```

Pg Dn

PROTECT(1) screen

1.2.4.1 SET --- Settling Error setup

1. Setup Sequence

Step 1. **Move to SET screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                 item #
```

1 / Q

Open PROTECT (0) screen

Select 1: SET

Step 2. **Set up SET**

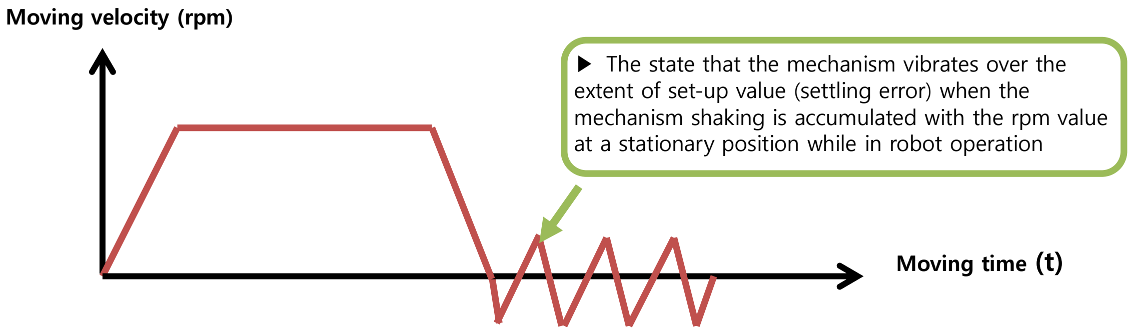
```
<XYZW : SET>
       SETTLING (rpm)
X: 2000         Y : 2000
Z: 2000         W: 2000
```

① SETTLING ERROR setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description



Important

■ Items to be checked

1) Check the mechanism is correctly assembled.
2) Check that the load is within the proper load capacity.
3) Check that the gain set-up value for each axis is correct.

1.2.4.2 FOW --- Following Error setup

1. Setup Sequence

Step 1. Move to FOW screen

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                  item #
```

2
R

Open PROTECT (0) screen

Select 2: FOW

Step 2. Set up FOW

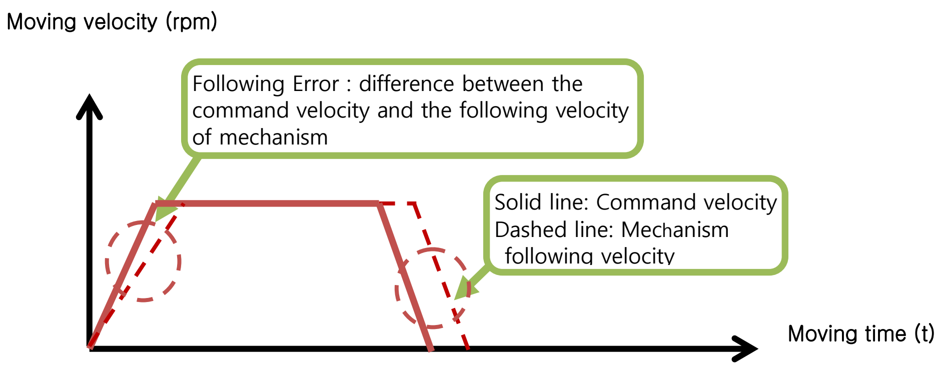
```
<XYZW : FOW>
  FOLLOWING (256pulse)
X: 1024         Y : 1024
Z: 1024         W: 1024
```

① FOLLOWING ERROR setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description



Important

■ Items to be checked

1) Check that the load and the gain of each axis are correct.

2) Adjust the maximum velocity(Mv) and the acelleraton and deceleration time (At) value.

1.2.4.3 OVS --- Over Speed Error setup

1. Setup Sequence

Step 1. **Move to OVS screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                     item #
```

3
S

Open PROTECT (0) screen

Select 3: OVS

Step 2. **Set up OVS**

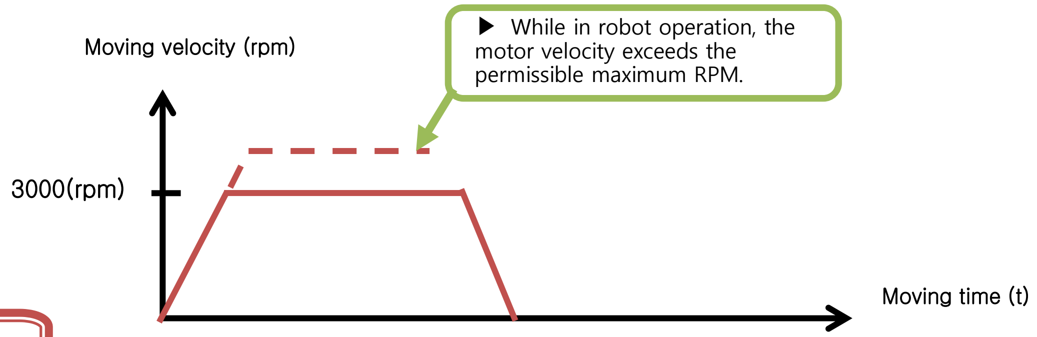
```
<XYZW : OVS>
      OVER SPEED (rpm)
X: 3500         Y : 3500
Z: 3500         W: 3500
```

① OVER SPEED ERROR setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description



Important

■ Items to be checked

1) Check the abnormality of the motor encoder.
2) Check the mechanism interconnection cable.
3) If, when in continuous interpolation a robot moves to the next position, abrupt direction change occurs, correct the point so that the moving angle becomes smoother.

1.2.4.4 OVT --- Overload Time setup

1. Setup Sequence

Step 1. **Move to OVT screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                item #
```

4
L

Open PROTECT (0) screen

Select 4: OVT

Step 2. **Set up OVS**

```
<XYZW : OVT>
      OVER LOAD (100ms)
X: 25           Y : 25
Z: 25           W: 25
```

① OVERLOAD TIME setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

① : Description

▷ Set up the overload permission time when in operation at maximum torque.

- minimum value: 10
- maximum value: 110

CAUTION

▶ After changing the parameter, reboot the controller to adapt the changed parameter.

1.2.4.5 OVA --- Over Accel Error setup

1. Setup Sequence

Step 1. **Move to OVA screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                item #
```

5
M

Open PROTECT (0) screen

Select 5: OVA

Step 2. **Set up OVA**

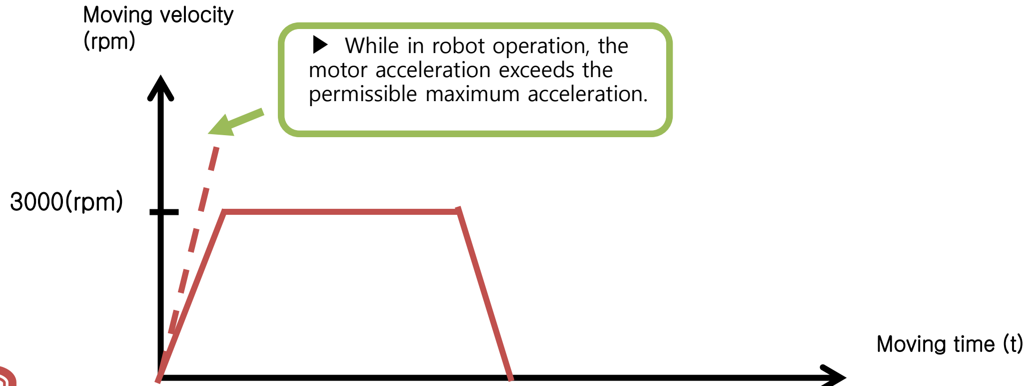
```
<XYZW : OVA>
      OVER ACCEL (msec)
X: 5000         Y : 5000
Z: 5000         W: 5000
```

① OVER ACCEL ERROR setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

① : Description



Important

■ Items to be checked when OVER ACCEL ERROR occurs

1) Check the abnormality of the motor encoder.
2) Check the mechanism interconnection cable.
3) If, when in continuous interpolation a robot moves to the next position, abrupt direction change occurs, correct the point so that the moving angle becomes smoother.

1.2.4.6 OVL --- Over Load Error setup

1. Setup Sequence

Step 1. **Move to OVL screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                  item #
```

6
N

Open PROTECT (0) screen

Select 6: OVL

Step 2. **Set up OVL**

```
<XYZW : OVS>
     OVER LOAD RATE(%)
X: 110          Y : 110
Z: 110          W: 110
```

① OVER LOAD RATE setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description

- Permissible load value setup

Important

■ Items to be checked when OVER LOAD ALARM occurs

1) Too overly setup may damage parts in the controller.
2) In order to generate the alarm sensitively responding to the exterior shock, lower the set-up value under the standard value.

CAUTION

▶ After changing the parameter, reboot the controller to adapt the changed parameter.

1.2.4.7 IPA --- Inposition Amount setup

1. Setup Sequence

Step 1. **Move to IPA screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                 item #
```

7
G

Open PROTECT (0) screen

Select 7: IPA

Step 2. **Set up IPA**

```
<XYZW : IPA>
        INPOS (1pulse)
X: 100          Y : 100
Z: 100          W: 100
```

① IN POSITION AMOUNT setup screen

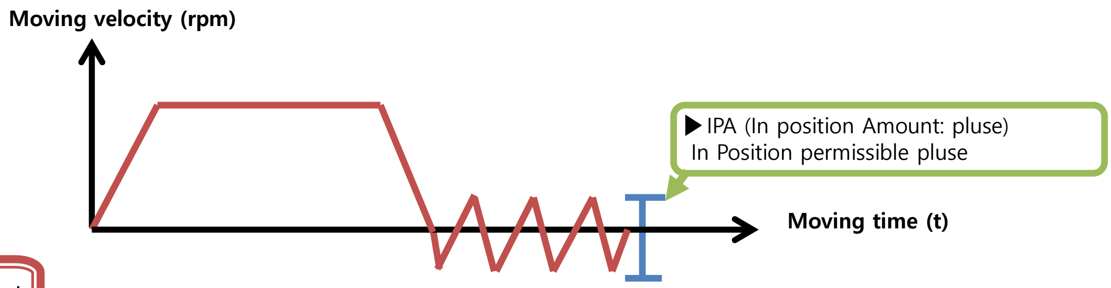
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description

▷ INPOSITION permissible pulse amount

▷ Inposition Error occurs when the mechanism shakes with both over-rated IPE and IPA value.



Important

▶ Items to be checked when Inposition Error occurs.

1) Check the mechanism is correctly assembled.
2) Check the load is within the proper load capacity range.
3) If IPA is set to "0", do not check Inposition while a job is being executed. At this time, use INPOS command in JOB.

1.2.4.8 IPE --- Inposition Error setup

1. Setup Sequence

Step 1. **Move to IPE screen**

```
<XYZW : PROTECT(0)>
1: SET          2: FOW
3: OVS          4: OVT
5: OVA          6: OVL
7: IPA          8: IPE

                  item #
```

8
H

Open PROTECT (0) screen

Select 8: IPE

Step 2. **Set up IPE**

```
<XYZW : IPE>
      INPOS ERROR (8ms)
X: 200          Y : 200
Z: 200          W: 200
```

① INPOSITION ERROR setup screen

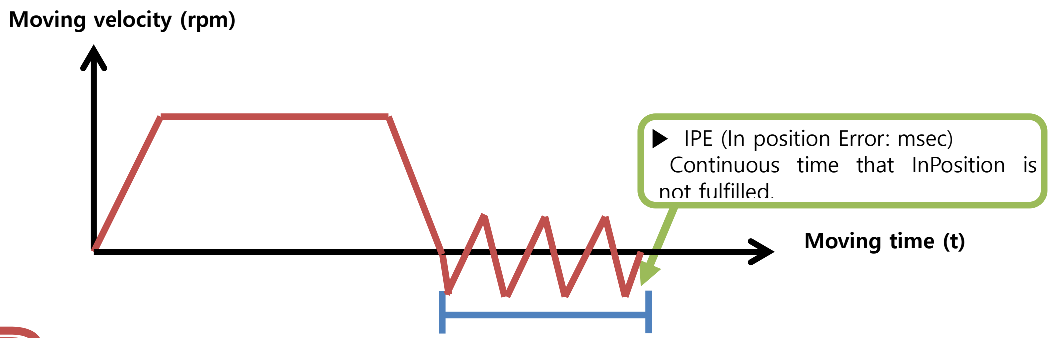
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description

▷ Time for checking INPOSITION

▷ Inposition Error occurs when the mechanism shakes with both over-rated IPE and IPA value.



Important

Items to be checked when Inposition Error occurs
1) Check the mechanism is correctly assembled.
2) Check the load is within the proper load capacity range.
3) If IPE is set to "0", do not check Inposition while a job is being executed. At this time, use INPOS command in JOB.

1.2.4.9 TRQ --- Torque Limit setup

1. Setup Sequence

Step 1. **Move to TRQ screen**

```
<XYZW : PROTECT(1)>
1: TRQ            2: TOL
3: OND            4: DB



                  item #
```

1
Q

Open PROTECT (1) screen

Select 1: TRQ

Step 2. **Set up TRQ**

```
<XYZW : TRQ>
        TRQ LIMIT [%]
X: 300            Y : 300
Z: 300            W: 300
```

① TRQ LIMIT setup screen

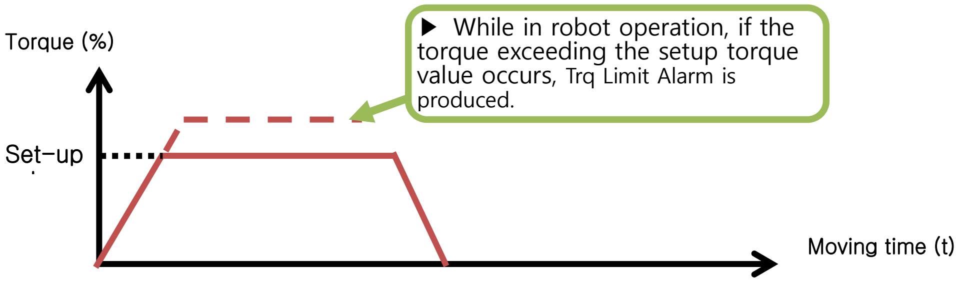
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description

▷ Parameter that limits the maximum torque of a motor

▷ Unit of TRQ is %, and the range is 1 to 300



Important

▶ The torque alarm may be produced by using TRQ JOB command.

▶ For sensitive response to the exterior shock, lower the TRQ value.

1.2.4.10 TOL --- Conversion value for change of Torque Limit interval

1. Setup Sequence

Step 1. **Move to TOL screen**

```
<XYZW : PROTECT(1)>
1: TRQ              2: TOL
3: OND              4: DB



              item #
```

2
R

Open PROTECT (1) screen

Select 2: TOL

Step 2. **Set up TOL**

```
<XYZW : TOL>
     TRQ LIMIT TOLA [8ms]
X: 20               Y : 20
Z: 20               W: 20
```

① TRQ LIMIT TOLERANCE setup screen

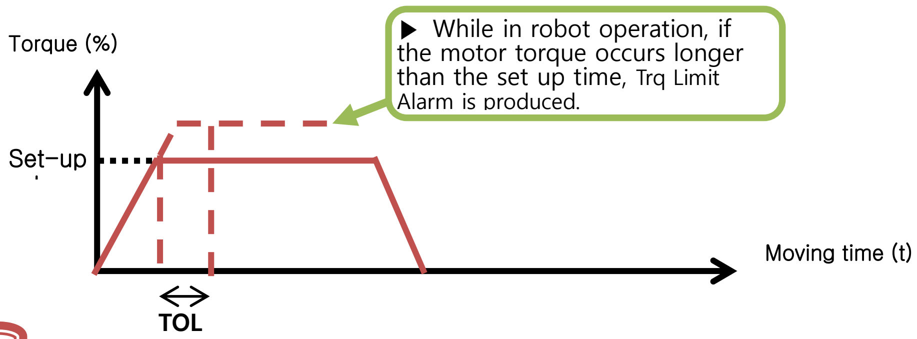
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

① : Description

▷ Reference time to produce the alarm when TRQ command of JOB command is used

▷ Unit of TOL is 10ms, and the range is 1 to 2000



Important

▶ For sensitive response to the exterior shock, lower the TRQ value.

1.2.4.11 OND --- Brake On Delay setup

1. Setup Sequence

Step 1. **Move to OND screen**

```
<XYZW : PROTECT(1)>
1: TRQ            2: TOL
3: OND            4: DB

                item #
```

3
S

Open PROTECT (1) screen

Select 3: OND

Step 2. **Set up OND**

```
<XYZW : OND>
     BK ON DELAY (8ms)
X: 10             Y : 10
Z: 10             W: 10
```

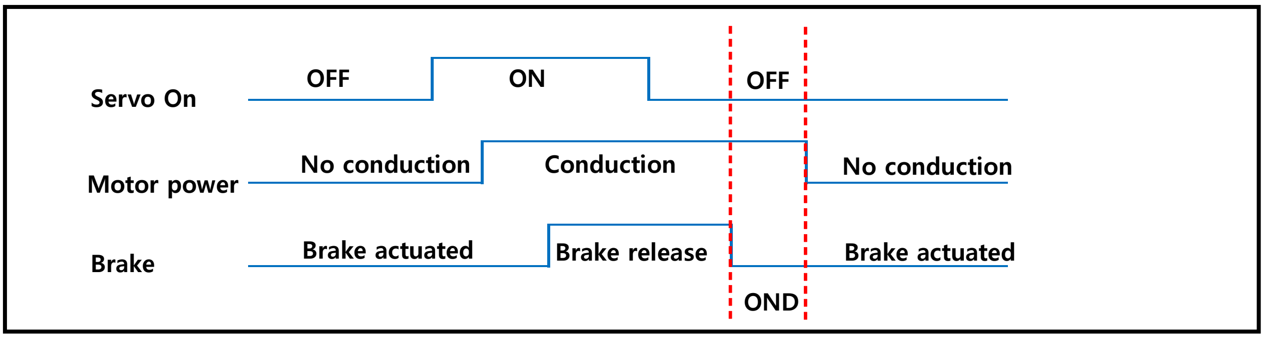
① BRAKE ON DELAY setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Because of mechanical element, Brake ON/OFF signal is delayed than Servo ON/OFF signal. Therefore, the mechanical parts of the robot may be displaced and trailed. To compensate for that, conduction time at Servo OFF point may be set up as OND.

▷ Unit of Brake ON time delay : msec



Important

▶ Too overly Brake ON/OFF Delay may damage a motor and a brake because the brake confines the motor at Inposition.

▶ When in Servo ON/OFF of the robot mechanism whose load acts in the gravity direction, Brake ON Delay can be used for compensation.

1.2.4.12 DB --- Dynamic Brake setup

1. Setup Sequence

Step 1. **Move to DB screen**

```
<XYZW : PROTECT(1)>
1: TRQ            2: TOL
3: OND            4: DB



             item #
```

4
L

Open PROTECT (1) screen

Select 4: DB

Step 2. **MOTOR DB screen**

```
<XYZW : MOTOR DB>
1: ALARM          2: NOMAL



             item #
```

MOTOR DB setup screen

Step 3-1. **ALARM STATUS**

```
<XYZW : DYNAMIC BK>
        ALARM STATUS
X: ENB            Y : ENB
Z: ENB            W: ENB
```

1
Q

① Select "1: ALARM STATUS"

Step 3-2. **NORMAL STATUS**

```
<XYZW : DYNAMIC BK>
       NORMAL STATUS
X: DIS            Y : DIS
Z: DIS            W: DIS
```

2
R

② Select "2: NORMAL STATUS"

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Use or non-use of DB(Dynamic Brake) can be individually set up for the alarm state and for the normal state.

▷ ENB : Dynamic Brake is used.

▷ DIS : Dynamic Brake is not used.

▷ Set-up values for DB based emergency stop are as follows.

Item	PARAMETER	Set-up value
Dynamic Brake is used	ALARM STATUS	ENB
	NORMAL STATUS	DIS

1.2.4.13 ESTOP --- Emergency Stop Configuration

1. Configuration Procedure

Step 1. **Move to ESTOP screen**

```
<XYZW : PROTECT(1)>
1: TRQ            2: TOL
3: OND            4: DB
5: ESTOP          6: MPDT
7: OVL2

                       item #
```

5 / M

Open PROTECT (1) screen.

Select 5: ESTOP.

Step 2. **E-STOP Screen**

```
<XYZW : E-STOP>
1: TRQ            2: TIME
3: OVS

                       item #
```

Emergency Stop configuration screen

Step 3-1. **EMERGENCY TRQ Configuration**

```
<XYZW : E-TRQ>
        TRQ LIMIT [%]
X: 300            Y : 300
Z: 300            W: 300
```

1 / Q

① Select "1: E-STOP TRQ".

Step 3-2. **EMERGENCY TIME Configuration**

```
<XYZW : E-TIME>
        STOP TIME(ms)
X: 300            Y : 300
Z: 300            W: 300
```

2 / R

② Select "2: E-STOP TIME".

Step 3-3. **EMERGENCY OVS Configuration**

```
<XYZW : E-OVS>
        OVER SPEED (rpm)
X: 5000              Y : 5000
Z: 5000              W: 5000
```

3
S

③ Select "3: E-STOP OVS".

▶ Change the default value (Use numeric keypad) & How to save (Refer to Page 1-12 > 1.1.2 Parameter Configuration.)

■ Explanation

① : E-STOP TRQ Explanation

▷ Parameter for setting torque limit to apply in emergency stop

▷ TRQ unit is %, ranging 1~300.

② : E-STOP TIME Explanation

▷ Parameter for setting time allowed for completion of emergency stop

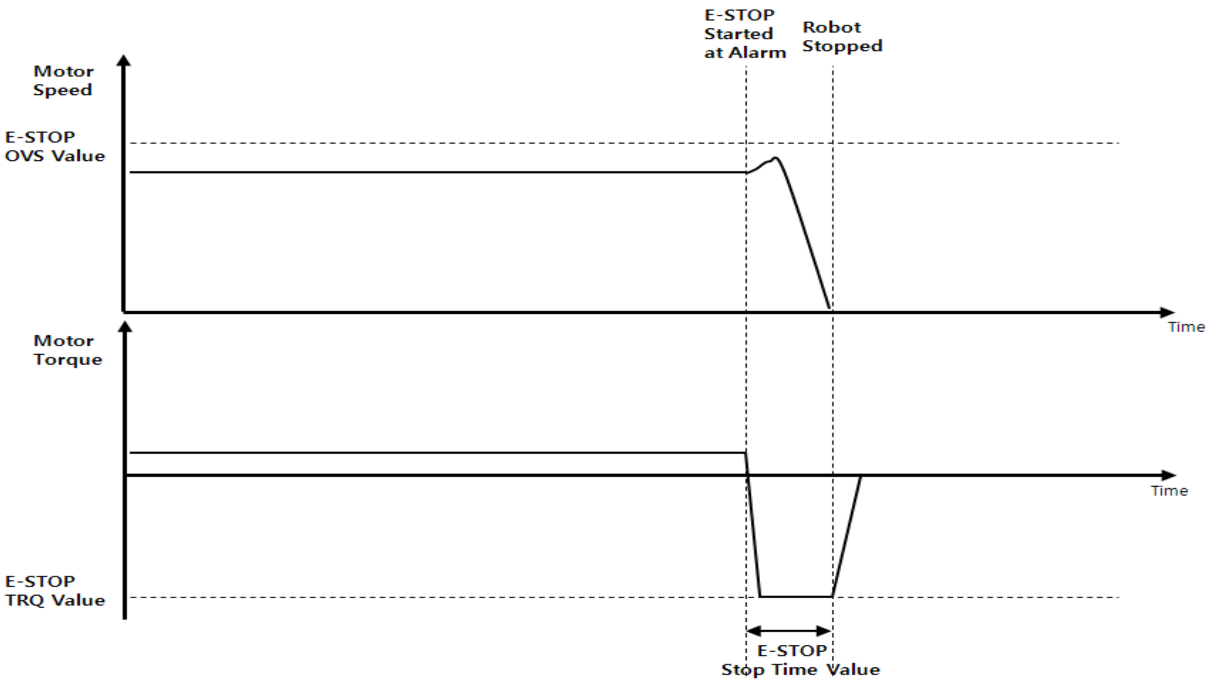
▷ When the time for emergency stop exceeds the default value, it turns into an alarm status and if set to 0, it does not check protection functions related to time for emergency stop.

▷ TIME unit is ms, ranging 0~1500.

③ : E-STOP SPEED Explanation

▷ Reference parameter for detecting an overspeed alarm in emergency stop.
When the motor speed goes beyond the default value, it turns into an alarm status and gives off an “E-STOP OVS” alarm.

▷ OVS unit is rpm, ranging 500~6000.



[Speed and Torque in Emergency Stop]

1.2.4.14 MPDT --- Motor Power Detection Time Configuration

1. Configuration Procedure

Step 1. **Move to MPDT screen**

```
<XYZW : PROTECT(1)>
1: TRQ          2: TOL
3: OND          4: DB
5: ESTOP        6: MPDT
7: OVL2

                    item #
```

6
N

Open PROTECT (1) screen.

Select 6: MPDT.

Step 2. **MPDT Screen**

```
<XYZW : MPDT>
    MOTOR POWER(ms)
TIME: 2000
```

Item for setting detection time for motor output.

▶ Change default value (Use numeric keypad) & How to save (Refer to Page 1-12 > 1.1.2 Parameter Settings.)

■ Explanation

① : Motor Power Detection Time

▷ Reference parameter for detecting motor power shutdown

▷ When the motor power shutdown status is longer than the parameter time, a “Motor Power Fail” alarm occurs. If the parameter set value is set to 2000, it does not detect if motor power shuts down.

▷ TIME unit is ms, ranging 70~3000.

1.2.4.15 OVL2--- Over Load 2 Error Configuration

1. Configuration Procedure

Step 1. **Move to OVL2 screen**

```
<XYZW : PROTECT(1)>
1: TRQ            2: TOL
3: OND            4: DB
5: ESTOP          6: MPDT
7: OVL2

                  item #
```

7 G

Open PROTECT (1) screen.

Select 7: OVL2.

Step 2. **OVL2 Screen**

```
<XYZW : OVL2>
1: OVL            2: OVT

                  item #
```

OVL2 Screen

Step 3. **OVER LOAD RATE 2 Configuration**

```
<XYZW : OVL2>
    OVER LOAD RATE 2(%)
X: 95             Y : 95
Z: 95             W: 95
```

1 Q

① "1: OVER LOAD RATE 2" configuration screen

Step 4. **OVL2 TOLERAN Configuration**

```
<XYZW : OVL2>
   OVL2 TOLERAN(msec)
X: 200            Y : 200
Z: 200            W: 200
```

2 R

② "2: OVL2 TOLERANCE" configuration screen

▶ Change default value (Use numeric keypad) & How to save (Refer to Page 1-12 > 1.1.2 Parameter Configuration.)

■ Explanation

① : Description of OVER LOAD RATE 2

▷ Reference parameter for detecting an alarm for an average motor load rate.

▷ Checks an average motor load rate in real time and gives off an alarm when beyond the default value. When an average load rate is generally 100%, an "overload" alarm occurs, but the default value can be changed when needing to generate an "overload" alarm according to user definition.

▷ OVL unit is %, ranging 1~100.

② : Description of OVL2 TOLERANCE

▷ Parameter for setting time allowed for the average load rate of a motor

▷ OVT unit is ms, ranging 10~2000.

1.2.5 ROBOT CONF

1. Items to be set up

ROBOT CONF	Description	Range	Set-up value
1. RENB	Stop the operation of robot mechanism	See the explanation	
2. USAX	Mutually change the axis	See the explanation	
3. XENB	Enable the axis	See the explanation	
4. P/R	Select In Position or In Range	See the explanation	
5. IRNG	In Range region	See the explanation	
6. UIRNG	Select the user In Range region and the output port	See the explanation	
7. SYNC	Synchronous robot setup	See the explanation	

2. Screen Configuration

Step 1. **Move to ROBOT CONF**

```
<XYZW : PARA(0)>
1: BODY          2: MOTION
3: GAIN          4: PROTECT

              group #
```

Pg Dn

Open PARAMETER (0) screen

Select Pg Dn

```
<XYZW : PARA(1)>
1: ROBOT CONF
2: FIXJOB

              group #
```

1
Q

Open PARAMETER (1) screen

Select 1: ROBOT CONF

Step 2. **ROBOT CONF setup screen**

```
<XYZW : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

              item #
```

ROBOT CONFIGURATION setup screen

1.2.5.1 RENB --- Select use or non-use of the selected channel robot

1. Setup Sequence

Step 1. **Move to RENB screen**

```
<XYZW : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                     item #
```

1 Q

Open ROBOT CONF screen

Select 1: RENB

Step 2. **Set up RENB**

```
<XYZW : RENB>
          ROBOT ENABLE
MASK:    DIS
```

ENTER

ROBOT ENABLE setup screen

DIS : Disable the robot

ENB : Enable the robot

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Set-up value ENTER change : ENB → DIS

▷ In order to use the selected channel robot, activate "ENB" the ROBOT ENABLE set-up value.

Item	Set-up value	Description
ROBOT ENABLE	ENB	Selected channel robot is usable.
	DIS	Selected channel robot is not usable.

CAUTION

▶ In order to use the selected channel robot, activate "ENB" the RENB parameter.

1.2.5.2 USAX --- Robot Using Axis setup

1. Setup Sequence

Step 1. **Move to USAX screen**

```
<XYZW : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                    item #
```

2 R

Open ROBOT CONF screen

Select 2: USAX

Step 2. **Set up USAX**

```
<XYZW : USAX>
     USAING AXIS SETTING
X: 1            Y : 2
Z: 3            W: 4
```

1 Q

4 L

① Set up USING AXIS SETTING

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

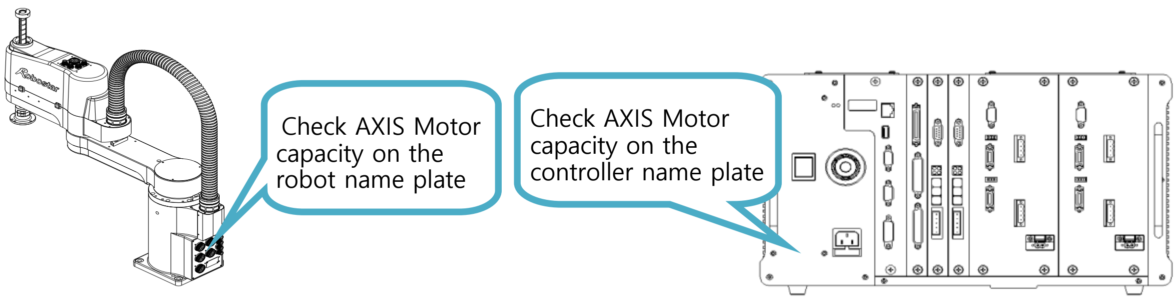
■ Comment

① : Description

▷ Controller servo pack output set-up value may be set to 1 to 4, and generally, in four-axis based system, it can be set up that A(X)-axis: 1, B(Y)-axis: 2, Z-axis: 3, and W-axis: 4.

▷ UXAX setup is related to the Hardware setup. Therefore, refer to the operation manual.

E.g.) If X is 1, X-axis output of the selected robot is output to No. 1 SERVO.



CAUTION	▶ USING AXIS set-up value is related to H/W setup. Therefore, refer to the operation manual, for change of the value.

1.2.5.3 XENB --- Robot Axis Enable setup

1. Setup Sequence

Step 1. **Move to XENB screen**

```
<XYZW : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                item #
```

3 S

Open ROBOT CONF screen

Select 3: XENB

Step 2. **Set up XENB**

```
<XYZW : XENB>
     ROBOT AXIS ENABLE
X: DIS           Y : ENB
Z: ENB           W: ENB
```

ENTER

① Disable the unused axis.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Set-up value ENTER Change : ENB → DIS

Item	Set-up value	Description
ROBOT AXIS DISABLE	ENB	Selected axis is used
	DIS	Selected axis is not used

▷ This item disables the unused axis depending on the robot type.

▷ The coordinate value of the disabled corresponding axis is not displayed on Teach Pendant.

▷ If X-AXIS is set to ON:

```
<XYZW : USAX>
     ROBOT AXIS ENABLE
X: DIS           Y : ENB
Z: ENB           W: ENB
```

Chage to point ⇒

```
<XYZW : AUTO>
F: TEST P:0 NW B L
X:               Y : 10.0
Z: 20.0          W: 20.0
```

1.2.5.4 P/R --- In Position / In Range selection

1. Setup Sequence

Step 1. **Move to P/R screen**

```
<XYZW : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                item #
```

4 / L

Open ROBOT CONF screen

Select 4: P/R

Step 2. **Set up P/R setup**

```
<XYZW : P/R>
     IN POS/RNG SELECT
      CURRENT : INPOS
1: INPOS         2: INRNG

               input #
```

1 / Q **or** **2 / R**

① In “CURRENT”, the select item is displayed.

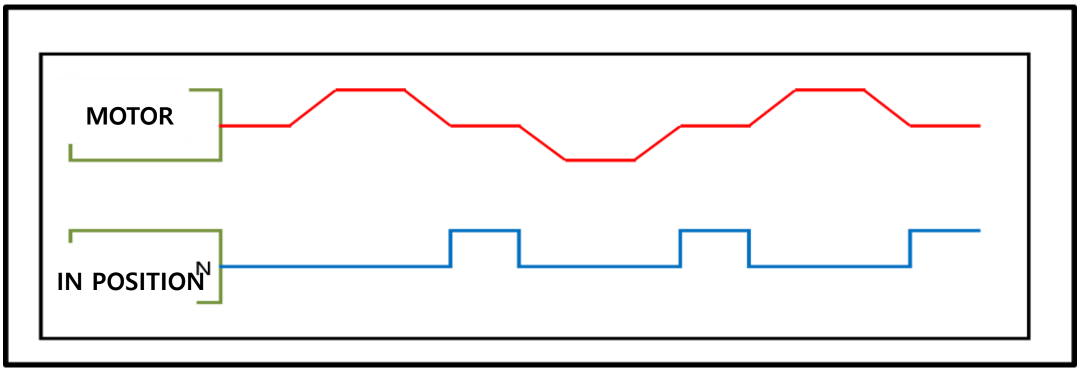
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

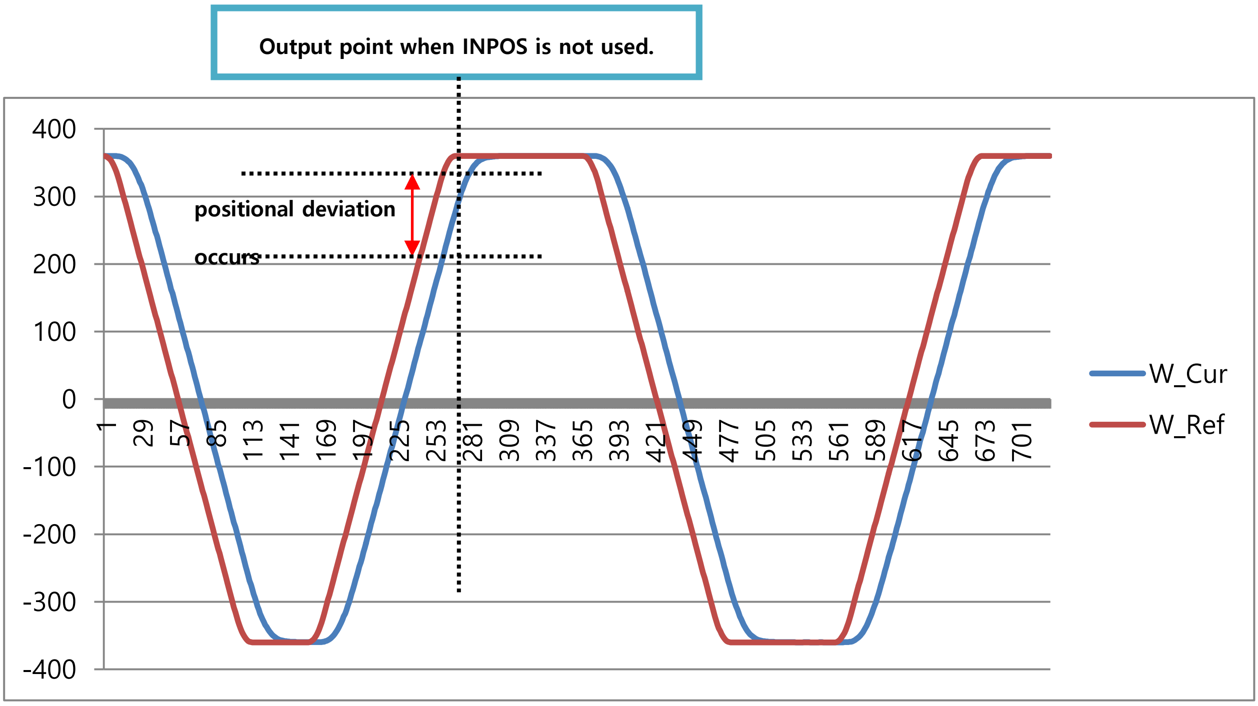
▷ In Position output of SYSTEM I/O signal of the controller has plural functions (“In Position” output and “In Range” output). The desired function of the plural functions can be set up in P/R Parameter.

▷ <u>INPOS : In Position Output</u>

- If there is the shaking within the predefined pulse (IPA) for the predefined time(IPE) at the stationery position while in robot operation, In Position output is turned on/off at every moving points while the robot is operating.



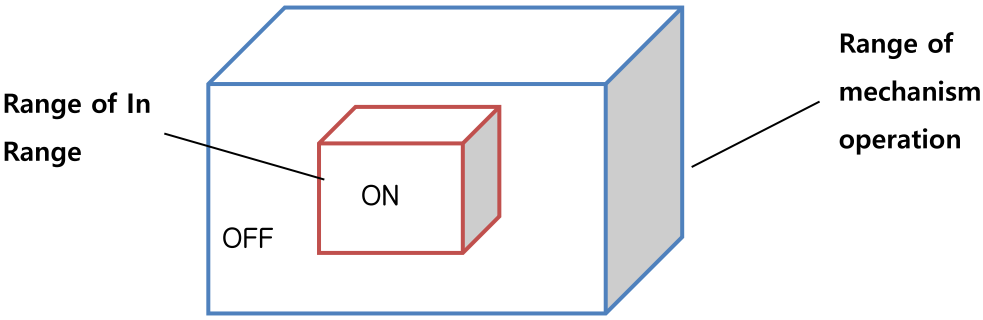
▷ Graph showing the positional deviation at the time of IN POSINON output



< Graph of the time of IN POSINON output>

▷ INRNG : In Range Output

- The robot mechanism is "On" within the IN RANGE setup range, and "Off" outside the setup range.
- The output is always(during Run and Teach Pendant operation) activated after Origin searching has been completed.



- If all the axes of the robot mechanism are, as shown in the above figure, within the range of In Range, IN POSITION output signal of SYSTEM I/O is "ON"; otherwise, "OFF".

1.2.5.5 IRNG --- In Range setup(RO)

1. Setup Sequence

Step 1. **Move to IRNG screen**

```
<XYZW : ROBOT CONF>
1: RENB            2: USAX
3: XENB            4: P/R
5: IRNG            6: UIRNG
7: SYNC

                 item #
```

5 M

Open ROBOT CONF screen

Select 5: IRNG

Step 2. **Set up IRNG**

```
<XYZW : IRNG>
          IN RANGE
X : 0            0
Y : 0            0
Z : 0            0
W: 0             0
```

① IN RANGE setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

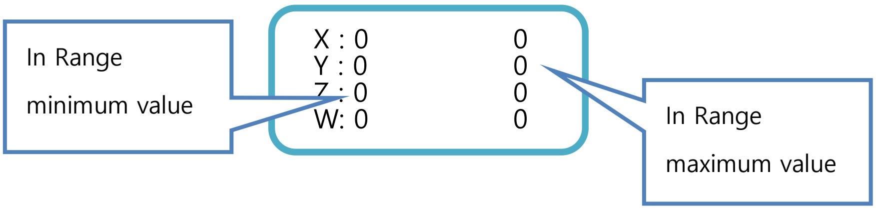
■ Comment

▷ Set the output range when in use of In Range

- Enter the minimum value and the maximum value respectively.

▷ The values set in In Range are valid only when P/R Parameter is set to IRNG.

▷ When P/R parameter is set to IRNG and if all the axes of the robot are within the range set up for In Range, In Position output signal of SYSTMEM I/O is "ON".



1.2.5.6 IRNG --- In Range Setting (TR)

※ In Range configuration screen may display different output screens according to version (RO ↔ TR).

1. Configuration Procedure

Step 1. **Move to IRNG screen**

```
<TZRRX : ROBOT CONF>
1: RDIS          2: USAX
3: XENB          4: P/R
5: IRNG          6: URNG
7: DSYNC

                        item #
```

5 M

Select "5: IRNG" from ROBOT CONF screen.

Step 2. **Move to IRNG screen**

```
<TZRRX : ROBOT CONF>
1: IN RANGE
2: ALARM

                        item #
```

1 Q

Select "1: IN RANGE" from IRNG screen.

Step 3. **IN RANGE Configuration**

```
<TZRRX : IRNG>
          IN RANGE
T: 0             0
Z: 0             0
L: 0             0
R: 0             0
X: 0             0
```

① "In Range" configuration screen

Step 4. **ALARM Configuration**

```
<TZRRX : IRNG>
          IN RANGE
ALARM: OFF
```

2 R

Select "2: ALARM" from IRNG screen.

② "ALARM" configuration screen

▶ Change default value (Use numeric keypad) & How to save (Refer to Page 1-12 > 1.1.2 Parameter Configuration.)

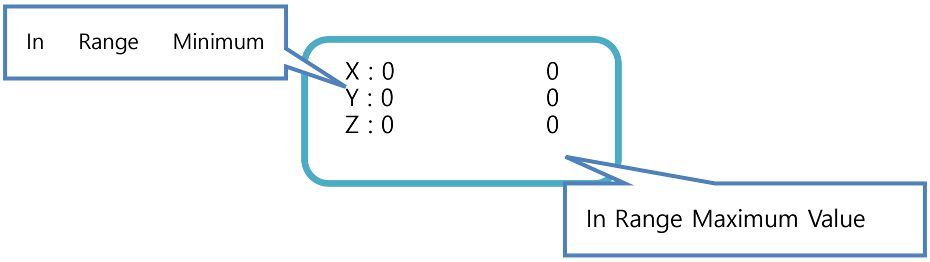
■ Explanation

▷ Enters a minimum value and maximum value of the axis to be used for In Range output.

▷ A robot capable of XY Coordinate conversion sets a range with X, Y, Z values.

▷ The values set in In Range is effective only when P/R parameter is set to IRNG.

▷ When P/R parameter is set to IRNG, In Position output in System I/O is On as long as all axis on robot stay within the area set in In Range.



▷ When beyond In Range scope, whether to generate an alarm is set.
OFF: Alarm not generated.
ON : In case of getting out of In Range in the middle of performing motion commands or FORWARD movement, it is set to give off an alarm -- Error Code: 1439 “IN RANGE Error”.

1.2.5.7 UIRNG --- User In Range setup(RO)

1. Setup Sequence

Step 1. **Move to UIRNG screen**

```
<RGA80B: ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                   item #
```

6 N

Open ROBOT CONF screen

Select 6: UIRNG

Step 2. **USER IN RANGE**

```
<RGA80B: UIRANG>
    USER IN RANGE
1: RANG1         2: RANG2
3: RANG3         4: RANG4
```

1 Q

4 L

Select RANGE1 to RANG4 (Select “1”)

```
<UIRANG: RANG1>
    USER IN RANGE SET

  CORD : ANGLE_TYPE


RANG   IO
```

ENTER

Set up the coordinate type

(ENTER to change ANGLE_TYPE<->XY_TYPE)

Step 3. **SETUP RANGE**

```
<UIRANG: RANG1>
    USER IN RANGE SET

  CORD : ANGLE_TYPE


RANG   IO
```

F1

Select RANG

```
<UIRANG: RANG1>
    USER IN RANGE
A: -10000        10000
B: -10000        10000
Z: -10000        10000
W:-10000         10000
```

① Set up the desired range for User In Range

Step 4. **OUTPUT RANGE**

```
<UIRANG: RANG1>
      USER IN RANGE SET

  CORD : ANGLE_TYPE



RANG   IO
```

F2

Select IO

```
<UIRANG : RANG1>
        OUTPUT SETUP
 OUTPUT : 79




        OUTPUT #.(0-79)
```

② Enter the output port number which turns the output "ON" when the relevant User In Range is reached.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

- ▷ User In Range setup is possible up to four ranges totally.
- ▷ Enter the minimum value and the maximum value per axis which are to be used for User In Range output.
- ▷ The set up values are valid only when P/R parameter is set to IRNG.
- ▷ When P/R parameter is set to IRNG and if all the axes of the robot are within the range set up for User In Range, the output from the designated port is turned "On".
- ▷ Enter the port number from which the signal outputs, when the mechanism is positioned at the range set up for User In Range.
- ▷ The output port set up for User In Range cannot be used in Robot Program and PLC Program.

1.2.5.8 UIRNG --- User In Range Configuration(TR)

※ User In Range configuration screen may have different output screens according to version (RO ↔ TR).

1. Configuration Procedure

Step 1. **Moe to URNG screen**

```
<TZRRX : ROBOT CONF>
1: RDIS          2: USAX
3: XENB          4: P/R
5: IRNG          6: URNG
7: DSYNC
```

6
N

Select “6: URNG” from ROBOT CONF screen.

Step 2. **Move to URNG<1> screen**

```
<TZRRX : URNG<1>>
1: URNG1         2: URNG2
3: URNG3         4: URNG4
5: URNG5         6: URNG6
7: URNG7         8: URNG8
9: URNG9         A: URNG10
```

Pg Dn

On URNG screen, move Pg Dn , Pq Up keys to select URNG page and enter the corresponding number.

Step 3. **Move to URNG<2> screen**

```
<TZRRX : URNG<2>>
1: URNG11        2: URNG12
3: URNG13        4: URNG14
5: URNG15        6: URNG16
7: URNG17        8: URNG18
9: URNG19        A: URNG20
```

Pg Up

Select “1: URNG1” from URNG<1> screen.

1
Q

Step 4. **Move to URNG1 screen**

```
<TZRRX : URNG1>
USER RANGE1
ENABLE: OFF
```

ENTER

Use “ENTER” key and select ON or OFF from ENABLE.

Step 5. **Move to URNG1 screen**

```
 <TZRRX : URNG1>
USER RANGE1
ENABLE: ON



RANG   OUT   ZDTA
```

F1

Select "F1" key and move to RANG configuration screen.

Step 6. **USER RANGE configuration**

```
 <TZRRX : URNG1>
          USER RANGE1
X: 0               0
Y: 0               0
Z: 0               0
```

① "User Range" configuration screen

Step 7. **OUTPUT configuration**

```
 <TZRRX : URNG1>
             OUT PUT
INDEX: 0
```

F2

Select "F2" key from Step5. screen and move to OUT PUT configuration screen.

② "Out put Index" configuration screen

Step 8. **ALARM configuration**

```
 <TZRRX : URNG1>
           Z DELTA(mm)
DELTA: 0
```

F3

Select "F5" key from Step5. screen and move to Z DELTA configuration screen.

③ "Z Delta(mm)" configuration screen

■ Explanation

▷ An alarm Error Code: 1439 "IN RANGE Error" does not take place if staying in USER RANGE, yet, beyond INRANGE.

▷ Enter USER IO OUTPUT number to display if IN/OUT in USER RANGE.

▷ Enter a travel distance value at which Z-axis can travel though remaining in USER RANGE.

▷ When the travel distance along Z-axis is greater than the default value though remaining in USER RANGE, an alarm Error Code: 1439 "IN RANGE Error" occurs.

1.2.5.9 SYNC --- Synchronous Robot setup

1. Items to be set up

Set-up values for Sync Parameter of RCQ controller are valid if the Robot Type is a synchronous robot. The set-up value is automatically changed by the robot type selected when in S/W initialization.

SYNC	Description	Range	Set-up value
1 : ENB	MASK: Set Synchronous Motion	See the explanation	
	TYPE : Select Synchronization Method	See the explanation	
2 : AXIS	Set up Synchronization Axis	See the explanation	
3 : DIR	Set up Synchronization Direction	+, -	
4 : ALARM	ENABLE: Check positional difference when Synchronization Point is saved.	ENB, DIS	
	RANGE: Setting the larm range for synchronous position point between axes	0.0~10.0	

2. Screen Configuration

Step 1. **Move to SYNC screen**

```
<DUAL : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                 item #
```

7
G

Open ROBOT CONF screen

Select 7: SYNC

Step 2. **SYNC Parameter**

```
<DUAL : SYNC>
1: ENB           2: AXIS
3: DIR           4: ALARM

                 item #
```

SYNC Parameter item

3. SYNC ROBOT TYPE

SYNC robot item which can be initialized through SETUP Parameter.

Step 1. **Move to SETUP screen**

```
<XYZW : PARA(2)>
1: SETUP



                    group #
```

1 / Q

On PARAMETER (2) screen, select “1: SETUP”

Step 2. **SYNC ROBOT**

```
<XYZW : CONF>
ROBOT TYPE: DUAL
1: SCARA          2: XYZW
3: DESK           4: DUAL
5: TR
                  8: BGT

                     item #
```

4 / L

Select synchronous robot group 4:DUAL

```
<XYZW : CONF>
ROBOT TYPE: DUAL
1. D2S            2. D2O
3. D3S_           4. D3O
5. D4S            6. D4O
7. DDSO           8. DDOS

                     item #
```

Synchronous robot SETUP screen 1

```
<XYZW : CONF>
ROBOT TYPE: DUAL
1. DDOO           2. DUAL



                     item #
```

Synchronous robot SETUP screen 2

■ Comment

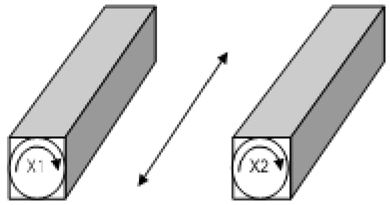
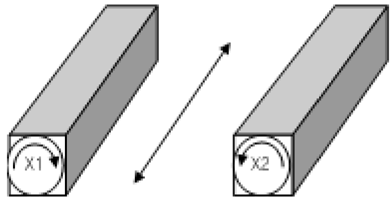
▷ When D2S to DDOO robot is set up, the robot is initialized with the default set-up value appropriate to the relevant robot type.

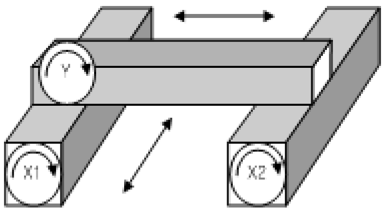
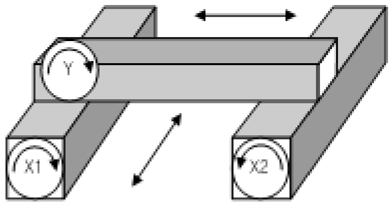
- Using SYNC Parameter, the initialized basic set-up value can be changed.

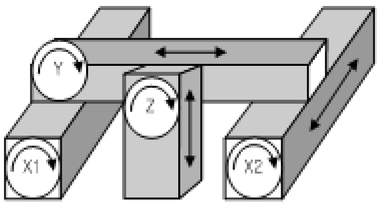
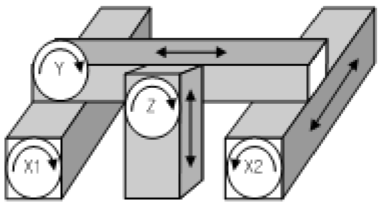
▷ When DUAL robot is set up, the user synchronization type can be formed by using SYNC Parameter.

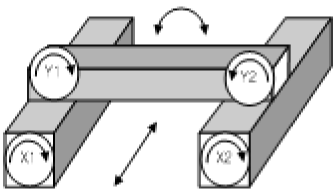
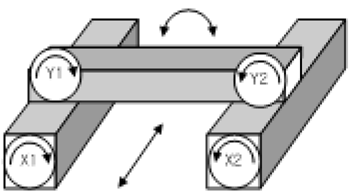
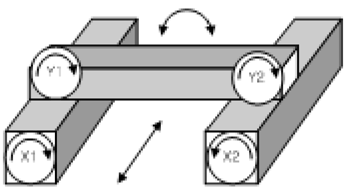
4. DEFAULT SYNC ROBOT TYPE

The followings are the default models of the setup parameter that are usable when in synchronous robot setup.

	D2S	D2O
XY-axis		

	D3S	D3O
XZ-axis		

	D4S	D4O
XW-axis		

	DDSO	DDOO	DDOS
XZ and YW-axis			

Important

- For "D2S to DDOO" robot, the set-up values appropriate to the synchronization type are initialized.
- After "D2S to DDOO" robot initialization, the set-up value can be changed by using SYNC Parameter.
- For "DUAL" robot, the synchronization type can be set up by using SYNC Parameter.

ENB setup

1. Setup Sequence

Step 1. Move to SYNC screen

```
<DUAL : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                 item #
```

7
G

Open SYNC screen

Select 7: SYNC

Step 2. Set up ENABLE

```
<DUAL : SYNC>
1: ENB           2: AXIS
3: DIR           4: ALARM

                 item #
```

1
Q

Select 1: ENB

```
<DUAL : SYNC_ENB>
     SYNC TYPE SETTING
1. MASK: OFF
2. TYPE  : ABSOLUTE
```

MASK: Set up Synchronous Motion

TYPE : Select Absolute/Relative Synchronization

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ MASK: Set up Synchronous Motion

- When the synchronous robot is set up, MASK value is ON state.
- If JOB is set up, the synchronization point is adapted to a point after JOB command.

Item	Set-up value	Description
MASK	OFF	Asynchronous
	ON	Synchronous
	JOB	Synchronous when JOB command is used

▷ TYPE: Synchronization type setup

- When the positional deviation occurs due to abrupt stop by the external EMG signal and if SVON operates only in ABS Mode, the position compensation is performed toward the master position.
- REL. Sync Mode moves from the SLAVE current position by the MASTER position movement amount.

Item	Set-up value	Description	Feature
TYPE	ABSOLUTE	Absolute synchronization	Compensation when in positional deviation.
	RELATIVE	Relative synchronization	Synchronization at the current position regardless of positional deviation.

▷ Synchronous driving types according to ENBLE setup are as follows.

- For the absolute synchronization in which MASK is ON state, change ORG->SEQ value to accord to the synchronization type.

Item	MASK	TYPE	ORG	JOG	RUN
ENB	OFF	ABSOLUTE	Individual	Individual	Individual
		RELATIVE	Individual	Individual	Individual
	ON	ABSOLUTE	Synchronous	Synchronous	Absolute Position Synchronization
		RELATIVE	Individual	Synchronous	Relative Position Synchronization
	JOB	ABSOLUTE	Individual	Individual	Absolute Position Synchronization when the command is used
		RELATIVE	Individual	Individual	Relative Position Synchronization when the command is used

Important

▶ In ABS Synchronous Mode, if the positional difference between the master and the slave is large and when in SVON operation, “OVER SPPED” alarm may occur.

▶ When in using ABS Synchronous Mode, if the positional difference is large, manually move the relevant slave axis to the master position.

▶ In REL Synchronous Mode, pay attention that the relevant slave moves by the master moving amount, regardless of the position of the master and the slave.

⚠ 주의 ▶ If “Please Power off!” message is displayed, turn the power on/off.

AXIS setup

1. Setup Sequence

Step 1. **Move to SYNC screen**

```
<DUAL : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                 item #
```

7 G

Open SYNC screen

Select 7: SYNC

Step 2. **Set up AXIS**

```
<DUAL : SYNC>
1: ENB           2: AXIS
3: DIR           4: ALARM

                 item #
```

2 R

Select 2: AXIS

```
<DUAL : SYNC_AXIS>
     SYNC AXIS SETTING
X: NORMAL        Y : NORMAL
Z: NORMAL        W: NORMAL
```

Set up MASTER/SLAVE axis

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ When in S/W initialization, the set-up value automatically changes according to the type of synchronous robot.

▷ When in MASTER axis setup, it is recommended to use X, and Y-AXIS.

Item	Set-up value	Description	Axis settable.
MASTER	MAST1	First reference axis	X, Y
	MAST2	Second reference axis	X, Y
SLAVE	SLAV1	Slave axis of MAST1	Y, Z, W
	SLAV2	Slave axis of MAST2	Y, Z, W
NORMAL	NORMAL	Normal axis	X,Y,Z,W

Important

▶ For execution of smooth synchronous motion, set X-AXIS to First Master when in master axis setup. For use of dual synchronous construction, such as DDSO, DDOS, or DDOO type, set Y-AXIS to Second Master.

▶ If MASTER axis is not set to the recommended X or Y-axis, the synchronous operation may not be performed when in interpolation operation, such as LMOV, AMOV, CMOV, etc.

CAUTION

▶ If "Please Power off!" message is displayed, turn the power on/off.

DIR setup

1. Setup Sequence

Step 1. **Move to SYNC screen**

```
<DUAL : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                 item #
```

7
G

Open SYNC screen

Select 7: SYNC

Step 2. **Set up DIRECTION**

```
<DUAL : SYNC>
1: ENB           2: AXIS
3: DIR           4: ALARM

                 item #
```

3
S

Select 3:DIR

```
<DUAL : SYNC_DIR>
      SYNC DIRECTION
X: +             Y : +
Z: +             W: +
```

Set up synchronization direction

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Synchronization direction can be set up.
- Changing DIR set up of *SYNC* Parameter is available for SLAVE axis only.

Item	Set-up value	Description
SLAVE	+	Forward sync.
	-	Reverse sync.

▷ The followings are parameter changes when in synchronous robot setup.

	2:MOTION -> 4:DISP		1:ROBOT CONF -> 7:SYNC -> 3:DIR	
Item	Set-up value	Description	Set-up value	Description
MASTER	+	Available	+	Available
	-	Not available	-	Not available
SLAVE	+	Available	+	Forward sync.
	-	Available	-	Reverse sync.
NORMAL	+	Available	+	Available
	-	Available	-	Not available

Important

▶ For MASTER axis

(1) PARA(0) -> 2: MOTION ->4: DISP value is fixed to "+" and cannot be changed.

(2) PARA(1) -> 1: ROBOT CONF -> 7:SYNC -> 3:DIR value is fixed to "+".

▶ For SLAVE axis

(1) PARA(0) -> 2: MOTION ->4: DISP value is changeable between "+" and "-", and when in change, only the sign of display position is changed.

(2) PARA(1) -> 1: ROBOT CONF -> 7:SYNC -> 3:DIR value is changeable between "+" and "-", and when in change, the rotation direction of a motor is changed.

▶ For NORMAL axis

(1) PARA(0) -> 2: MOTION ->4: DISP value is changeable between "+" and "-", and when in change, the rotation direction of a motor is changed.

(2) PARA(1) -> 1: ROBOT CONF -> 7:SYNC -> 3:DIR value is fixed to "+".

CAUTION

▶ If "Please Power off!" message is displayed, turn the power on/off.

ALARM setup

1. Setup Sequence

Step 1. **Move to SYNC screen**

```
<DUAL : ROBOT CONF>
1: RENB          2: USAX
3: XENB          4: P/R
5: IRNG          6: UIRNG
7: SYNC

                 item #
```

7
G

Open SYNC screen

Select 7: SYNC

Step 2. **Set up ALARM**

```
<DUAL : SYNC>
1: ENB           2: AXIS
3: DIR           4: ALARM

                 item #
```

4
L

Select 4:ALARM

```
<DUAL : ALARM>
    SYNC ALARM SETUP
1: ENABLE        2: RANGE

                 item #
```

Alarm Setup Screen

Step 2-1. **ALARM ENABLE**

```
<DUAL : ALARM>
    SYNC ALARM SETUP
1: ENABLE        2: RANGE

                 item #
```

1
Q

Setting the alarm for position difference when saving MASTER/SLAVE point

```
<DUAL : ALARM>
   POINT ALARM CHECK
ENABLE : ENB
```

ENABLE:

-> Activation of position alarm inspection

DISABLE:

-> Deactivation of position alarm inspection

Step 2-2. **ALARM RANGE**

```
<DUAL : ALARM>
      SYNC ALARM SETUP
1: ENABLE          2: RANGE



                      item #
```

2
R

Setting the alarm range for position difference when saving MASTER/SLAVE point

```
<DUAL : ALARM>
      POINT ALARM CHECK
RANGE  : 0.02
```

Setting the larm range for synchronous position point between axes

Initial value : 0.02

Setting range : 0.0 ~ 10.0

Unit : mm

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Alarm check against the positional difference between MASTER and SLAVE when in saving the point.

Item	Set-up value	Description
ALARM	ENB	Alarm when the positional difference between MASTER/SLAVE is 0.02or more.
	DIS	Alarm check is not performed.

▷ Setting the larm range for synchronous position point between axes

Item	Set-up value	Description
ALARM RANGE	Initial value	0.02
	Setting range	0.0~10.0
	Unit	mm

▷ Alarm check table according to the setup.

4: ALARM	1: ENB		ALARM CHECK
ENABLE	MASK	TYPE	
ENB	OFF	ABSOLUTE	X
		RELATIVE	X
	ON	ABSOLUTE	O
		RELATIVE	X
	JOB	ABSOLUTE	X
		RELATIVE	X

Important

▶ ALARM check is available only in absolute synchronous mode in which MASK state is ENABLE.

CAUTION

▶ If “Please Power off!” message is displayed, turn the power on/off.

1.2.6 **FIXJOB**

1. Setup Sequence

Step 1. **Move to FIXJOB screen**

```
<XYZW : PARA(1)>
1: ROBOT CONF
2: FIXJOB



            group #
```

2 / R

Open PARAMETER (1) screen

Select 2: FIXJOB

Step 2. **Set up FIXJOB**

```
<XYZW : FIXJOB>
1: ENB              2: SELJOB



            item #
```

1 / Q

Select 1: ENB

```
<XYZW : FIXJOB>
       FIX SYS MODE JOB
FIXJOB:  DIS
```

FIXJOB setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ If FIXJOB set-up value is "ENB", Program Select is not selected in SYSTEM MODE, and JOB selected in "2: SELJOB" is executed when START.

▷ If FIXJOB set-up value is "DIS", No. 0 job is executed only for CH 1.

1.2.7 SELJOB

1. Setup Sequence

Step 1. **Move to FIXJOB screen**

```
<XYZW : PARA(1)>
1: ROBOT CONF
2: FIXJOB

                    group #
```

2 R

Open PARAMETER (1) screen

Select 2: FIXJOB

Step 2. **Set up SELJOB**

```
<XYZW : FIXJOB>
1: ENB              2: SELJOB

                     item #
```

2 R

Select 2: SELJOB

```
<XYZW : SYS>
 SYSTEM DEFAULT JOB
 JOB NAME = TEST

FIND                 SAVE
```

F1

Select FIND

```
<RUN MODE>
 FILES    F:3
0        TEST      1        AJOB
2        BJOB
```

Select JOB which will be executed in SYSTEM MODE

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Only if FIXJOB set-up value is "ENB", the job to be executed in SYSTEM MODE can be selected.
▷ Select FIND to select a job to be executed in SYSTEM MODE.

1.2.8 SETUP

1. Items to be set up

PARAMETER(2)	Description	Set-up value
1 : SETUP	Spec. of the robot mechanism is set in the controller when shipment of the product	Factory setup

2. Screen Configuration

Step 1. Move to PARA screen

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                SELECT #
```

4 L

Open intial menu screen

Select 4: PARA

```
<XYZW : PARA(0)>
1: BODY         2: MOTION
3: GAIN         4: PROTECT

                group #
```

Pg Dn

Open PARAMETER (0) screen

Select Pg Dn

```
<XYZW : PARA(1)>
1: ROBOT CONF
2: FIXJOB

                group #
```

Pg Dn

Open PARAMETER (1) screen

Select Pg Dn

```
<XYZW : PARA(2)>
1: SETUP

                group #
```

1 Q

Open PARAMETER (2) screen

Select SETUP

Important

▶ SETUP value has been set in the factory according to the spec of robot mechanism.

▶ Since it has been set by a final inspector when in shipment of the factory, do not change the set-up value arbitrarily. If you want to change the value, contact Customer Support Team.

1.2.8.1 SETUP --- robot type setup

1. Setup Sequence

Step 1. Move to SETUP screen

```
<XYZW : PARA(2)>
1: SETUP



                group #
```

1 Q

Open PARAMETER (2) screen

Select 1: SETUP

Step 2-1. Set up SCARA ROBOT

```
<RSA60A: CONF>
ROBOT TYPE: RSA60A
1. SCARA        2. XYZW
3. DESK         4. DUAL
5. TR
                8. BGT

                   item #
```

1 Q

Select SCARA ROBOT

```
<RSA60A: CONF>
ROBOT TYPE: RSA60A
1: RSA-40A      2: RSA-50A
3: RSA-60A      4: RSA-70A
5: RSA-70B      6: RGA-80A
7: RGA-80B      8: RHA-80C
```

Pg Up

SCARA ROBOT PAGE1

```
<RSA60A: CONF>
ROBOT TYPE: RSA60A
1: RBSA-90A     2: RBSA-90B
3: RGA-A0A      4: RGA-A0B
5: RGA-A1A      6: RGA-A1B
7: SC-TEST
```

Pg Dn

SCARA ROBOT PAGE2

Step 2-2. Set up CARTESIAN ROBOT

```
<XYZW: CONF>
ROBOT TYPE: XYZW
1. SCARA        2. XYZW
3. DESK         4. DUAL
5. TR
                8. BGT

                   item #
```

2 R

Select CARTESIAN ROBOT

<XYZW: CONF>
ROBOT TYPE: XYZW

1: XY__	2: X_Z_
3: XYZ_	4: XZY_
5: XY_W	6: X_ZW
7: XYZW	8: XZYW

CARTESIAN ROBOT PAGE1

Pg Up

Pg Dn

<XYZW : CONF>
ROBOT TYPE: XYZW

1: XYZH	2: XZYR
3: XYXY	4: XYZXY_
5: XYZXYZ	6: XY-TEST

CARTESIAN ROBOT PAGE2

Step 2-3. Set up DESK TOP ROBOT

<RDT4_A: CONF>
ROBOT TYPE: RDT4_A

1. SCARA	2. XYZW
3. DESK	4. DUAL
5. TR	
	8. BGT

item #

Select DESK TOP ROBOT

3
S

<RDT4_A: CONF>
ROBOT TYPE: RDT4_A

1: RDS2_	2: RDS3_
3: RDS4_	4: RDT4_F
5: RDT4_A	

DESK TOP ROBOT PAGE1

Step 2-4. Set up DUAL ROBOT

<D4S: CONF>
ROBOT TYPE: D4S

1. SCARA	2. XYZW
3. DESK	4. DUAL
5. TR	
	8. BGT

item #

Select DUAL ROBOT

4
L

```
<D4S: CONF>
ROBOT TYPE: D4S
1: D2S          2: D2O
3: D3S          4: D3O
5: D4S          6: D4O
7: DDSO         8: DDOS
```

Pg Up

DUAL ROBOT PAGE1

```
<D4S: CONF>
ROBOT TYPE: D4S
1: DDOO         2: DUAL
```

Pg Dn

DUAL ROBOT PAGE2

Step 2-5. Set up TR ROBOT

```
<TML__: CONF>
ROBOT TYPE: TML__
1. SCARA        2. XYZW
3. DESK         4. DUAL
5. TR
                8. BGT

                   item #
```

5
M

Select TR ROBOT

```
<TML__: CONF>
ROBOT TYPE: TML__
1: TML__        2: TMS__
3: CML__        4: ROSEP300
```

TR ROBOT PAGE1

Step 2-6. Set up BGT

```
<BGT: CONF>
ROBOT TYPE: BGT
1. SCARA        2. XYZW
3. DESK         4. DUAL
5. TR
                8. BGT

                   item #
```

8
H

Select BGT channel

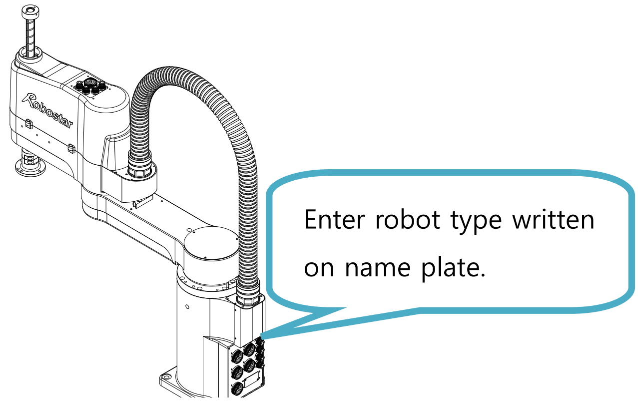
```
<BGT: CONF>
ROBOT TYPE: BGT
1: BGT
```

BGT PAGE1

■ Comment

▷ Select a robot type to be operated.

▶ E.g.) To operate SCARA robot, enter the robot type written on SCARA robot name plate.



▶ Nameplate (Label) in mechanical part --- mounted on connector at rear of robot (RO)

▶ A-axis: 450mm
B-axis: 350mm
Z-axis: 300mm

▶ Lot No. :
Unique number in shipping from factory

▶ Type of Mechanical part : Refer to Name Description in Handling Instruction.

SCARA ROBOT

MODEL : RGA100-4A

P.CODE : RGA10004B300P

LENGTH : A = 550, B = 450

STORKE : Z = 300

Ser. NO : 120625015

▶ Motor capacity per axis : 750W/A4
Minas Motor 750W

▶ Pulse value of motor encoders per axis

▶ Display per axis

Axis	Motor	Encoder
A	750W/A4	Abs.17bit
B	400W/A4	Abs.17bit
Z	400W/A4	Abs.17bit
W	200W/A4	Abs.17bit

▶ Display Peach per axis
▶ Display Gear per axis

▶ Length per axis

▶ A,B-axis offset value

▶ Display each axis

Axis	P/G	Length	offset
A	1/89	549.644	0.09
B	1/89	449.715	-0.14
Z	25/1.25	300	-13.50
W	1/30	360	

▶ Display EPOS per axis

▶ Display each axis

▶ Display Callb per axis

Axis	EPOS Data	Callb
A	113.369	0.942
B	145.708	1.215
Z	-13.500	-45.828
W	0.361	-445.95

▶ Nameplate (Label) in mechanical part --- mounted on connector at rear of robot (TR)

▶ A-axis: 2800mm
B-axis: 2800mm

▶ Lot No. :
Unique number in shipping from factory

▶ Type of Mechanical part : Refer to Name Description in Handling Instruction.

TRANSFER ROBOT

MODEL : ROSEP300

P.CODE : TM962

LENGTH : A = 280, B = 280

TOOL : W1 = 440, W2=440

Ser. NO : 120625015

▶ Motor capacity per axis: 750W/A4
Minas Motor 750W

▶ Display each axis

▶ Pulse value of motor encoders per axis

Axis	Motor	Encoder
A	200W/A4	Abs.17Bit
B	400W/A4	Abs.17Bit
Z	200W/A4	Abs.17Bit
W1	100W/A4	Abs.17Bit
W2	100W/A4	Abs.17bit

▶ Display Gear per axis

▶ Display Peach per axis

▶ Display offset per axis

▶ Display each axis

Axis	Gear	Pitch	Offset
A	1:89	1	330
B	1:89	1	330
Z	1:1.25	20	600mm
W1	1:30	1	340
W2	1:3	1	340

▶ Display EPOS per axis

▶ Display Callb per axis

▶ Display each axis

Axis	EPOS	Callb
A	13.369	0.942
B	45.708	1.215
Z	-13.50	-45.828
W1	0.361	-445.95
W2	0.361	-445.95

▶ Each axis –Software Limit

▶ Each axis+Software Limit

▶ Display each axis

Axis	-RANG	+RANG
A	0	330
B	0	330
Z	-20	600mm
W1	0	340
W2	0	340

1.3 PUBLIC PARAMETER

1.3.1 HW CONF

1. Items to be set up

HW CONF	Description	Range	Set-up value
1 : TMR	TIMER reference time	See the explanation	
2 : COMM	Computer communication spec.	See the explanation	
3 : I/O	Number of I/O board	See the explanation	
4 : TP	Teach Pendant EMG Type, Deadman, Model setup	See the explanation	
5 : SVON	Auto Servo ON enable	See the explanation	
6 : BRAM	JOB, POINT item Initial	See the explanation	
7 : MAX CH	Maximum channel number available	See the explanation	

2. Screen Configuration

Step 1. **Move to PARA screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                SELECT #
```

4
L

Open intial menu screen

Select 4: PARA

```
<PARAMETER>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST

SEL   INFO   PUB   EXIT
```

F3

Open PUBLIC PARAMETER group screen

```
<PUBLIC PARAMETER>
1: HW CONF      2: PALLET
3: PLC          4: ETC

                group #
```

1
Q

Select 1:HW CONF

<PUBLIC-HW CONF(0)>
1: TMR 2: COMM
3: I/O 4: TP
5: SVON 6: A I/O

item #

Pg Dn

HW CONF(0) setup screen

<PUBLIC-HW CONF(1)>
1: BRAM 2: MAX CH

item #

HW CONF(1) setup screen

1.3.1.1 TMR --- Timer's time setup

1. Setup Sequence

Step 1. **Move to TMR screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O


                  item #
```

1
Q

Open HW CONF (0) screen

Select 1: TMR

Step 2. **TMR setup screen**

```
<HW CONF - TMR>
      TMR SET (10msec)
TM0 : 1
TM1 : 1
```

TIMER setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ When a time is used in a robot program command, the reference time of system parameter is set up.

TMR system parameter	
Unit	millisecond
Setup range	1 to 60000

1.3.1.2 COMM -- Online communication setup

1. Setup Sequence

Step 1. Move to COMM screen

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: D-MAN
5: SVON         6: A I/O


           item #
```

2 R

Open HW CONF(0) screen

Select 2: COMM

Step 2. RS232 setup screen

```
<HW CONF - COM>
  COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR

            group #
```

1 Q

Select 1: RS232C

```
<COM-RS232C>
PROTOCOL:      N1
BAUD RATE:     230400
LENGTH:        8 bits
STIP BIT:      1 bits
PARITY:        disable
FLOW:          NONE
ID:            0
```

RS232C setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Spec. for computer and RS-232C communication

Item	Set-up value	
PROTOCOL	N1	Unihost communication is used
	CNET	C-Net protocol is used
	STRING	String command is used
BAUD RATE	9600 to 2500000	
LENGTH	5 to 8	

Item	Set-up value
STOP BIT	1 → 1.5 → 2 → 1
PARITY	disable → even → odd
FLOW	NONE → XON/XOF → HARDWARE
ID	0 to 255

OPTION COM CARD setup procedure

1. Setup Sequence

Step 1. **Move to COMM screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: D-MAN
5: SVON         6: A I/O

                item #
```

2 R

Open HW CONF(0) screen

Select 2: COMM

Step 2. **Move to OPT CARD screen**

```
<HW CONF - COM>
   COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR

                group #
```

2 R

Select 2: FIELD BUS

```
<COM-FDBUS>
1: CARD
2: USER I./O
3: PROFIBUS ENDIAN
4: MAP EXTENSION

                input #
```

1 Q

Select 1: CARD

Step 3. **CARD setup screen**

```
<FDBUS-CARD>
        OPT COM CARD
1: NONE         2: CC-LINK
3: PROFIBUS     4: D-NET

Selected :  NONE
```

1 Q

4 L

Select the relevant OPT CARD

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Select the relevant communication type when a master PLC is used.

Item	Set-up value	Communication type	PLC
COMM	1: NONE	No selection	When selecting OPTION I/O
	2: CC-LINK	CC-Link communication	MITSUBISHI PLC
	3: PROFIBUS	Profibus DP communication	SIEMENS PLC
	4: DEVICENET	DeviceNet communication	HILSCHER PLC, AB PLC, OMRON PLC

※ Initial setup is "NONE", and can be set up according to the optional communication type.

USER I/O setup procedure

1. Setup Sequence

Step 1. **Move to USER I/O screen**

```
<HW CONF - COM>
   COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR

                    group #
```

2 R

Open COMM screen

Select 2: FIELD BUS

```
<COM-FDBUS>
1: CARD
2: USER I./O
3: PROFIBUS ENDIAN
4: MAP EXTENSION

                  input #
```

2 R

Select 2: USER I/O

Step 2. **USER I/O setup screen**

```
<FDBUS-USER I/O>
       USER IN/OUT SEL
USER IO : SYS U I/O
```

ENTER

Select SYS U I/O" or "FIELD U I/O

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Use formation of USER I/O is set up when using Field Bus Card.

Item	Description
SYS USER I/O	Input and output using USER I/O of N1 System IO B/D
FIELDBUS USER I/O	Input and output using USER I/O of Field Bus Card

PROFIBUS ENDIAN

1. Setup Sequence

Step 1. **Move to PROFIBUS ENDIAN screen**

```
<HW CONF - COM>
   COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR

                 group #
```

Open COMM screen

Select 2: FIELD BUS

2 / R

```
<COM-FDBUS>
1: CARD
2: DPRAM USER I./O
3: PROFIBUS ENDIAN
4: MAP EXTENSION

                 input #
```

Select 3: PROFIBUS ENDIAN

3 / S

Step 2. **PROFIBUS EDIAN setup screen**

```
<FDBUS-ENDIAN>
1: Little Endian
2: Big Endian


Selected: LITTLE
```

EDIAN setup screen

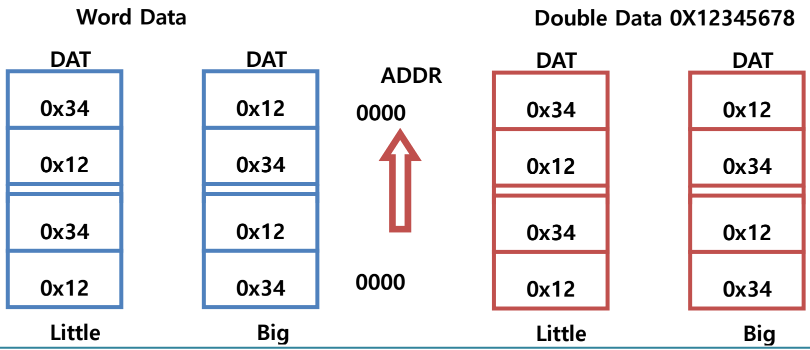
1 / Q **or** **2 / R**

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ When Profibus is selected, the sequence of the transmission and updating Word or Double Data in the communication with Master PLC is set up.

Item	Description
Little Endian	Data is byte-wise saved sequentially from the lower address.
Big Endian	The upper Byte and Word value are saved in the lower address.



MAP EXTENSION

1. Setup Sequence

Step 1. Move to MAP EXTENSION screen

<HW CONF - COM>
COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR
group #

2 R

Open COMM screen

Select 2: FIELD BUS

<COM-FDBUS>
1: CARD
2: DPRAM USER I./O
3: PROFIBUS ENDIAN
4: MAP EXTENSION
input #

4 L

Select 4: MAP EXTENSION

Step 2. MAP EXTENSION setup screen

<FDBUS MAP>
SEL FIELD BUS MAP
MAP : NORMAL

ENTER

NORMAL ↔ EXTENSION ↔ USER_DEFINE

■ Comment

▷ Parameter used when in the extension of FIELD BUS MAP, which is to be implemented later.

Item	Description	Map	Max number of axis
NORMAL	Basic Fieldbus Map	Fixed	4 axes
EXTENSION	Extended Fieldbus Map	Fixed	6 axes
USER DEFINE	User Defined Fieldbus Map	Flexible	User defined

CAUTION

▶ Function of MAP EXTENSION PARAMETER will be implemented later.

LINE SEPARATOR

1. 설정순서

Step 1. **Move to LINE SEPARATOR screen**

<PUBLIC-HW CONF(0)>
1: TMR 2: COMM
3: I/O 4: D-MAN
5: SVON 6: A I/O
item #

Open HW CONF(0) screen
Select 2: COMM

2 R

<HW CONF - COM>
COMMUNICATION SET
1: RS232C
2: FIELD BUS
3: LINE SEPARATOR
group #

Select 3: LINE SEPARATOR

3 S

Step 2. **LINE SEPARATOR setup screen**

<COM-LINE SEPARATOR>
LINE SEPARATOR
TYPE : CR+LF

CR+LF ↔ CR

ENTER

■ 해 설

▷ This is a parameter for setting the end time of a text string when in STRING command communication.

항 목	내 용
CR+LF	The end time of a text string is judged by Carriage Return + Line Feed, when in transmission and reception of the text string.
CR	The end time of a text string is judged by Carriage Return, when in transmission and reception of the text string.

CAUTION	▶When transmitting "Text string + CR + LF" after setting up the line separator parameter value to CR After "Text string + CR" is recognized, "LF" value is recognized as a first letter of the next transmission. ▶When transmitting "Text string + CR" after setting up the line separator parameter value to "CR + LF" Since transmission of the text string has not been completed, Timeout may be generated. ※ Refer to TIMEOUT of 3.37 RSTATE (Robot state check command) in Programming

1.3.1.3 I/O --- Number of Extension I/O board setup

1. Setup Sequence

Step 1. **Move to I/O screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O

                item #
```

3
S

Open HW CONF(0) screen

Select 3: I/O

Step 2. **DIO setup screen**

```
<HW CONF-I/O>
      I/O EXT B/D CNT
        VALUE = 0
```

I/O EXTENSION B/D setup

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ N1 controller can use IO cards for each slot, respectively.

Basically, N1-SYS I/O is used for one slot, and the user extension I/O is optionally used for the remaining two slots.

	Set-up value	Description
VALUE	0	Basic I/O(System I/O(24/12)+ User I/O(16/16))
	1	Basic I/O+ Option I/O (32/32)
	2	Basic I/O+ Option I/O (64/64)

Important

▶ N1-SYS I/O is supported for a first slot.

1.3.1.4 TP --- TEACH PENDANT setup

Emergency Type setup

1. Setup Sequence

Step 1. **Move to TP screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3. I/O          4. TP
55: SVON        6: A I/O

                  item #
```

4
L

Open HW CONF(0) screen

Select 4: TP

Step 2. **Set up EMG TYPE**

```
<HW CONF-T/P>
       T/Pendant TYPE
1: EMG TYPE
2: DEADMAN S/W
3: MODEL

                 group #
```

1
Q

Select 1: EMG TYPE

```
<T/P-EMG TYPE>
       T/P EMERGENCY
TYPE :   NC
```

ENTER

"ENTER" to select

NC ↔ NO

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Teach Pendant Emergency Type setup

Set-up value	Description
NC	T/P Emergency - Normal Closed Type
NO	T/P Emergency - Normal Opened Type

DEADMAN Switch setup

1. Setup Sequence

Step 1. **Move to DEADMAN screen**

<HW CONF-T/P>
T/Pendant TYPE
1: EMG TYPE
2: DEADMAN S/W
3: MODEL

group #

2 R

Select 2 : DEADMAN S/W

Step 2. **Set up DEADMAN**

<T/P-DEADMAN S/W>
DEADMAN SWITCH
MASK: DIS

ENTER

"ENTER" to select
ENB ↔ DIS

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Use or non-use of DEADMAN switch is set up.

Set-up value	Description
ENB	DEADMAN Switch function is used
DIS	DEADMAN Switch function is not used.

MODEL setup

1. Setup Sequence

Step 1. Move to MODEL screen

```
<HW CONF-T/P>
         T/Pendant TYPE
1: EMG TYPE
2: DEADMAN S/W
3: MODEL


                    group #
```

3 S

Select 3 : MODEL

Step 2. Set up MODEL

```
<T/P-MODEL>
           T/P MODEL
TYPE :   TPM
```

ENTER

"ENTER" to select

TP6 ↔ TPM

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Teach Pendant Model is set up.

Set-up value	Description
TPM	TPM-9000 Model is used
TP6	TP6 Model is used

1.3.1.5 SVON --- Auto Servo-On setup

1. Setup Sequence

Step 1. **Move to SVON screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O

                  item #
```

5 M

Open HW CONF(0) screen

Select 5: SVON

Step 2. **Set up SVON**

```
<HW CONF-AUTO SVON>
      AUTO SERVO ON
MASK :  ENB
```

ENTER

SVON setup item

"ENTER" key is used to change the value to the desired value.

```
<HW CONF-AUTO SVON>
     AUTO SVON DISABLE
MASK :  DIS
```

Change SVON setup

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Use or non-use of Auto Servo ON is setup.

Set-up value	AUTO SERVO ON	Description
ENB	Used	At the time requiring Servo On, such as Run, Jog, etc., Servo On is enabled automatically.
DIS	Not used	A user can turn on/off Servo by pressing Servo On key.

1.3.1.6 A I/O --- Analog Input/Output setup

1. Setup Sequence

Step 1. **Move to A I/O screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O

                 item #
```

6
N

Open HW CONF(0) screen

Select 6: A I/O

Step 2. **Set up A I/O**

```
<HW CONF-A I/O>
      Analog I/O B/D CNT
          VALUE = 0
```

Set up Analog I/O Board

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Analog I/O Board Output type is selected.

Set-up value	Description
0	Analog I/O function is not used.
1	When Analog I/O is used, all 4 Ports are used as a user I/O.
2	When Analog I/O is used, the last Port is used as the linear velocity output, and the remaining 3 Ports are used as a user Output.

CAUTION

▶ Function of Analog I/O Board will be implemented later.

▶ Setup details of Analog I/O is subjected to change.

▶ When using the functions, refer to Analog I/O Card Manual.

1.3.1.7 BRAM --- Back Up Ram Initialization

1. Setup Sequence

Step 1. **Move to BRAM screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O

                item #
```

[Pg Dn]

Open HW CONF(0) screen

Select Pg Dn

```
<PUBLIC-HW CONF(1)>
1: BRAM         2: MAX CH

                item #
```

[1 Q]

Open HW CONF(1) screen

Select 1: BRAM

Step 2. **BRAM initialization**

```
<HW CONF-BRAM>
      BACK-UP RAM INIT
1: JOB INIT
2: GPNT INIT
3: GINT INIT
4: GFLOAT INIT

                input #
```

[1 Q] [4 L]

Perform initialization of the relevant item

1 : JOB, PNT initialization

2 : GLOBAL POINT initialization

3 : GLOBAL INT initialization

4 : GLOBAL FLOAT initialization

Step 3. **JOB INIT screen**

```
Press ENTER Key to
Initialize JOB

                ESC : EXIT
```

[1 Q]

JOB initialization screen

Important

▶ Pay attention that when in BRAM initialization, the relevant data is deleted.

GPNT INIT screen

Press ENTER Key to Initialize G_POINT_V ESC : EXIT

2 / R

GLOBAL POINT initialization screen

GINT INIT screen

Press ENTER Key to Initialize G_INT_V ESC : EXIT

3 / S

GLOBAL INT initialization screen

GFNT INIT screen

Press ENTER Key to Initialize G_FLOAT_V ESC : EXIT

4 / L

GLOBAL FLOAT initialization screen

Step 4. **Perform initialization**

Initialized! ESC : EXIT

ENTER

Select ENTER to perform initialization

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

CAUTION ▶ Before BRAM initialization, back up the data through Unihost. ▶ Be cautious that after BRAM initialization, the relevant data cannot be recovered.

1.3.1.8 MAX CH --- Max Channel Selection

1. Setup Sequence

Step 1. **Move to MAX CH screen**

```
<PUBLIC-HW CONF(0)>
1: TMR          2: COMM
3: I/O          4: TP
5: SVON         6: A I/O

                item #
```

Pg Dn

Open HW CONF(0) screen

Press Pg Dn

```
<PUBLIC-HW CONF(1)>
1: BRAM         2: MAX CH

                item #
```

2 R

Open HW CONF(1) screen

Select 2: MAX CH

Step 2. **Select MAX CH**

```
<MAX_CH>
    MAX CHANNEL USE

     VALUE = 1
```

1 Q to 3 S

Set the maximum available number of channel

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ DEFAULT MAX CHANNEL is "1" channel, and up to maximum "3" channels can be used.

▷ No. 3 channel is for BGT only, which can drive, using ports, Background JOB.

1.3.2 **PALLET (Only For SCARA or XY robot Type)**

1. Items to be set up

PALLET	Description	Range	Set-up value
1 : DATA	Number of workpieces in a pallet	1 to 99	
2 : CNT	Start point of workpieces in a pallet	1 to 99	
3 : OUT	Formation of output after all palletizing works are completed	See the explanation	
4 : PATTERN	Palletizing pattern	0 to 3	

2. Screen Configuration

Step 1. Move to PALLET screen

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                SELECT #
```

Key: 4 L

Open intial menu screen

Select 4: PARA

```
<PARAMETER>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST


SEL  INFO  PUB  EXIT
```

Key: ▲ ▼

Select a robot by using "robot select " ▲▼ key.

```
<PARAMETER>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST


SEL  INFO  PUB  EXIT
```

Key: F3

Open PUBLIC PARAMETER group screen

```
<PUBLIC PARAMETER>
1: HW CONF      2: PALLET
3: PLC          4: ETC



                group #
```

Key: 2 R

Select 2: PALLET

Step 2. **Set up PALLET NO**

```
<PUB-PALLET >
            Pallet NO=010

  F1    : Input NO.
  ENT   : Pallet    Set
  ESC   : EXIT

NO
```

F1

Select NO

```
<PUB-PALLET >

  Pallet NO= 10
```

1 Q | 0 V | ENTER

Select the pallet number to be set. (E.g., No. 10)

```
<PUB-PALLET >
            Pallet NO=010

 Press ESC to exit


DATA   CNT   OUT   PATTERN
```

ESC

PALLET enter screen

```
<PUB-PALLET >
            Pallet NO=010

  F1    : Input NO.
  ENT   : Pallet    Set
  ESC   : Exit

NO
```

PALLET NO setup screen

CAUTION

▶ Before entering "PALLET DATA", be sure to perform initialization (one time).

■ Procedure of pallet data initialization

Step 1. **Move to PALLET screen**

```
<PUBLIC PARAMETER>
1: HW CONF      2: PALLET
3. PLC          4: ETC



                  group #
```

2 / **R**

Open PUBLIC PARAMETER group screen

Select 2: PALLET

Step 2. **PALLET initialization**

```
<PUB-PALLET >
  Pallet NO=000

  F1    : Input NO.
  ENT   : Pallet Set
  ESC   : Exit

NO            ___
```

F4

PALLET setup DATA initialization

```
Press ENTER Key to
Initialize PALLET



                  ESC : EXIT
```

ENTER

Perform initialization

Step 3. **Initialization completed**

```
       INITIALIZED!!



                  ESC : EXIT
```

Initialization completed

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

DATA --- Pallet workpiece number setup

1. How to enter the data

Step 1. **Move to PALLET screen**

```
<PUB-PALLET>
  Pallet NO=09

 Press ESC to exit


DATA  CNT  OUT  PATTERN
```

F1

On PALLET NO.9 screen, select "DATA"

Step 2. **Enter DATA**

```
<PALLET-DATA>
  Pallet NO=09
1.XWM= 5
2.YWM=01
3.ZWM=01

DATA  CNT  OUT  PATTERN
```

5 M | ENTER

Enter X value (e.g., 5)

Step 3.

```
<PALLET-DATA>
  Pallet NO=09
1. XWM= 5
2. YWM=01
3. ZWM=01

Update OK? (ENT/ESC)
```

ESC

Select ESC

```
<PALLET-DATA>
  Pallet NO=09
1. XWM= 5
2. YWM=01
3. ZWM=01

Update OK? (ENT/ESC)
```

ENTER

Select ENTER

```
<PUB-PALLET >
  Pallet NO=09

 Press ESC to exit


DATA  CNT  OUT  PATTERN
```

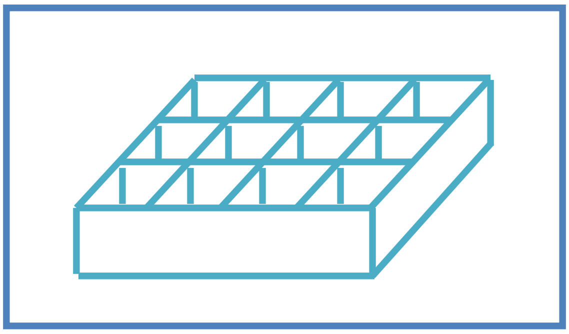
Updating completed

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

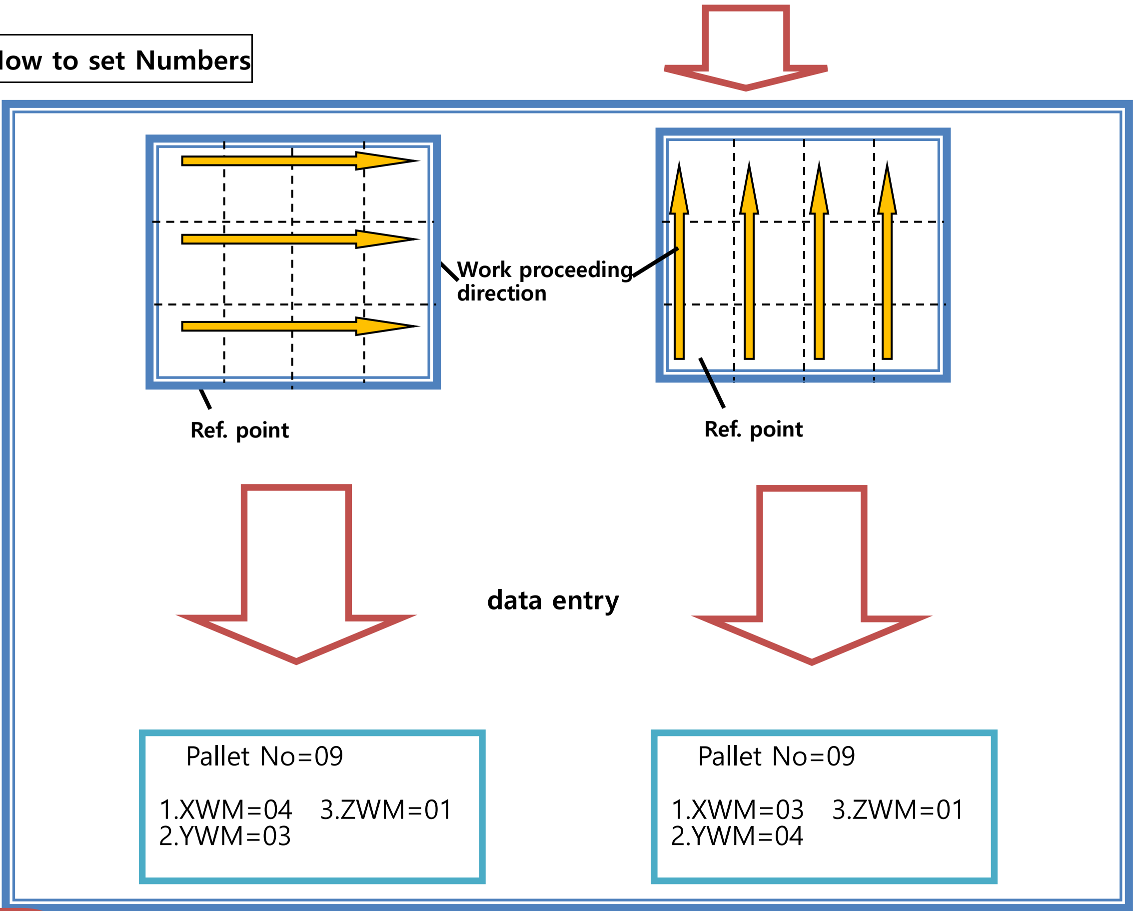
■ Comment

▷ How to set up data

Set up the work proceeding direction by XWM in reference to the reference point.



How to set Numbers



Important

1) The number of workpiece may be entered so that it is not consistent with the axis directionof the mechanism.
2) Set up the number according to PMOV work direction
3) Pay attention to POINT Teaching. (Refer to Program Manual, “PMOV commands”.

CNT --- Setup of Start point of pallet workpiece

Although power is turned off, the next execution position is memorized.

Step 1. **Move to PALLET screen**

```
<PUB-PALLET >
  Pallet NO=09

 Press ESC to exit


DATA  CNT  OUT  PATTERN
```

F2

① On PALLET NO.9 screen, select "CNT"

Step 2. **Enter CNT**

```
<PALLET-CNT>
  Pallet NO=09
1. Xcnt= 2
2. Ycnt=01
3. Zcnt=01


DATA  CNT  OUT  PATTERN
```

2 R ENTER

Enter Xvalue (e.g., work for row 2)

Step 3. **Store CNT**

```
<PALLET-CNT>
  Pallet NO=09
1. Xcnt= 2
2. Ycnt=01
3. Zcnt=01


Update OK? (ENT/ESC)
```

ESC

Select ESC

```
<PALLET-CNT>
  Pallet NO=09
1. Xcnt= 2
2. Ycnt=01
3. Zcnt=01


Update OK? (ENT/ESC)
```

ENTER

Select ENTER

```
<PUB-PALLET >
  Pallet NO=009

 Press ESC to exit


DATA   CNT   OUT   PATTERN
```

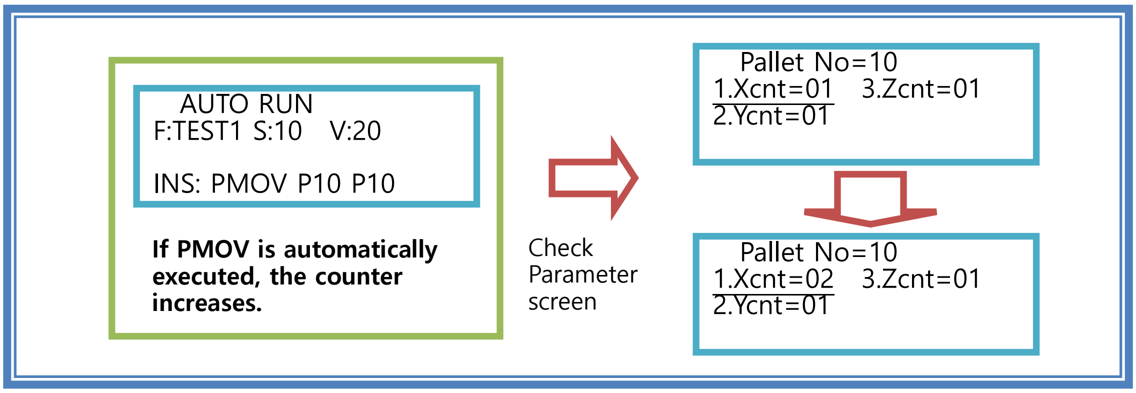
▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

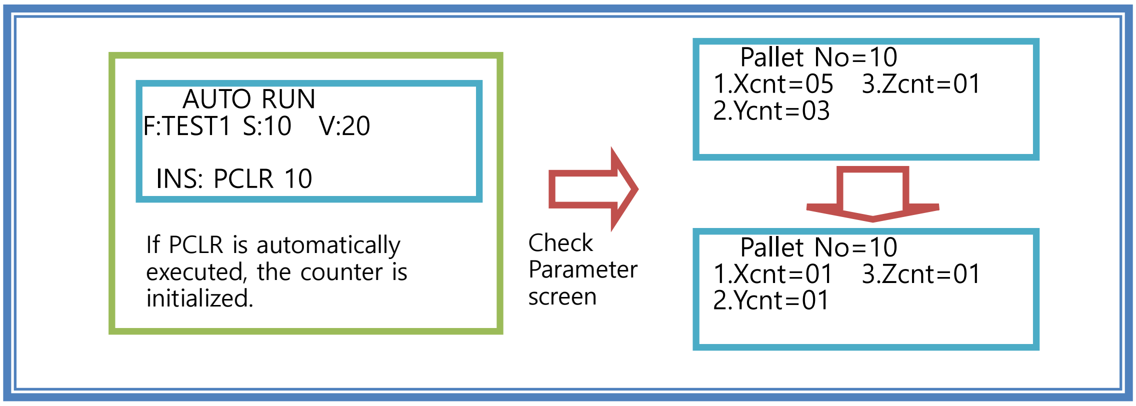
▷ This is the position to be moved in a pallet when PMOV is performed in Program execution.

▷ To start the palletizing from the first → enter "1" all.

▷ CNT value is automatically increased after PMOV is performed during Job Program execution.



▷ If PCLR command is executed during Job Program execution, all CNT values are initialized. (becomes "1" all)



▷ After the work has been completed one time, the CNT value is initialized. (to "1").

OUT --- Setup of formation of output when palletizing is completed

1. Setup Sequence

Step 1. **Move to PALLET screen**

```
<PUB-PALLET>
  Pallet NO=09

 Press ESC to exit

DATA   CNT   OUT   PATTERN
```

F3

On PALLET NO.9 screen, select "OUT"

Step 2. **Enter OUT**

```
<PALLET-OUT>
  Pallet NO=09
1. Port= 01
2. Time=500
3. Type=04

DATA   CNT   OUT   PATTERN
```

① Refer to comment

Step 3. **Store OUT**

```
<PALLET-OUT>
  Pallet NO=09
1. Port= 01
2. Time=500
3. Type=04

Update OK? (ENT/ESC)
```

ESC

Select ESC

```
<PALLET-OUT>
  Pallet NO=09
1. Port= 01
2. Time=500
3. Type=04

Update OK? (ENT/ESC)
```

ENTER

Select ENTER

```
<PUB-PALLET >
  Pallet NO=009

 Press ESC to exit

DATA   CNT   OUT   PATTERN
```

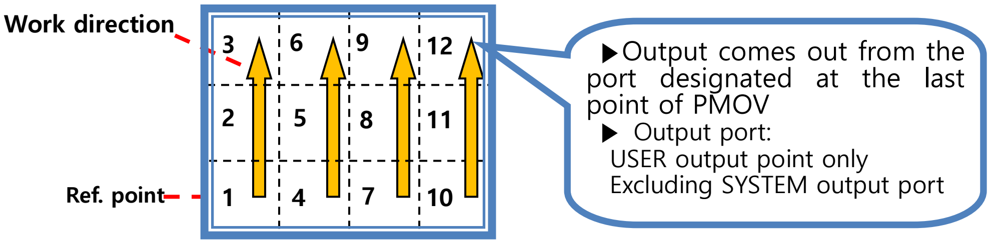
Updating completed

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ **1.Port**---- Output comes out from the port designated at the last point of PMOV.

Output port	
OUT 0 to 15	User output
OUT 16 to 79	Extended output (option)

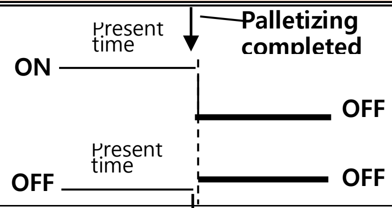
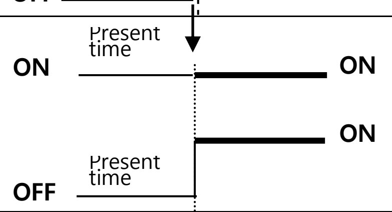
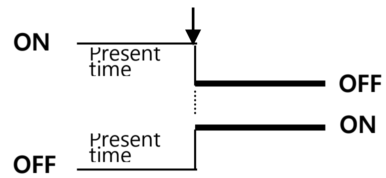
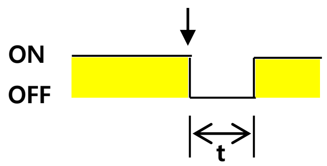
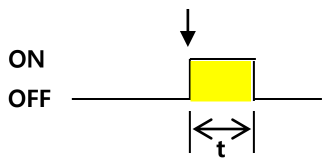
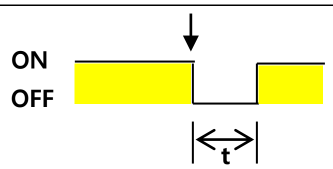
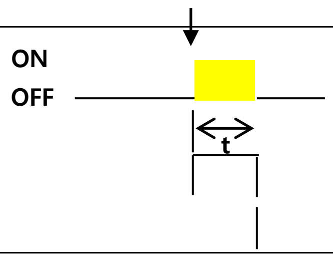


▷ **2.Time** ---- Pulse ON time when the output type is a pulse output.

TIME	
Unit	10 msec
Setup range	0 to 999

▷ **3.TYPE** --- Setup of completion signal output type after the palletizing has been completed.

Types of output

Setup No	Output type	Meaning
0		▶ Turned off regardless of the output state of the present time. ▶ Data is set up for the Port only.
1		▶ Turned on regardless of the output state of the present time. ▶ Data is set up for the Port only.
2		▶ The output of the resent time is inverted. ▶ Data is set up for the Port only.
3		▶ Off-pulse is output for the time set up in Time(ms). ▶ In the case of 2-stack loading in Z axis, the signal is output only after the second stack work has been completed.
4		▶ On-pulse is output for the time set up in Time(ms). ▶ In the case of 2-stack loading in Z axis, the signal is output only after the second stack work has been completed.
5		▶ Off-pulse is output for the time set up in Time(ms). ▶ In the case of 2-stack loading in Z axis, one signal is output after the first stack work has been completed, and one signal is output after the second stack work has been completed, respectively.
6		▶ On-pulse is output for the time set up in Time(ms). ▶ In the case of 2-stack loading in Z axis, one signal is output after the first stack work has been completed, and one signal is output after the second stack work has been completed, respectively.

PATTERN --- Palletizing pattern setup

1. Setup Sequence

Step 1. **Move to PALLET screen**

```
<PUB-PALLET>
  Pallet NO=09

 Press ESC to exit


DATA   CNT   OUT   PATTERN
```

F4

On PALLET NO.9 screen, select "PATTERN"

Step 2. **Enter PATTERN**

```
<PALLET-PATTERN>
          Pallet NO=09
             Pattern
             NO=00


DATA   CNT   OUT   PATTERN
```

① Refer to Comment

Step 3. **Store PATTERN**

```
<PALLET-PATTERN>
          Pallet NO=09
             Pattern
             NO=00


Update OK? (ENT/ESC)
```

ESC

Select ESC

```
<PALLET-PATTERN>
          Pallet NO=09
             Pattern
             NO=00


Update OK? (ENT/ESC)
```

ENTER

Select ENTER

```
<PUB-PALLET >
  Pallet NO=009

 Press ESC to exit


DATA   CNT   OUT   PATTERN
```

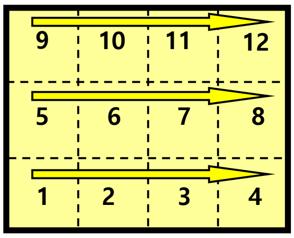
Updating completed

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

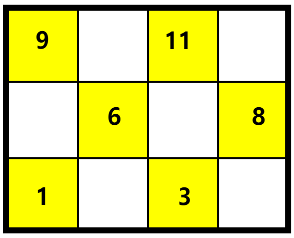
■ Comment

▷ ① Palletizing pattern has 4 kinds (NO. 0 to 3).

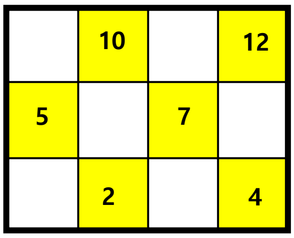
Pattern NO.0



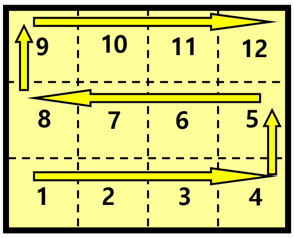
Pattern NO.1



Pattern NO.2



Pattern NO.3



1.3.3 PLC

1. Items to be set up

PLC	Description	Range	Set-up value
1 : PNO	Set up the robot channel to execute PLC.	1 to 3	1
	Set up the job number of PLC to perform Auto Run.	0 to 9	0
2 : AUTO	Select Auto Run.	YES, NO	NO
3 : IO ENB	Designate I/O Port used in PLC.	YES, NO	NO
4 : BCD	Select the use of BCD UNIT	USED, NONE	NONE
5 : OCL	CLEAR the output port.	USED, NONE	NONE

2. Screen Configuration

Step 1. **Move to PLC screen**

```
<PUBLIC PARAMETER>
1: HW CONF        2: PALLET
3: PLC            4: ETC



                  group #
```

3
S

Open PUBLIC PARAMETER screen

Select 3: PLC

```
<HWCONF-PLC>
        GROUP : PLC
1: PNO            2: AUTO
3: IO ENB         4: BCD
5: OCL

                  item #
```

PLC setup screen

For further information, refer the descriptions of each parameter or Chapter 6. Simple PLC functions.

1.3.3.1 PNO --- PLC Job Number setup

1. Setup Sequence

Step 1. Move to PNO screen

```
<HWCONF-PLC>
  GROUP : PLC
1: PNO           2: AUTO
3. IO ENB        4: BCD
5: OCL

                     item #
```

1
Q

Open PLC screen

Select 1: PNO

Step 2. Set up PNO

```
<PLC-PNO>
        PLC JOB NUMBER

CHANNEL= 1
JOB NUM = 0
```

① Select PLC operation channel

② Set up PLC JOB NUMBER

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ ① The channel in which PLC JOB is saved is selected.

▷ ② PLC Job Number is set up.

- PLC Job Number to be run simultaneously with powering on is set up.
- One of ten, 0 to 9, is selected.

1.3.3.2 AUTO --- Auto Run setup

1. Setup Sequence

Step 1. Move to AUTO screen

```
<HWCONF-PLC>
   GROUP : PLC
1: PNO              2: AUTO
3: IO ENB           4: BCD
5: OCL

                              item #
```

2 R

Open PLC screen

Select 2: AUTO

Step 2. Set up AUTO

```
<PLC-AUTO>
         AUTO RUN : NO
1: YES              2: NO

                              input #
```

1 Q or 2 R

PLC AUTO RUN setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ AUTO RUN setup

- Whether the PLC JOB set up in PNO is to be performed simultaneously with powering on is set up.

AUTO	
YES	PLC is automatically run when powering on.
NO	Auto run is not performed.

1.3.3.3 ENB --- PLC Used I/O Port setup

1. Setup Sequence

Step 1. **Move to ENB screen**

```
<HWCONF-PLC>
  GROUP : PLC
1: PNO              2: AUTO
3: IO ENB           4: BCD
5: OCL

                      item #
```

3
S

Open PLC screen

Select 3: ENB

Step 2. **Set up ENB**

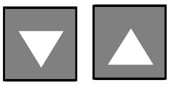
```
<PLC-ENB>
        PLC I/O ENABLE
OP1 : NO            OP2 :-
OP3 :-              OP4 :-
OP5 :-              OP6 :-
```

PLC Used I/O Port setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ Parameter for setting PLC Used I/O Port.

▷ Item is moved : 

▷ Set-up value is changed: ENTER

- The above screen varies depending on the number of I/O Board.
- A screen that the number of I/O Board is set to 1.
- I/O Port that PLC Job uses is set up. The I/O Port that is not set to YES cannot input/output in PLC Job.

1.3.3.4 BCD --- BCD UNIT setup

1. Setup Sequence

Step 1. **Move to BCD screen**

```
<HWCONF-PLC>
   GROUP : PLC
1: PNO              2: AUTO
3: IO ENB           4: BCD
5: OCL

                              item #
```

4
L

Open PLC screen

Select 4: BCD

Step 2. **Set up BCD**

```
<PLC-BCD>
            BCD SELECT

     VALUE = USED
```

ENTER

BCD setup screen

```
<PLC-BCD>
            BCD SELECT

     VALUE = NONE
```

① Using “ENTER” key, change the set-up value.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Parameter for setup of JOB number entry type

- If VALUE is 'NONE' : Using System input port PROG0(LSB) to PROG4(MSB), 32 Job numbers (P00 to P31) can be selected, and after the port setting, if PROG_SEL is entered, the controller recognizes the relevant Job number.

- If VALUE is 'USED' : Using User input port IN0 to IN7, Job numbers P00 to P99 can be selected, and PROG_SEL signal is not used.

Number of place of '10' digit				Number of place of '1' digit			
IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0
'0' to '9'				'0' to '9'			

- If VALUE is 'USED' : Each of (IN0, IN4) is LSB, and (IN3, IN7) is MSB.
- If VALUE is 'USED' : when the port is set up to '9' or higher, the controller recognizes it as '9'. That is, all the ports of each digit [1001] to [1111] are recognizes as '9'.

BCD		
item	NONE	USED
INPUT	SYSTEM INPUT	USER INPUT
PORT	PROG0 to PROG4	IN0 to IN7
PROG_SEL	Signal is used	Signal is not used
JOB	32 Job numbers are used (P00 to P31)	100 Job numbers are used (P00 to P99)
LSB	PRG0	IN0, IN4
MSB	PROG4	IN3, IN7
ETC		Ports [1001] to [1111] are all recognized as '9'.

Important

▶ In the case that BCD set-up value is "NONE", in order to recognize the controller JOB number, use PROG_SEL signal after the port input.

1.3.3.5 OCL --- OUTPUT CLEAR setup

1. Setup Sequence

Step 1. **Move to OCL screen**

```
<HWCONF-PLC>
  GROUP : PLC
1: PNO          2: AUTO
3: IO ENB       4: BCD
5: OCL

                    item #
```

5 M

Open PLC screen

Select 5: OCL

Step 2. **Set up OCL**

```
<PLC-OCL>
        OUTPUT CLEAR

        VALUE =USED
```

ENTER

① PLC OCL setup screen

Using “ENTER” key, change to the value to besset up.

```
<PLC-OCL>
        OUTPUT CLEAR

        VALUE =NONE
```

Change PLC OCL setup.

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Parameter for setup of user output port clearing function when in alarm.

VALUE	Port status when in alarm.
NONE	Port is retained in the current status.
USED	User output port (User I/O output, Extended I/O output) is cleared to '0'.

CAUTION

▶ The function of port clearing is restricted to the port involving in JOB, but the port involving in the internal Simple PLC is not cleared even though the alarm occurs.

1.3.4 **ETC**

1. Items to be set up

HW CONF	Description	Range	Set-up value
1:W.TIME	WORK TIMER operation time	See the explanation	
2:CINF	Controller information	See the explanation	
3:PASSWORD	PASSWORD setup	See the explanation	
4:NAME	Controller name setup	See the explanation	
5:BRK PASS	Break Password is used	See the explanation	

2. Screen Configuration

Step 1. **Move to PARA**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                  SELECT #
```

4
L

Open intial menu screen

Select 4: PARA

Step 2. **Select ROBOT**

```
<JOB MAIN>
 NO             TYPE
*CH1            RGA80B
 CH2            RGAA0B


SEL     INFO     PUB     EXIT
```

Select a robot by using "robot select " ▲▼ key.

Step 3. **Move to PUBLIC PARAMETER**

```
<PARAMETER>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST


SEL     INFO     PUB     EXIT
```

F3

Open PUBLIC PARAMETER group screen

Step 4. **PUBLIC PARAMETER**

```
<PUBLIC PARAMETER>
1: HW CONF       2: PALLET
3: PLC           4: ETC



                    group #
```

4
L

Select 4: ETC

Step 5. **ETC screen**

```
<PUBLIC-ETC>
1: W.TIMER       2: CINF
3: PASS          4: NAME
5: BRK PASS


                    item #
```

ETC screen

1.3.4.1 W.TIMER --- WORK TIMER operating time

1. Setup Sequence

Step 1. **Open ETC**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS

item #

Open ETC screen

Step 2. **Move to W.TIMER**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS

item #

1
Q

Select 1: W.TIMER

Step 3. **W.TIMER setup screen**

<ETC-W.TIME>
Working Time
17D
20H 52M 20S

W.TIMER setup screen

■ Comment

▷ While the robot controller is being supplied with the power, WORK TIMER operates.

▷ The time that the power is supplied up to now is displayed, and for example '17D 20H 52M 20S' means the use of 17 days, 20 hours, 52 minutes, and 20 seconds.

1.3.4.2 CINF --- controller information

1. Setup Sequence

Step 1. **ETC Open**

```
<PUBLIC-ETC>
1: W.TIMER        2: CINF
3: PASS           4: NAME
5: BRK PASS

                        item #
```

Open ETC screen

Step 2. **Move to CINF**

```
<PUBLIC-ETC>
1: W.TIMER        2: CINF
3: PASS           4: NAME
5: BRK PASS

                        item #
```

2
R

Select 2: CINF

Step 3. **CINF screen**

```
<ETC-CINF>
Model : N1 – 111111 –U
1) 106   2) 106    3) 106
4) 106   5) 106    6) 106
Version : 02.00.ST
```

CINF screen

■ Comment

▷ Information on the motor capacity

- N1 – **111111** – U : Denotes the information on the capacity of the currently used motor

▷ Program version

- **02.00.ST** : Denotes the version of the currently operated software

1.3.4.3 PASSWORD --- PASSWORD Change Mode

1. Screen Configuration

Step 1. **ETC screen**

```
<PUBLIC-ETC>
1: W.TIMER       2: CINF
3: PASS          4: NAME
5: BRK PASS

                 item #
```

Open ETC screen

Step 2. **Move to PASS screen**

```
<PUBLIC-ETC>
1: W.TIMER       2: CINF
3: PASS          4: NAME
5: BRK PASS

                 item #
```

Select 3: PASS

3
S

```
<ETC-PASS>
  PASSWORD SET
1: MODE
2: VALUE

                 group #
```

PASSWORD Parameter item

MODE setup

1. Setup Sequence

Step 1. **Move to MODE screen**

```
<ETC-PASS>
  PASSWORD SET
1: MODE
2: VALUE


                    group #
```

1 Q

Select 1: MODE

Step 2. **Password setup screen**

```
<PASS-MODE>
       PASSWORD MODE
MODE: NONE
```

ENTER

Change by “ENTER” key

NONE<->JOB<->PARA<->ALL

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ PASSWORD range setup item

Item	Set-up value	Description
MODE	NONE	PASSWORD is not used.
	JOB	PASSWORD setup for JOB item
	PARA	PASSWORD setup for PARAMETER item
	ALL	PASSWORD setup for all controller items

VALUE setup

1. Setup Sequence

Step 1. **Move to MODE screen**

```
<ETC-PASS>
  PASSWORD SET
1: MODE
2: VALUE

                    group #
```

2
R

Select 2: VALUE

Step 2. **Password change screen**

```
<PASS-PASS>
        Change password

           VALUE = 0
```

PASSWORD change screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ PASSWORD change

※ Initial PASSWORD is set to "0000".

CAUTION	▶ If PASSWORD is lost, contact the Customer Support Team of Robostar.

1.3.4.4 NAME --- controller name setup

1. Setup Sequence

Step 1. **Open ETC**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS
item #

Open ETC screen

Step 2. **Move to NAME**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS
item #

① Select 4: NAME

4 L

Step 3. **Set up & update NAME**

<ETC-NAME>
Controller Name Set
CUR :
NEW : TEST
SAVE

Enter controller name (Enter "TEST")

← T | SVON E | 3 S | ← T | ENTER

<ETC-NAME>
Controller Name Set
CUR :
NEW : TEST
SAVE

SAVE

F4

<ETC-NAME>
Controller Name Set
CUR :
NEW : TEST
Update OK? (ENT/ESC)

Select ENTER

ENTER

```
<PUBLIC-ETC>
1: W.TIMER        2: CINF
3: PASS           4: NAME
5: BRK PASS

                  item #
```

CONTROLLER name change is completed

Step 4. **Check the changed details**

```
<PUBLIC-ETC>
1: W.TIMER        2: CINF
3: PASS           4: NAME
5: BRK PASS

                  item #
```

Select 4: NAME

4
L

```
<ETC-NAME>
Controller name set

CUR : TEST
NEW :

                SAVE
```

Check "CUR: TEST"

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ NAME change

※ PARAMETER that changes the name of controller.

Important

▶ When connecting to Unihost, the changed controller name can be confirmed, and this is useful for controller data backup through the host.

1.3.4.5 BRK PASS --- Brake Password setup

1. Setup Sequence

Step 1. **Open ETC**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS

item #

Open ETC screen

Step 2. **Move to BRK PASS**

<PUBLIC-ETC>
1: W.TIMER 2: CINF
3: PASS 4: NAME
5: BRK PASS

item #

5
M

① Select 5: BRK PASS

Step 3. **Brake Password setup screen**

<ETC-BRK_PASS>
Change password

VALUE = 0

BRK PASSWORD change screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Explanation

※ To prevent Brake Off due to user’s mistake when in performance of Brake On/Off, Brake On/Off operation has to be performed after checking the password.

CAUTION

▶ The function of Brake Password is supported only for Transfer Robot type.

▶ If TML__, TMS__, CML__, ROSEP300 robot is selected, Brake Password is enabled.

Chapter 2 ORG(Origin) Mode

☞ Origin searching function may be firstly performed only one time when the controller is powered on.

☞ Sequence of the Origin searching may be determined by well taking into account the peripheral hindrances.

☞ In SCARA robot, after the Origin searching sequence has been changed and if the operation of linear interpolation or palletizing is not precise, please check the parameter offset value.

2.1 Overview

■ What is Origin searching?

Making the reference points of robot mechanism sensors, Z-phase of an encoder, and a controller counter be coincident with one another.

■ After the power switch is turned on, the origin searching must be performed.

- All of the robots adopting an incremental encoder must set the origin that a controller can recognize after the power is supplied thereto, because the controller cannot know the current position of the robot.

-Once the Origin searching has been accomplished, the origin remains until the power switch is turned off.

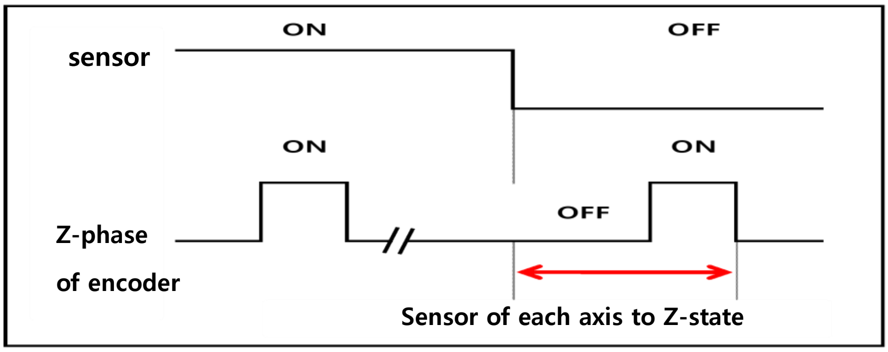
■ Orientation and sequence of and a sensor for Origin searching are set up in "5.ORIGIN / SET".

■ If the origin searching is performed in H/W Limit state, robot axes move in the direction that the H/W Limit state is released, and then, if all of the axes are not in the H/W Limit state, the origin searching is started.

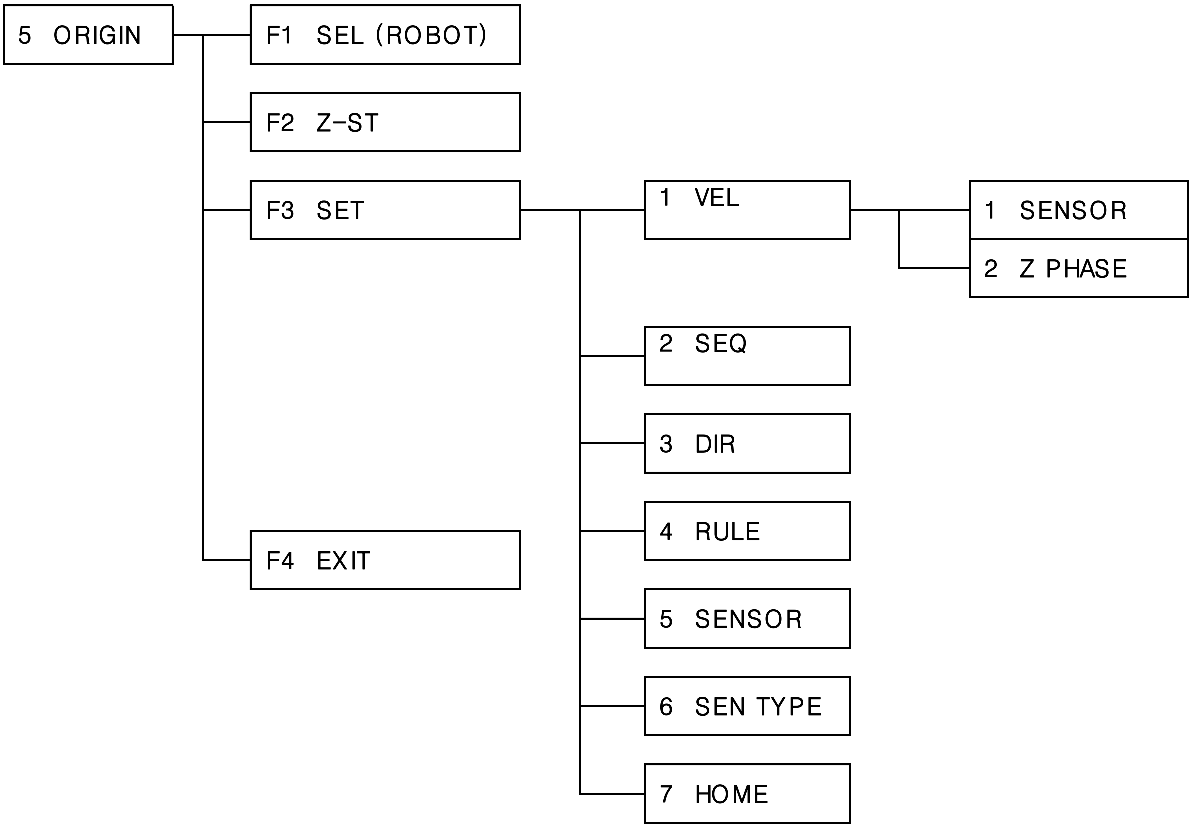
■ If Origin searching is re-performed after Origin searching has been completed, a system moves to the previous Origin position position first, and then Origin searching is again executed.

■ Sequence of moving to the previous Origin position observes the sequence of Origin searching of Origin Parameters, and the moving velocity can be set up through SENSOR SEARCH VEL and Z SEARCH VEL Parameter items.

■ Relationship between Origin sensor and ENCORDER Z-phase



2.1.1 ORIGIN Parameter Flowchart



2.1.2 ORG

1. Items to be set up

HW CONF	Description	Range	Set-up value
1:VEL	SENSOR: Velocity rate when Origin sensor moves	1 to 100	
	Z. PHASE: Velocity rate for Z-phase moving	1 to 100	
2:SEQ	Sequence of Origin searching	1, 2, 3, 4	
3:DIR	Direction of moving to Origin	+, -	
4:RULE	Setting how to search Origin	See the explanation	
5:SENSOR	Origin sensor	CW, CCW, ORG	
6:SEN TYPE	Types of Origin sensor	NO, NC	*NC TYPE is recommended
7:HOME	Origin coordinate value	See the explanation	

2. Screen Configuration

Step 1. **Move to PARA**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                SELECT #
```

5
M

Select 5: ORIGIN

Step 2. **Select ROBOT**

```
<ORIGIN MODE>
 NO    TYPE    RULE    STE
 * 1   XYZW    NONE    ---
   2   XY__    NONE    ---


SEL    Z-ST    SET     EXIT
```

▲
▼

Select a robot by using "robot select " ▲▼ key.

Step 3. **Move to ORIGIN setup screen**

```
<ORIGIN MODE>
 NO    TYPE    RULE    STE
 * 1   XYZW    NONE    ---
   2   XY__    NONE    ---


SEL    Z-ST    SET     EXIT
```

F3

Select Robot and then Set up

```
<XYZW : ORG>
 1: VEL          2: SEQ
 3: DIR          4: RULE
 5: SENSOR       6: SEN TYPE
 7: HOME

                   item #
```

ORIGIN setup screen

▶ Screen of a teaching pendant after the origin searching has been completed:

1) The number of pulses from an origin sensor to a Z-phase of an encoder is displayed as follows.

 ▷ When the resolution of an encoder is 2500 pulses (4-multiplication)

 1 ≤ Display range ≤ 10000

 ▷ When the resolution of an encoder is 17 bits (131072)

 1 ≤ Display range ≤ 131072

2) The number of pulses of Z-phase may change within the range of 200 pulses whenever the origin searching is performed.

3) If the variation width of the pulse value is too large after the origin searching has been performed, it is impossible to perform the exact point moving.
Sensors and sensor dogs of the robot mechanism must be checked.

▶ When the origin searching has been completed:

1) ORG has been completed for the whole channels: ORG LED is turned ON
2) ORG has not been completed for the whole channels: ORG LED is turned OFF
3) ORG has not been completed for one of the whole channels : ORG LED blinks

▶ If the origin searching is again performed:

1) ORG LED of a teaching pendant blinks or turned off.
2) "Org" is displayed on a 7-segement at the front of the controller.
3) ORIGIN OK is not output on the system I/O.

2.1.3 VEL --- When in Origin searching, moving velocity rate setup

2.1.3.1 SENSOR --- Origin sensor moving velocity

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

Open ORG screen

Step 2. **Move to VEL screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

1
Q

Select 1: VEL

Step 3. **Set up SENSOR VEL**

```
<XYZW : ORG VEL>
1: SENSOR       2: Z_PHASE

                item #
```

1
Q

Select 1: SENSOR

```
<XYZW : SENSOR>
   SENSOR SEARCH VEL(%)
X: 10           Y : 10
Z: 5            W: 10
```

① Set up moving velocity of Origin sensor

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

2.1.3.2 Z.PHASE --- Moving velocity when in detection of Z-phase

1. Setup Sequence

Step 1. **Open ORIGIN screen**

<XYZW : ORG>
1: VEL 2: SEQ
3: DIR 4: RULE
5: SENSOR 6: SEN TYPE
7: HOME

item #

Open ORG screen

Step 2. **Move to VEL screen**

<XYZW : ORG>
Select 1: VEL 2:
SEQ
3: DIR 4: RULE
5: SENSOR 6: SEN TYPE
7: HOME

1
Q

Select 1: VEL

Step 3. **Set up Z-phase VEL**

<XYZW : ORG VEL>
Select 1: SENSOR 2: Z_PHASE

item #

2
R

② Select 2: Z-phase

<XYZW : Z PHASE>
Z PHASE SEARCH VEL(%)
X: 5 Y : 5
Z: 5 W: 5

Z-phase velocity setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

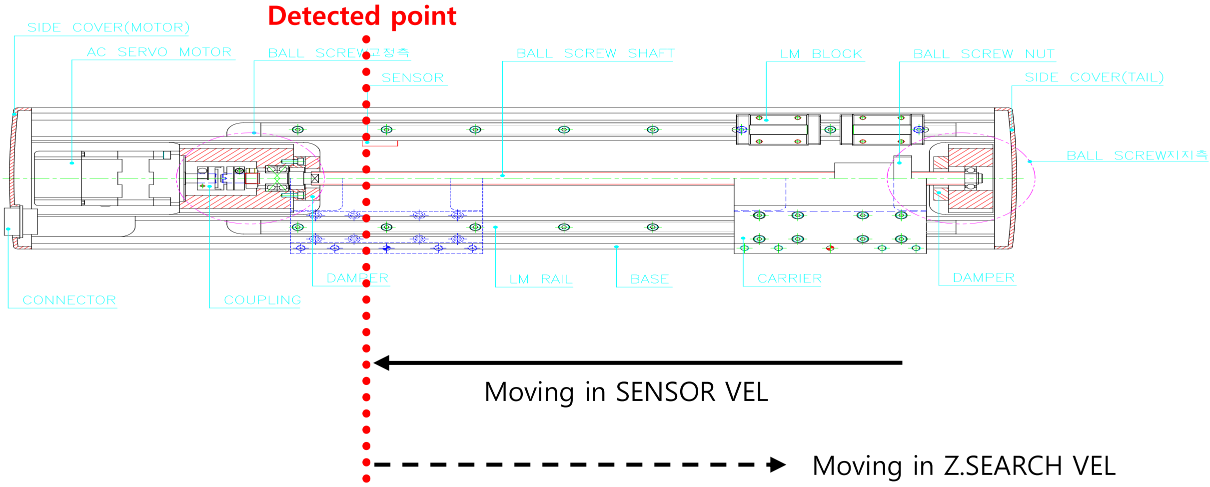
■ Comment

Explanation

▷ SENSOR VEL – Set up the velocity of each axis when in Origin searching

Explanation

▷ Z.SEARCH VEL - Set up the velocity of each axis when Z-phase is performed after the sensor's detection operation



Origin searching velocity(SENSOR VEL & Z. SEARCH VEL)

= Percentage of 500 (rpm) = 500 (rpm) x percentage (%)

CAUTION

- If the Origin searching velocity is too high, "Over Load" & "Limit Error" occur. Therefore, lower the velocity as possible.
- If the controller is for an absolute type, VEL Parameter cannot be accessed.

2.1.4 SEQ --- Setup operating sequence per axis when in Origin searching,

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
Select 1: VEL          2:
SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME
```

Open ORG screen

Step 2. **Move to SEQ screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

2
R

Select 2: SEQ

Step 3. **SEQUENCE setup screen**

```
<XYZW : SEQ>
       ORG SEQUENCE
X: 2            Y : 2
Z: 1            W: 2
```

Set up SEQ (Z-axis is first executed)

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Setup of operating sequence per axis when in Origin searching

- Setup range : 1 to 4

CAUTION	▶ If there are hindrances possible to be crashed within the operation area when in Origin searching, set up that **Z-axis first performs Origin searching**, and then the remaining axes do it, as shown in the above screen.

2.1.5 DIR --- Direction setup when in Origin searching

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

Open ORG screen

Step 2. **Move to DIR screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

3
S

Select 3: DIR

Step 3. **DIR setup screen**

```
<XYZW : DIRECTION>
     ORG DIRECTION
X: +           Y : +
Z: +           W: +
```

Origin direction setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

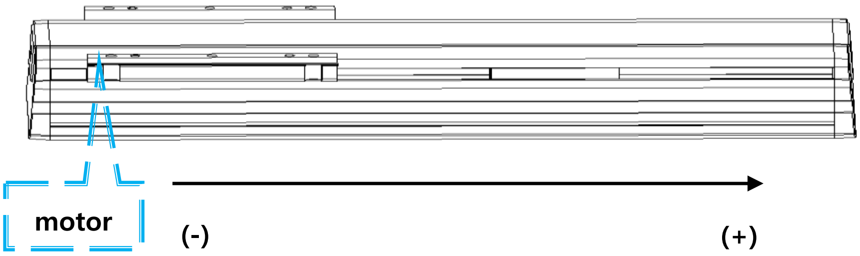
▷ When in Origin searching, the axis moving direction is set up.

CAUTION	▶ If the controller is for an absolute type, DIR Parameter cannot be accessed.

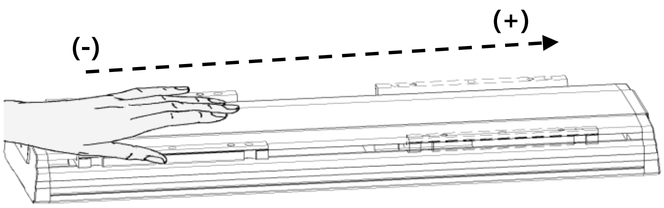
2.1.6 [Supplement] Setup of Origin direction & Moving direction

▶ The following setup method is explained for Robostar Standard for Robot Setup.

▶ In Robostar Standard, after the motor-side is set to Origin and if a point goes farther away from the motor, the coordinate increases (toward +).



Step 1. **MOTION -> DISP sign setup**



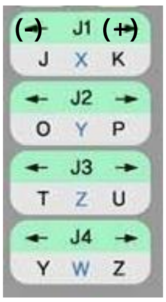

```
XYZW:EDIT>  V :    50
F:TEST     P1 US B   M
X: -89.69             Y : 2.83
Z: 59.72              W: 22.84

EXCH CORD PJUMP FWRD
```

If moving the Robot mechanism with a hand (to +) as shown above and the coordinate displayed on T/P screen does not increase, change the sign of DISP.

```
<XYZW : DISP>
   DISPLAY DIRECTION
X: +              Y : +
Z: +              W: +
```

```
<XYZW : DISP>
   DISPLAY DIRECTION
X: -              Y : +
Z: +              W: +
```

Step 2. **MOTION -> MOV sign setup**

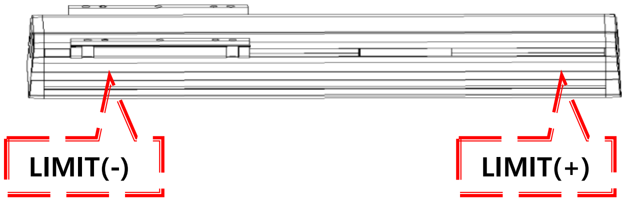


This is set up, if Robot moves in the reverse direction of Jog key on T/P.

```
<XYZW : MOVE>
 MOVING KEY DIRECTION
X: +              Y : +
Z: +              W: +
```

```
<XYZW : MOVE>
 MOVING KEY DIRECTION
X: -              Y : +
Z: +              W: +
```

Step 3. **MOTION->SEN sign setup**



Move Robot to generate H/W Limit alarm. Clear the alarm and move the robot in the same direction of JOG Key motion. If the robot moves or does not escape in the reverse direction, change the sign as PARA(0) → 2: MOTION → 6: SENS sign.

```
<XYZW : SENS>
      SENSOR DIRECITON
X: +               Y: +
Z: +               W: -
```

→

```
<XYZW : SENS>
      SENSOR DIRECITON
X: -               Y: +
Z: +               W: +
```

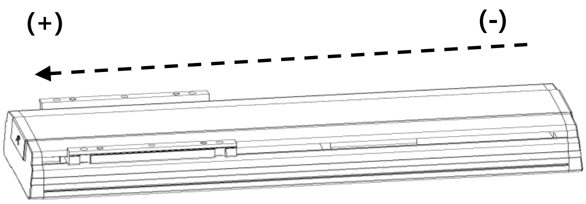
Step 4. **ORIGIN -> DIR setup**

--

```
<MAIN MENU>
1. JOB             2. RUN
3. HOST            4. PARA
5. ORIGIN          6. I/O
7. SYSTEM          8. GPNT
9. INT/FLT         A. ALARM
                    SELECT #
```

```
<XYZW : ORG>
1: VEL             2: SEQ
3: DIR             4: RULE
5: SENSOR          6: SEN TYPE
7: HOME
                     item #
```

Robot's Origin direction is set up.



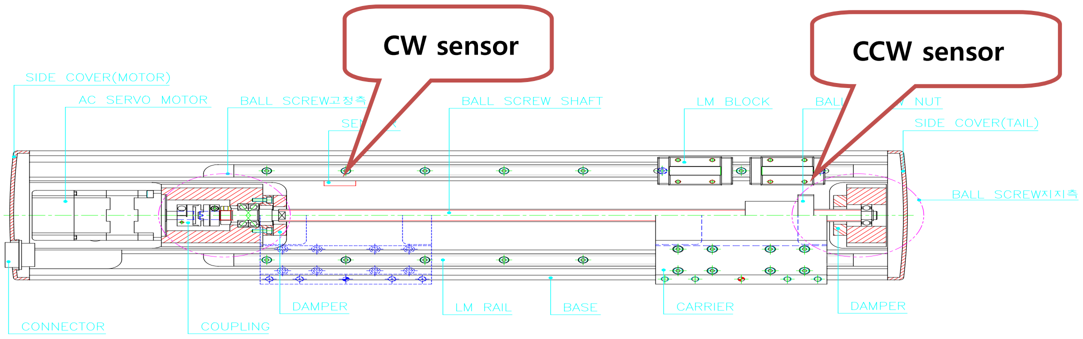
If Origin searching goes in the reverse direction, change the sign.

```
<XYZW : DIRECTION>
      ORG DIRECTION
X: +               Y : +
Z: +               W: +
```

→

```
<XYZW : DIRECTION>
      ORG DIRECTION
X: -               Y : +
Z: +               W: +
```

Step 5. **ORIGIN -> SENSOR setup**



A sensor lying in the direction of Robot Origin is set up.

If, at Origin position, Origin fail or H/W Limit alarm is generated, change the sensor.

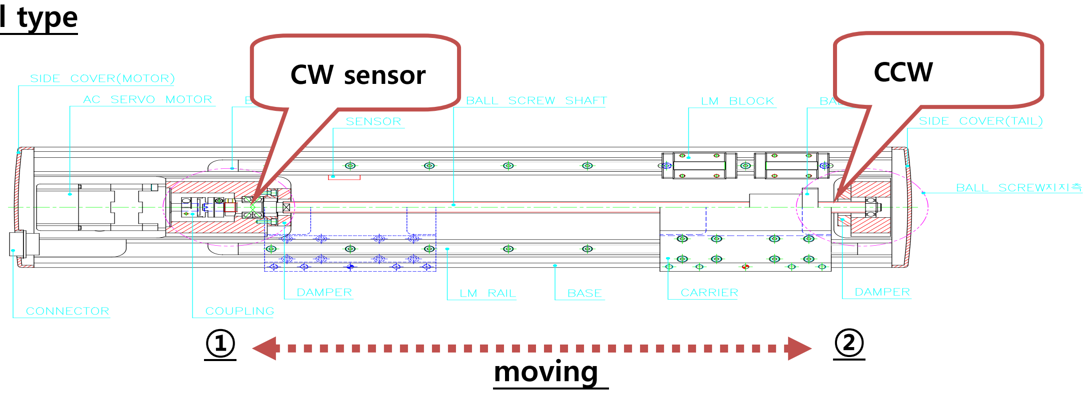
The sign of H/W LIMIT is affected by PARA(0) → 2: MOTION → 6: SENS sign.

<XYZW : SENSOR>
ORIGIN SENSOR
X: CW Y : CW
Z: CW W: CW

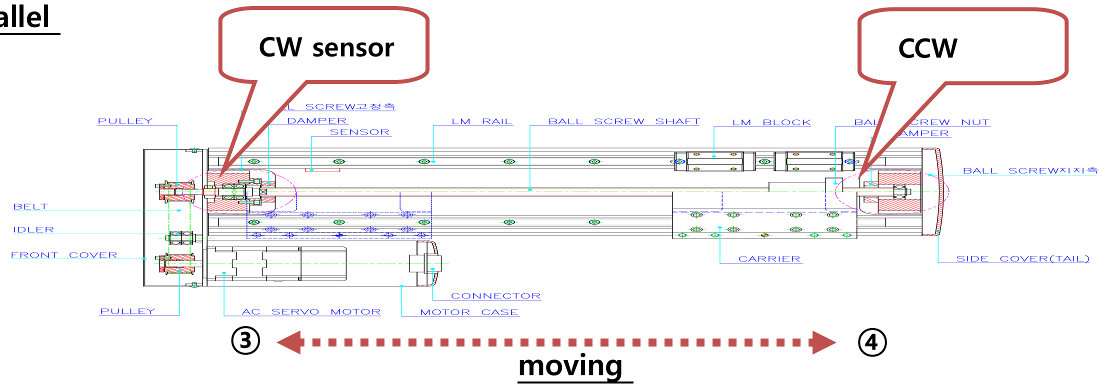
<XYZW : SENSOR>
ORIGIN SENSOR
X: CCW Y : CW
Z: CW W: CW

▶ Cartesian coordinate robot

Serial type



Parallel



* Origin Posture Set-up Value

Origin Position No.	Robot type	Para(0) → 2. MOTION			5. ORIGIN → 6. SEN TYPE	
		4. DISP	5. MOVE	6. SENS	2. DIR Minas motor	6. SEN TYPE
①	Serial type	+	-	+	-	CW
②		-	+	+	-	CCW
①`		+	+	-	-	CCW
②`		-	-	-	-	CW
③	Parallel type	-	-	+	-	CW
④		+	+	+	-	CCW
③`		-	+	-	-	CCW
④`		+	-	-	-	CW

(1) ① to ④ : Standard setting, (2) ①` to ④` : Setup value when in reverse of 5.MOVE value,

(3). 4.DISP setup is independent.

2.1.7 **RULE --- Origin searching rule**

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
 1: VEL            2: SEQ
 3: DIR            4: RULE
 5: SENSOR         6: SEN TYPE
 7: HOME

                    item #
```

Open ORG screen

Step 2. **Move to RULE screen**

```
<XYZW : ORG>
 1: VEL            2: SEQ
 3: DIR            4: RULE
 5: SENSOR         6: SEN TYPE
 7: HOME

                    item #
```

4
L

Select 4: RULE

Step 3. **RULE setup screen**

```
<XYZW : RULE>
         ORIGIN RULE

 RULE : NONE
```

ORIGIN RULE setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ Origin searching rule is set up.

Item	Set-up value	Description
RULE	NONE	Does not move when in Origin searching, and the current position is set to Origin.
	SENSOR+Z_PHASE	After Origin sensor detects the position set to 5.ORG→3:DIR/5:SENSOR, Z-phase is executed.
	SENSOR	After Origin sensor detects the position set to 5.ORG→3:DIR/5:SENSOR, it stops.

2.1.8 **SENSOR --- Sensor setup when in Origin searching**

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

Open ORG screen

Step 2. **Move to SENSOR screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

5
M

Select 5: SENSOR

Step 3. **SENSOR setup screen**

```
<XYZW : SENSOR>
        ORG SENSOR
X: CW           Y : CW
Z: CW           W: CW
```

SENSOR setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup".)

■ Comment

▷ When in Origin searching, sensors of each sensor are setup
→ Refer to 2.1.6 [Supplement] Setup of Origin direction & Moving direction (page. 2-10)

CAUTION	▶ If the controller is for an absolute type, DIR Parameter cannot be accessed.

2.1.9 SEN TYPE --- Setup of Origin sensor type

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                  item #
```

Open ORG screen

Step 2. **Move to SEN TYPE screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                  item #
```

6
N

Select 6: SEN TYPE

Step 3. **SEN TYPE setup screen**

```
<XYZW : SENSOR>
    ORG SENSOR SENSOR
X: NO           Y : NO
Z: NO           W: NO
```

① SEN TYPE setup screen

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ When 5.SENSOR is set to ORG, SENSOR’s contact type can be selected.

Item	Set-up value	Description
SENSOR	NO	Origin sensor – normally open type
	NC	Origin sensor – normally closed type

2.1.10 HOME --- Origin coordinate value setup

1. Setup Sequence

Step 1. **Open ORIGIN screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

Open ORG screen

Step 2. **Move to HOME screen**

```
<XYZW : ORG>
1: VEL          2: SEQ
3: DIR          4: RULE
5: SENSOR       6: SEN TYPE
7: HOME

                item #
```

7
G

Select 7: HOME

Step 3. **HOME setup screen**

```
<XYZW : SENSOR>
      ORIGIN HOME 0
X: 0            Y : 0
Z: 0            W: 0


H0    H1    H2    H3
```

F1

F4

Select HOME

F1: H0 F2: H1

F3: H2 F4: H3

▶ How to change set-up values (using number keys) and to save the values (See Page 1-12, 1.1.2 Parameter Setup”.)

■ Comment

▷ HOME coordinate value setup : H0 to H3

- The coordinate value which does not deviate from the work area (Refer to Parameter "RANG") is entered.
- Refer to HMOV in "Comment to commands"

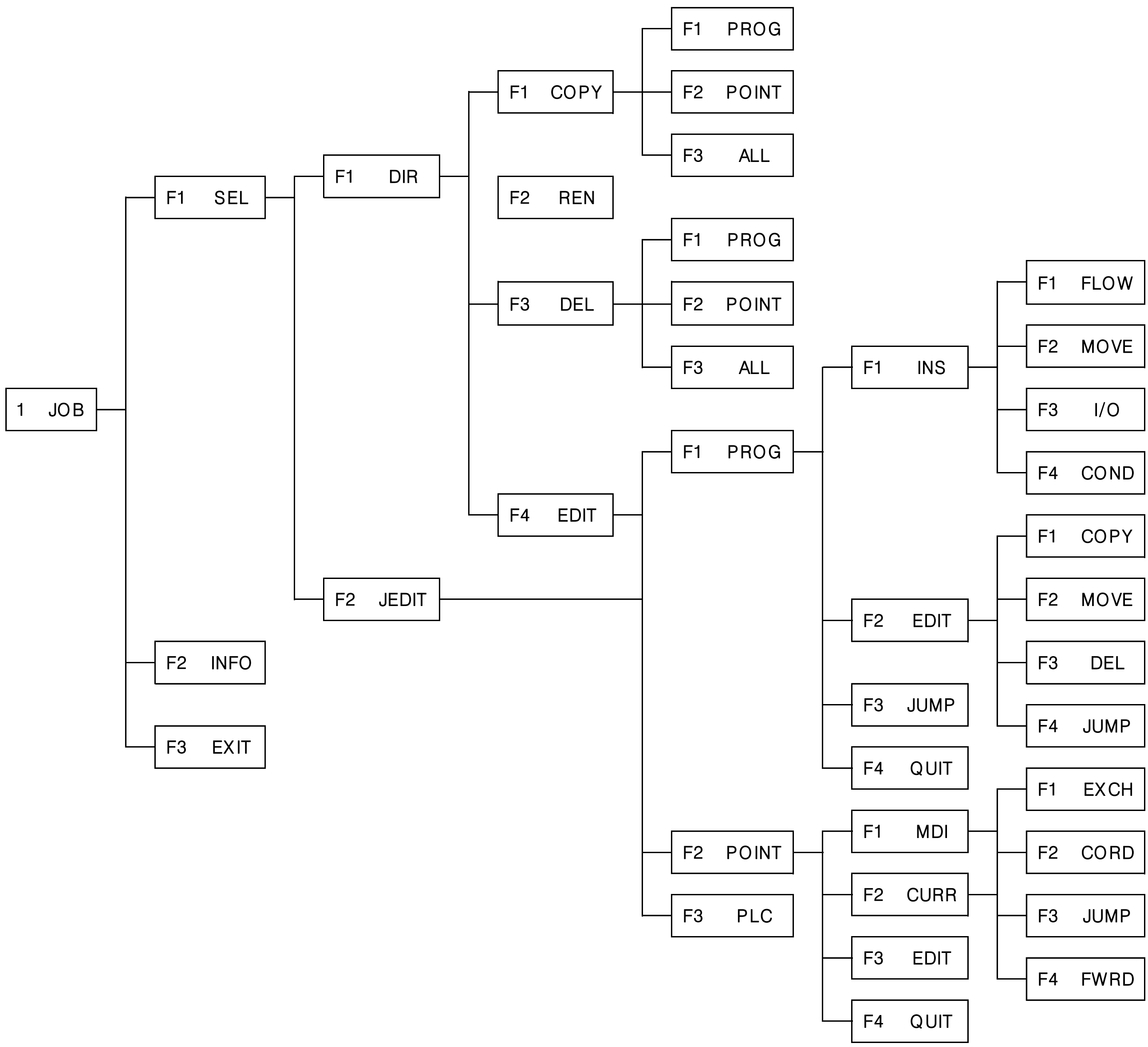
Chapter 3 JOB Mode

☞ For writing JOB, first ① User I/O is designated and ② Job point is specified. Based on these, Job can be written.

3.1 Overview

- A mode for writing (JEDIT Mode) or editing (DIR Mode) the Job Program robot. 1 JOB is comprised of 1 program file(PRO) and point file(PNT).
- In this controller, up to maximum 200 (0 to 199) jobs can be registered.
- 1 program file may have maximum 2,000 (0 to 1999) steps, and 1 point file may have the maximum 2,000 (0 to 1999) points.
- The name of each JOB must be 5 letters or less, including alphabet, number, or their combination.

3.2 JOB Mode Flowchart



3.3 **JEDIT (Job Edit)** Mode

3.3.1 **Overview**

- Selecting the number of unwritten (NEW) program and entering the commands is referred to as New Writing.
- Program can be written by using a teaching pendant or the dedicated software (Unihost).

▷ For information on how to use Unihost, refer to “Unihost Operation Manual”.

▷ For information on how to use a teaching pendant, refer to the following explanations.

3.3.2 **How to manipulate**

☞ New Writing of the program

- How to new writing the program will be explained with reference to the simple exercise.

Exercise

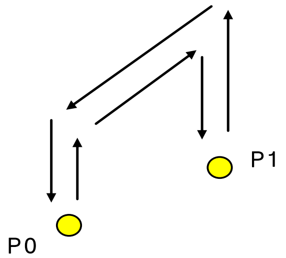
Job of reciprocal motion from P0 to P1

- JOB No. : 1

- Name JOB file : TEST 1

- Moving velocity : max. 50%

- Use of I/O : None



Program

```
0000    MAIN
0001    VEL 500
0002    LABL A0
0003    PLUP 5
0004    JMOV P0
0005    JMOV P1
0006    GOTO A0
0007    EOP
```

3.3.3 Enter JOB file name and number

JOB No. : 1
JOB file name : TEST 1

☞ **It is preferable that JOB file name is entered as "product name".**

1. Entry Sequence

Step 1. **Move to JOB screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

1 Q

Open intial menu screen and select "1: JOB"

Step 2. **Select ROBOT**

```
<JOB MAIN>
 NO             TYPE
*CH1            RGA80A
 CH2            RSA60A


SEL   INFO      EXIT
```

▲ ▼ F1

(Move cursor and then) Select robot

Step 3. **Select JOB EDIT item**

```
<RGA80A : JOB EDIT>





DIR     JEDIT
```

F2

Select JEDIT

Step 4. **Enter file name & number**

```
<RGA80A : JOB EDIT>
 FILE NAME = TEST1
```

Enter "TEST1"

SHIFT | ← T | SVON E | 3 S | ← T

SHIFT | 1 Q | ENTER

```
<RGA80A : EDIT>
 FILE NUM= 1
```

1 Q ENTER

Enter JOB number "1"

(If pressing Enter without entering the number, a window on which the unassigned number is entered appears.)

Step 5. **Write Program**

```
<RGA80A :EDIT>
 JOB NAME = TEST1



PROG    POINT    PLC
```

F1

Select "PROG"

```
<RGA80A : EDIT>
 STEP    INT        OP1        OP2
*0000    -
 0001    -
 0002    -
 0003    -
 0004    -
INS      EDIT       JUMP       QUIT
```

■ Write Program

▶ 3.3.5 How to insert How to insert Program (3-7 page)

▶ 3.3.6 How to save How to update Program (3-17 page)

Step 6. **Point teaching**

```
<RGA80A :EDIT>
 JOB NAME = TEST1



PROG    POINT    PLC
```

F2

Select "POINT"

```
<RGA80A : EDIT> V:         50
F:TEST    P0    NW         B L
X: 0            Y : 0
Z: 0            W: 0


MDI      CURR   EDIT       QUIT
```

■ Point teaching

▶ 3.3.7 Point Teaching method (3-18 page)

▶ 3.4.8Point Updating (3-93 page)

CAUTION

▶ When entering JOB file name and number, do not use duplicate name and number between channels.

3.3.4 **Typical robot command (Reference for inserting Program)**

Group	Selection key	FLOW	MOVE	I/O	COND
Selection key		F1	F2	F3	F4
Item move	F1	MAIN	JMOV	OUT	VEL
	F2	FOR	LMOV	POUT	FOS
	F3	IF	CMOV	IN	DLAY
	F4	WHILE	AMOV	PIN	INT
	F1	EOP	IMOV	CIN	ACC
	F2	NEXT	PMOV	CWIN	PFOS
	F3	ELSE	HMOV	COUT	PLUP
	F4	ENDWL		CWOUT	REAL
	F1	SUBR	SVON	SYS	FORM
	F2	RET	SVOF		TOOL
	F3	CALL	WITH		PCLR
	F4	ENDIF	ENDWT		POS
	F1	LABL	MVR		LIMT
	F2	GOTO	HERE		OFFS
	F3	JCALL	PASS		CNT
	F4		PCNT		INPOS
	F1	STOP			TMR
	F2	EXIT			NO
	F3				LEFT
	F4				RIGHT
	F1				FIX
	F2				TRQ
	F3				TQL
	F4				XCHG
	F1				DEC

3.3.5 How to insert Program

1. Insertion Sequence

Step 1. **Move to PROGRAM screen**

```
<RGA80A : EDIT>
STEP    INT    OP1   OP2
*0000    -
 0001    -
 0002    -
 0003    -
 0004    -
INS     EDIT     JUMP     QUIT
```

Open Program screen

Step 2. **Select INSERT**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
*0000    -
 0001    -
 0002    -
 0003    -
 0004    -
INS     EDIT     JUMP     QUIT
```

F1

Select INS (insert)

Step 3. **Enter MAIN**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
*0000    -
 0001    -
 0002    -
 0003    -
 0004    -
FLOW    MOVE      I/O   COND
```

F1

Select FLOW

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
*0000   MAIN
 0001    -
 0002    -
 0003    -
 0004    -
MAIN     FOR       IF   WHILE
```

F1

Select MAIN

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
*0001
 0002    -
 0003    -
 0004    -
MAIN     FOR       IF   WHILE
```

ENTER

Enter MAIN

Step 4. **Move to Group Command.**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
*0001   -
 0002   -
 0003   -
 0004   -
FLOW    MOVE     I/O   COND
```

ESC

Escape to group command

Step 5. **Enter VEL 500**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
*0001   -
 0002   -
 0003   -
 0004   -
FLOW    MOVE     I/O   COND
```

F4

Select COND

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
*0001   VEL
 0002   -
 0003   -
 0004   -
VEL     FOW    DLAY    INT
```

F1

Select VEL

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
*0001   VEL 500
 0002   -
 0003   -
 0004   -
VEL     FOW    DLAY    INT
```

5 M | 0 V | 0 V

Enter “500”

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002
 0003   -
 0004   -
VEL     FOW    DLAY    INT
```

ENTER

Enter “VEL 500”

Step 6. **Move to Group Command.**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2
 0000   MAIN
 0001   VEL 500
*0002   -
 0003   -
 0004   -
FLOW    MOVE      I/O   COND
```

ESC

Escape to group command

Step 7. **Enter LABL A0**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002   -
 0003   -
 0004   -
FLOW    MOVE      I/O   COND
```

F1

Select FLOW

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002
 0003   -
 0004   -
MAIN     FOR        IF   WHILE
```

▼

Move screen.

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002
 0003   -
 0004   -
EOP     NEXT   ELSE   ENDWL
```

▼

Move screen.

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002
 0003   -
 0004   -
SUBR    RET   CALL   ENDIF
```

▼

Move screen.

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002   LABL
 0003   -
 0004   -
LABL    GOTO    JCALL
```

F1

Select LABL

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002   LABL A0
 0003   -
 0004   -
LABL    GOTO    JCALL
```

SHIFT | I/O A

SHIFT | 0 V

Enter "A0"

(SHIFT LED ON)

(SHIFT LED OFF)

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003
 0004   -
LABL    GOTO    JCALL
```

ENTER

Enter "LABL A0"

Step 8. Move to Group Command.

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003   -
 0004   -
FLOW    MOVE    I/O   COND
```

ESC

Escape to group command

Step 9. Enter PLUP 5

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003   -
 0004   -
FLOW    MOVE    I/O   COND
```

F4

Select COND

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003   
 0004   -
VLE     FOS       DLAY      INT
```

▼

Move screen.

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003   PLUP
 0004   -
ACC     PFOS      PLUP      REAL
```

F3

Enter "PLUP"

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
*0003   PLUP 5
 0004   -
ACC     PFOS      PLUP      REAL
```

5 M

Enter "5"

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004   
ACC     PFOS      PLUP      REAL
```

ENTER

Enter "PLUP 5"

Step 10. Move to Group Command.

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004   -
FLOW    MOVE      I/O   COND
```

ESC

Escape to group command

Step 11. **Enter JMOV P0**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004
FLOW    MOVE       I/O   COND
```

Select MOVE

F2

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004   JMOV
JMOV   LMOV CMOV AMOV
```

Select JMOV

F1

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004   JMOV P
JMOV   LMOV CMOV AMOV
```

Enter "P"

SHIFT → P (SHIFT LED ON)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
*0004   JMOV P0
JMOV   LMOV CMOV AMOV
```

Enter "0"

SHIFT 0 V (SHIFT LED OFF)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
*0005
JMOV   LMOV CMOV AMOV
```

Enter "JMOV P0"

ENTER

Step 12. **Enter JMOV P1**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
*0005   JMOV
JMOV   LMOV CMOV AMOV
```

Select JMOV

F1

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
*0005   JMOV P
JMOV   LMOV CMOV AMOV
```

Enter "P"

SHIFT → P (SHIFT LED ON)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0001   VEL 500
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
*0005   JMOV P1
JMOV   LMOV CMOV AMOV
```

Enter "1"

SHIFT 1 Q (SHIFT LED OFF)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006
JMOV   LMOV CMOV AMOV
```

Enter "JMOV P1"

ENTER

Step 13. **Move to Group Command.**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006   -
FLOW    MOVE      I/O   COND
```

Escape to group command

ESC

Step 14. **Enter GOTO A0**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006   -
FLOW    MOVE      I/O   COND
```

F1

Select FLOW

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006
MAIN    FOR       IF   WHILE
```

▼

Screen move

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006
EOP     NEXT   ELSE ENDWL
```

▼

Screen move

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006
SUBR    RET   CALL   ENDIF
```

▼

Screen move

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006   GOTO
LABL    GOTO   JCALL
```

F2

Select GOTO

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006   GOTO A
LABL    GOTO   JCALL
```

Enter “A”

SHIFT | I/O A (SHIFT LED ON)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0002   LABL A0
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
*0006   GOTO A0
LABL    GOTO   JCALL
```

Enter “0”

SHIFT | 0 V (SHIFT LED OFF)

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
*0007
LABL    GOTO   JCALL
```

Enter “GOTO A0”

ENTER

Step 15. **Enter EOP**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
*0007
LABL    GOTO   JCALL
```

Screen move

▲

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
*0007
SUBR    RET    CALL   ENDIF
```

Screen move

▲

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0003   PLUP 5
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
*0007   EOP
EOP     NEXT   ELSE ENDWL
```

F1

Select EOP

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008
EOP     NEXT   ELSE ENDWL
```

ENTER

Enter “EOP”

Step 16. **Move to Group Command.**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008
FLOW    MOVE      I/O   COND
```

ESC

Escape to group command

Step 17. **Update Program**

```
<RGA80A : EDIT>
 STEP   INT    OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008
INS     EDIT      JUMP      QUIT
```

ESC

Refer to 2.2.3 ▶ Updating Program

3.3.6 How to update Program

1. Updating Sequence

Step 1. **Program writing completed**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008   -
INS     EDIT  JUMP  QUIT
```

Program writing has been completed.

Step 2. **Updating**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008   -
INS     EDIT  JUMP  QUIT
```

F4

Select QUIT

Step 3. **Select updating or not**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008   -
Update OK? (ENT/ESC)
```

ENTER

Select “ENTER” for Update OK?

Step 4. **Update completed**

```
<RGA80A : EDIT>
 STEP   INT   OP1   OP2   i
 0004   JMOV P0
 0005   JMOV P1
 0006   GOTO A0
 0007   EOP
*0008   -
Update Complete!
```

ENTER

Select “ENTER”

3.3.7 Point Teaching method

1. Teaching Sequence

Step 1. **JOB EDIT item**

```
<RGA80A : EDIT>
 JOB NAME = TEST1



PROG   POINT   PLC
```

F2

On JOB EDIT screen, select "POINT"

Step 2. **Set up Point teaching Mode**

```
<RGA80A : EDIT> V:        50
F:TEST   P0          NW        B L
A: 0.69              B : 0.83
Z: 0.72              W: 0.84



MDI      CURR   EDIT   QUIT
```

F1

F2

Select POINT TEACHING Mode

F1: MDI Mode

F2: CURR Mode

Step 3 **① MDI Mode**

```
<RGA80A : EDIT> V:        50
F:TEST   P0          NW        B L
A: 0.69              B : 0.83
Z: 0.72              W: 0.84



EXCH     CORD PJUMP FWRD
```

F1

① How is MDI selected?

② CURR Mode

```
<RGA80A : EDIT> V:        50
F:TEST   P0          NW        B L
A: 0.69              B : 0.83
Z: 0.72              W: 0.84



EXCH     CORD PJUMP FWRD
```

F2

② How is CURR selected?

■ Comment

①, ② :

- Point Teaching
  - 1.MDI Mode ⇨ ★ Refer to 3.4.6 PPoint Edit (position change)), 1) MDI
  - 2.CURR Mode
    - Direct teaching
    - Play Back teaching

■ About point teaching the job → Two kinds of teaching!

1. MDI Mode (Manual Direct Input)

   → Without moving the robot, the coordinate value is entered by using number keys.

2. CURR Mode (Current Mode Teaching)

   → After moving the robot to the desired location by using hand or axis moving keys, the current position is entered. (When the robot moves, the change in the coordinate value can be checked.

   - Direct teaching : Hands directly moves the robot
   - Play Back Teaching : Axis moving keys in a teaching pendant moves the robot

■ Explanation for point screen (1)

```
<RGA80A : EDIT>   V :       50
F:TEST    P0      NW        B   L
A: 0.69           B : 0.83
Z: 0.72           W: 0.84


MDI     CURR     EDIT     QUIT
```

- ▶ F: TEST (F=File name)
- ▶ P:0 (P=Point number)
- ▶ NW: (=New: not saved)
- ▶ US: (=Use: saved)
- \- B indication Yes→Axis confined due to Brake OFF,
- \- B indication No→Axis released from confinement due to Brake ON
- ▶ A,B,Z,W: Current position value of each axis
- ▶ MDI: MDI Mode Teaching
- ▶ CURR: CURRMode Teaching
- ▶ EDIT: Used for moving, deleting, and copying the pointent
- ▶ QUIT: Used to finally save the point file

■ Explanation for point screen (2)

```
<RGA80A : EDIT> V :         50
F:TEST    P0      NW        B L
A: 0.69           B : 0.83
Z: 0.72           W: 0.84


EXCH     CORD PJUMP FWRD
```

- ▶ EXCH (=Exchange: The current position value is saved in a temporary memory.)
- ▶ CORD (=Coordnate: Conversion of Angular coordinate ↔ Cartesian coordinate)
- ▶ (Applied for SCARA robot only)
- ▶ PJUMP (=Point Jump: Jumping the point)
- ▶ FWRD (=Forward: Check the saved(US) point

(1) CURR Teaching Sequence

1. Teaching Sequence

Step 1. **Change P0 coordinate value**

```
<RGA80A : EDIT>  V :        50
F:TEST   P0      NW         B L
A: 0.69          B : 0.83
Z: 0.72          W: 0.84


EXCH    CORD PJUMP FWRD
```

1) Move the robot to the desired location by using hand or axis moving key.
2) At this time, the changing state of the coordinate value can be monitored. (Refer to Comment)

Step 2. **Temporarily update P0 point**

```
<RGA80A : EDIT>  V :        50
F:TEST   P0      NW         B L
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


EXCH    CORD PJUMP FWRD
```

F1

Select EXCH

Step 3. **Update P0 point**

```
<RGA80A : EDIT>  V :        50
F:TEST   P0      NW         B L
A: 0.69          B : 0.83
Z: 0.72          W: 0.84


Used OK? (ENT/ESC)
```

ENTER

Select ENTER

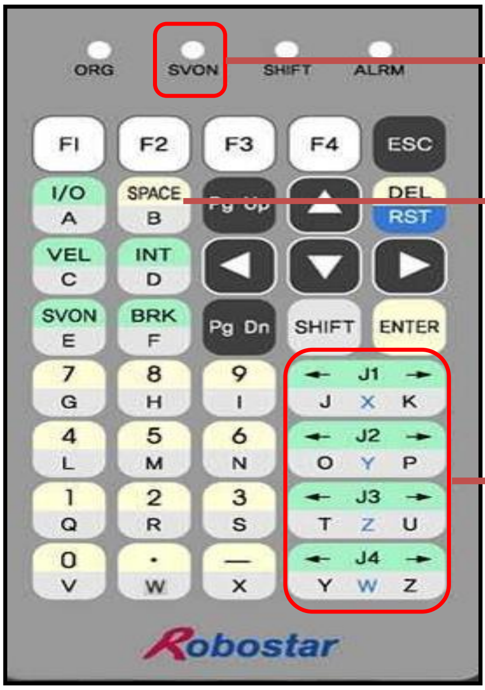
Step 4 **P0 Update completed**

```
<RGA80A : EDIT>  V :        50
F:TEST   P0      US         B L
A: 0.69          B : 0.83
Z: 0.72          W: 0.84


EXCH    CORD PJUMP FWRD
```

Change NW → US

■ Comment

☐ Keys used for CURR MODE (1)



▶ Servo ON / OFF changing key
- SVON lamp lights up due to this key depression: Servo ON,
- SVON lamp turns off due to this key depression: Servo OFF

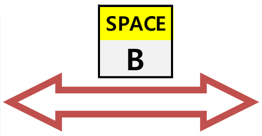
▶ JOG motion
- Robot moves while theses keys are being pressed.

▶ INCH motion
- When the keys are pressed, the root moves the unit moving distance set up in Parameter.

JOG

<RGA80A : EDIT>
F:TEST P0 NW B L
A: 0.69 B : 0.83
Z: 0.72 W: 0.84

MDI CURR EDIT QUIT



INCH

<RGA80A : EDIT>
F:TEST P0 NW B I
A: 0.69 B : 0.83
Z: 0.72 W: 0.84

MDI CURR EDIT QUIT

Step 5. **Move to P1 point screen**

```
<RGA80A : EDIT>  V :      50
F:TEST   P1      NW       B L
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


EXCH   CORD PJUMP FWRD
```

Pg Dn

Move to point P1 screen

Step 6. **Change P1 coordinate value**

```
<RGA80A : EDIT>  V :      50
F:TEST   P1      NW       B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


EXCH   CORD PJUMP FWRD
```

Again move the robot to the desired location by using hand or axis moving key.

Step 7. **Temporarily update P1 point**

```
<RGA80A : EDIT>  V :      50
F:TEST   P1      NW       B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


EXCH   CORD PJUMP FWRD
```

F1

The point is temporarily saved

Step 8 **Update P1**

```
<RGA80A : EDIT>  V :      50
F:TEST   P1      US       B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


Used OK? (ENT/ESC)
```

ENTER

P1 is saved.

■ Comment

☐ Keys used for CURR MODE (2)

BRK OFF

```
F:TEST    P0          NW          B L
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

BRK
F

BRK ON

```
F:TEST    P0          NW            L
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

▶ Change velocity in JOG Mode

JOG

```
F:TEST    P0          NW          B L
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

VEL
C

```
F:TEST    P0          NW          B M
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

VEL
C

```
F:TEST    P0          NW          B H
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

▶ Change unit distance in INCH Mode

INCH

```
F:TEST    P0          NW          B l
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

VEL
C

```
F:TEST    P0          NW          B m
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

VEL
C

```
F:TEST    P0          NW          B h
A: 89.69.             B : 2.83
Z: 59.72              W: 22.84

EXCH    CORD PJUMP FWRD
```

(2) Finally updating the point

2. Updating Sequence

Step 9 **Move to previous screen**

```
<RGA80A : EDIT>  V :       50
F:TEST   P1      US        B L
A: 22.02         B : 70.38
Z: 59.72         W: 102


EXCH    CORD PJUMP FWRD
```

ESC

Move to previous screen

Step 10. **Final updating**

```
<RGA80A : EDIT>  V :       50
F:TEST   P1      US        B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


MDI     CURR   EDIT   QUIT
```

F4

Select QUIT

```
<RGA80A : EDIT>  V :       50
F:TEST   P1      US        B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


Update OK? (ENT/ESC)
```

ENTER

Select ENTER

```
<RGA80A : EDIT>  V :       50
F:TEST   P1      US        B L
A: 29.02         B : 70.38
Z: 59.72         W: 102


Update Complte!
```

ENTER

Select ENTER once again

```
<RGA80A : EDIT>
JOB NAME  =  TEST1



PROG   POINT   PLC
```

Finally save the point

■ Comment

■ Explanation for point screen (3)

▶ EXCH (=Exchange: The current position value is saved in a temporary memory.)

```
F:TEST    P0        NW        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

EXCH    CORD PJUMP FWRD
```

F1

```
F:TEST    P0        US        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

EXCH    CORD PJUMP FWRD
```

▶ CORD (=Coordnate: Convert Angular coordinate ↔ Cartesian coordinate) : (Applied for SCARA robot only.)

```
F:TEST    P0        NW        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

EXCH    CORD PJUMP FWRD
```

F2

```
F:TEST    P0        NW        B L
X: 599.89           B : 0.23
Z: 59.72            W: 22.84

EXCH    CORD PJUMP FWRD
```

▶ PJUMP (=Point Jump: Jump the point)

```
F:TEST    P2        NW        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

EXCH    CORD PJUMP FWRD
```

F3

```
F:TEST    P2        NW        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

Point # = A
```

1 Q | 0 V | 0 V | ENTER

```
F:TEST    P100      NW        B L
A: 0.69             B : 0.83
Z: 0.72             W: 0.84

EXCH    CORD PJUMP FWRD
```

Pg Dn Increase by 1 point number:: P1 → P2 → .....

Pg Up Decrease by 1 point number: ..... → P1 → P0

(3) How to check the updated point (Suppose the case that Point P2 among points taught above is checked)

3. Checking Sequence

Step 1 **Move to P1 screen**

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   L
A: 89.69         B : 2.83
Z: 59.72         W: 22.84

EXCH   CORD PJUMP FWRD
```

Move to point P1 screen

Step 2. **FORWARD**

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84

EXCH   CORD PJUMP FWRD
```

VEL C

First select Forward moving velocity.

▶ Refer to 3.3.7 Point Teaching method (3-18 page), Comment: Keys used for CURR MODE (2).

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84

EXCH   CORD PJUMP FWRD
```

F4

Select FWRD

Step 3-1. **Select JMOV**

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84

JMOVE   LMOVE
```

F1 **ENTER**

Select JMOV

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84

Pull Up = 10
```

1 Q **0 V** **ENTER**

If there is Z-axis, enter PULL UP value.

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


Ready? (ENT/ESC)
```

ENTER

Execute JMOV

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


Forward Complete!
```

JMOV completed

Step 3-2. **Select LMOV**

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


JMOVE   LMOVE
```

F2

Select LMOV

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


Ready? (ENT/ESC)
```

ENTER

Execute LMOV

```
<RGA80A : EDIT>  V :        50
F:TEST    P1     US    B   M
A: 89.69         B : 2.83
Z: 59.72         W: 22.84


Forward Complete
```

LMOV completed

Important

▶ In JEDIT Mode, when Job Program writing and Job Point teaching have been completed, the robot can execute the job program by using a teaching pendant, external PLC, etc.
For execution of Job Program, refer to "Chapter 4 About RUN Mode".

3.4 DIR (= Directory) Mode

3.4.1 Overview

- **Conventionally registered JOB files are edited, copied, and deleted.**
- **JobPoint and JobSequence Program are edited or added.**
- **Among the registered JOB file, Point and Program are copied, moved, and deleted.**

3.4.2 JOB file edition

- DIR Move Sequence

Step 1. **Move to MAIN**

```
        SYSTEM   MODE

CH   MODE   J_NUM    STATE
 1   AUTO NONE       IDL
 2   AUTO NONE       IDL


 1R    2R            EXIT
```

F4

Power on and then on intial screen select "F4".

```
          Robostar
     NewRo RCQ-Series
     Version :N1-1.11.ST
   FPGA :05(110613_ABS)

       Press ENTER Key
```

ENTER

Select "ENTER"

Step 2. **Select JOB**

```
 <MAIN MENU>
1. JOB              2. RUN
3. HOST             4. PARA
5. ORIGIN           6. I/O
7. SYSTEM           8. GLB PNT
9. INT/FLT          A. ALARM

                     SELECT #
```

1 Q

Select 1: JOB

```
 <JOB MAIN>
  NO               TYPE
*CH1               RGA80A
  CH2              RSA60A


SEL   INFO                 EXIT
```

F1

Select ROBOT

Step 3. **Move to DIR Mode**

```
<RGA80A : JOB EDIT>



DIR   JEDIT
```

F1

Select DIR

```
< RGA80A: DIR>
 * 0. TEST1          5   STEP
   .__               .__
   .__               .__
   .__               .__
   .__               .__
   .__               .__
 COPY     REN       DEL       EDIT
```

Open DIRECTORY Mode screen

3.4.3 JOB file copy, file name change or deletion

■ **COPY ---- The updated JOB file is copied.**

- Copy with a new file name (3-30 page)
- Overwrite the existing Job file (3-32 page)

3.4.3.1 Copy with a new file name

Step 1. **Select COPY Program**

```
<RGA80A : DIR>
 * 0. TEST1          5   STEP
   .__               .__
   .__               .__
   .__               .__
   .__               .__
   .__               .__
 COPY    REN       DEL      EDIT
```

Select the desired program by marking *.

▲ ▼

Step 2. **COPY**

```
<RGA80A : DIR>
 * 0. TEST1          5   STEP
   .__               .__
   .__               .__
   .__               .__
   .__               .__
   .__               .__
 COPY    REN       DEL      EDIT
```

Select COPY

F1

```
<RGA80A : COPY>
 JOB NAME = TEST1



 PROG    POINT     ALL
```

Select ALL (PROG+POINT)

F3

Step 3. **Enter file name**

```
<RGA80A : COPY>
 JOB NAME = TEST1



 NEW FILE NAME = GOLD A
```

Enter file name "GOLD"

SHIFT	7 G	← O
4 L	INT D	ENTER

Step 4. **Enter file number**

```
<RGA80A : COPY>
 JOB NAME = TEST1



 NEW FILE NUM = 1
```

Enter file number "1"

1 Q ENTER

Step 5. **Copy completed**

```
<RGA80A : COPY>
 JOB NAME = TEST1



Copy Complete !
```

Copy completed

ENTER

```
<RGA80A : DIR>
 * 0. TEST1        5   STEP
   1. GOLD         5   STEP
    ._             ._
    ._             ._
    ._             ._
    ._             ._
 COPY    REN      DEL      EDIT
```

Completed screen

※ If the already updated file number is entered, "Already Num Used!" warning message appears.
To solve this : After pressing "Enter" Key, enter another file number.

■ Comment

☐ Explanation for screen (1)

File number | File | Number of steps of the saved Job Program

```
* 0. TEST1      5   STEP
  1. GOLD       5   STEP
   ._           ._
   ._           ._
COPY   REN     DEL    EDIT
```

▶ COPY: The saved Job file is copied

▶ REN: The saved Job Program name is changed to another.

▶ DEL: The saved Job file is deleted.

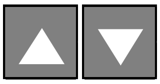
▶ EDIT: The saved Job Program and Job point are edited and added.

3.4.3.2 Overwrite the existing Job file

Step 1. **Select COPY Program**

```
<RGA80A : DIR>
* 0. TEST1         5   STEP
  1. GOLD          5   STEP
   .—              .—
   .—              .—
   .—              .—
   .—              .—
 COPY    REN     DEL     EDIT
```

Select the desired program by marking *.



Step 2. **COPY**

```
<RGA80A : DIR>
* 0. TEST1         5   STEP
  1. GOLD          5   STEP
   .—              .—
   .—              .—
   .—              .—
   .—              .—
 COPY    REN     DEL     EDIT
```

Select COPY

F1

```
<RGA80A : COPY>
 JOB NAME = TEST1



 PROG    POINT   ALL
```

F1 or F2

F1: PROGRAM copy

F2: POINT copy

F3: PROGRAM + POINT copy

Step 3. **Enter file name**

```
<RGA80A : COPY>
 JOB NAME = TEST1



 NEW FILE NAME = GOLD A
```

Enter file name “GOLD”

SHIFT	7 G	← O
4 L	INT D	ENTER

Step 4. **Enter file number**

```
<RGA80A : COPY>
 JOB NAME = TEST1



 NEW FILE NUM = 1 A
```

Enter file number "1"

1 Q	ENTER

Step 5.

Copy completed

```
<RGA80A : COPY>
JOB NAME = TEST1


  DIR 선택
Copy Complete !
```

ENTER

Copy is completed

```
<RGA80A : DIR>
* 0. TEST1        5   STEP
* 1. GOLD         5   STEP
   .__            .__
   .__            .__
   .__            .__
   .__            .__
COPY    REN      DEL      EDIT
```

DIR screen after copy is completed

■ Comment

☐ Explanation for screen (2)

File number

File name

Number of steps of the saved Job Program

```
* 0. TEST1        5   STEP
* 1. GOLD         5   STEP
   .__            .__
   .__            .__
PROG    POINT   ALL
```

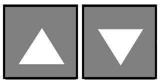
- ▶ PROG : Copy Job Program
- ▶ POINT: Copy Job point
- ▶ ALL: Copy Job Program + point

■ **REN (= Rename)** ---- The name of the updated JOB file is changed to another

Step 1. **Select RENAME**

```
<RGA80A : DIR>
* 0. TEST1          5   STEP
  1. GOLD           5   STEP
   .__              .__
   .__              .__
   .__              .__
   .__              .__
 COPY    REN     DEL     EDIT
```

Select the desired program by marking *.



```
<RGA80A : DIR>
* 0. TEST1          5   STEP
  1. GOLD           5   STEP
   .__              .__
   .__              .__
   .__              .__
   .__              .__
 COPY    REN     DEL     EDIT
```

Select REN

F2

Step 2. **Enter file name**

```
<RGA80A : REN>
JOB NAME = TEST1



NEW FILE NAME = STAR A
```

Enter file name "STAR"

SHIFT	3 S	← T
I/O A	2 R	ENTER

Step 3. **Changing completed**

```
<RGA80A : REN>
* 0. STAR           5   STEP
  1. GOLD           5   STEP
   .__              .__
   .__              .__
   .__              .__
   .__              .__
 Rename Complete!
```

Changing completed

ENTER

```
<RGA80A : DIR>
* 0. TEST1          5   STEP
  1. GOLD           5   STEP
   .__              .__
   .__              .__
   .__              .__
   .__              .__
 COPY    REN     DEL     EDIT
```

Completed screen

■ **DEL (= Delete)** ---- The updated JOB file is deleted.

Step 1. **Select DELETE Program**

```
<RGA80A : DIR>
 * 0. STAR          5   STEP
   1. GOLD          5   STEP
    .__              .__
    .__              .__
    .__              .__
    .__              .__
 COPY     REN      DEL      EDIT
```

Select the desired program by marking *.

Step 2. **Select DELETE**

```
<RGA80A : DIR>
 * 0. STAR          5   STEP
   1. GOLD          5   STEP
    .__              .__
    .__              .__
    .__              .__
    .__              .__
 COPY     REN      DEL      EDIT
```

Select DEL

F3

Step 3. **Select DELETE item**

```
<RGA80A : DEL>
 JOB NAME = STAR



 PROG     POINT    ALL
```

Select ALL

F3

Step 4. **Delete**

```
<RGA80A : DEL>
 JOB NAME = STAR



 Delete OK? (ENT/ESC)
```

Select ENTER

ENTER

Step 5. **Deletion completed**

```
<RGA80A : DEL>
 JOB NAME = STAR




 Delete Complete!
```

ENTER

Delete

```
<RGA80A : DIR>
 * 0. GOLD          5   STEP
    .—              .—
    .—              .—
    .—              .—
    .—              .—
    .—              .—
 COPY     REN      DEL      EDIT
```

Deletion completed

※ A file number deleted after the updated job is deleted can be used.

3.4.4 **Program edition**

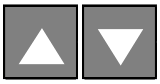
■ **EDIT** ---- Program edition

```
FILE NAME   : TEST1
     E.g.)        000      How to insert VEL500?
     MAIN
```

Step 1. **Select Program**

```
<RGA80A : DIR>
 * 0. TEST1          7   STEP
   .__                .__
   .__                .__
   .__                .__
   .__                .__
   .__                .__
 COPY    REN     DEL     EDIT
```

Select the desired program by marking *.



Step 2. **Select EDIT**

```
<RGA80A : DIR>
 * 0. TEST1          7   STEP
   .__                .__
   .__                .__
   .__                .__
   .__                .__
   .__                .__
 COPY    REN     DEL     EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
 JOB NAME  =  TEST1




 PROG    POINT    PLC
```

Select PROG

F1

Step 4. **Edit Program**

```
<RGA80A : EDIT>
  STEP    INS     OP1    OP2
*0000     MAIN
  0001    JMOV P1
  0002    JMOV P2
  0003    JMOV P3
  0004    JMOV P4
INS       EDIT    JUMP   QUIT
```

Select INS(=INSERT)

F1

Step 5. **Move to position to be inserted**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2   i
 0000   MAIN
*0001   JMOV P1
 0002   JMOV P2
 0003   JMOV P3
 0004   JMOV P4
FLOW    MOVE   I/O   COND
```

Move cursor

▼

Step 6. **Enter VEL 50**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2   i
 0000   MAIN
*0001   A JMOV P1
 0002   JMOV P2
 0003   JMOV P3
 0004   JMOV P4
FLOW    MOVE   I/O   COND
```

Select COND

F4

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2   i
 0000   MAIN
*0001   VEL A
 0002   JMOV P2
 0003   JMOV P3
 0004   JMOV P4
VEL     FOS   DLAY   INT
```

Select VEL

F1

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002   A JMOV P1
 0003   JMOV P2
 0004   JMOV P3
VEL     FOS   DLAY   INT
```

Enter "500"

5 M	0 V	0 V	ENTER

Step 7. **Move to group command**

```
<RGA80A : EDIT>
STEP    INS    OP1   OP2   i
 0000   MAIN
 0001   VEL 500
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Escape to group command.

ESC

```
<RGA80A : EDIT>
STEP    INS    OP1    OP2    i
 0000   MAIN
 0001   VEL 500
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT   JUMP   QUIT
```

ESC

Select ESC

Step 8. **Update**

```
<RGA80A : EDIT>
STEP    INS    OP1    OP2
 0000   MAIN
 0001   VEL 500
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT   JUMP   QUIT
```

F4

Select QUIT

```
<RGA80A : EDIT>
STEP    INS    OP1    OP2
 0000   MAIN
 0001   VEL 500
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
Update OK? (ENT/ESC)
```

ENTER

Select ENTER

Step 9. **Update completed**

```
<RGA80A : EDIT>
STEP    INS    OP1    OP2
 0000   MAIN
 0001   VEL 500
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
Update Complete !
```

ENTER

Update

```
<RGA80A : EDIT>
 JOB NAME = TEST1



 PROG    POINT    PLC
```

Update completed

■ **EDIT** ---- Program editing

FILE NAME : TEST1

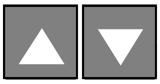
E.g.) 000 MAIN
001 VEL 500
002 JMOV P1

⇨ **How to change to VEL 200?**

Step 1. **Select Program**

```
<RGA80A : DIR>
* 0. TEST1          8  STEP
  .—                 .—
  .—                 .—
  .—                 .—
  .—                 .—
  .—                 .—
COPY    REN     DEL     EDIT
```

Select the desired program by marking *.



Step 2. **Select EDIT**

```
<RGA80A : DIR>
* 0. TEST1          8  STEP
  .—                 .—
  .—                 .—
  .—                 .—
  .—                 .—
  .—                 .—
COPY    REN     DEL     EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
JOB NAME = TEST1



PROG    POINT   PLC
```

Select PROG

F1

Step 4. **Edit Program**

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
*0000    MAIN
 0001    VEL 500
 0002    JMOV P1
 0003    JMOV P2
 0004    JMOV P3
INS      EDIT   JUMP   QUIT
```

Select INS(=INSERT)

F1

Step 5. **Move to position to be changed**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2   i
 0000   MAIN
*0001   VEL 50
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Move cursor

Step 6. **EDIT**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2 e
 0000   MAIN
*0001   VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Change with EDIT (i → e)

. W

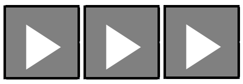
```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2 e
 0000   MAIN
*0001   A VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Select ENTER

ENTER

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2 e
 0000   MAIN
*0001   VEL A00
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Move cursor



Step 7. **Change VEL 500 → VEL 200**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2 e
 0000   MAIN
*0001   VEL A
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Delete "500"

DEL RST DEL RST DEL RST

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2 e
 0000   MAIN
 0001   VEL 200
*0002   A JMOV P1
 0003   JMOV P2
 0004   JMOV P3
FLOW    MOVE   I/O   COND
```

Enter "200"

2 R | 0 V | 0 V | ENTER

Step 8. **Update after change**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 200
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT    JUMP   QUIT
```

Escape to update after change

ESC

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 200
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT    JUMP   QUIT
```

Select QUIT

F4

Step 9. **Update completed**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 200
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
Update OK? (ENT/ESC)
```

Select ENTER

ENTER

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 200
*0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
Update Complete !
```

Update

ENTER

```
<RGA80A : EDIT>
 JOB NAME = TEST1




 PROG    POINT    PLC
```

Updating change completed

■ **EDIT** ---- Program STEP deletion

FILE NAME : TEST1

E.g.) 000 MAIN

001 VEL 500

002 JMOV P1 ⇨ **How to delete?**

Step 1. **Select Program**

```
<RGA80A : DIR>
 * 0. TEST1          8   STEP
   ._                ._
   ._                ._
   ._                ._
   ._                ._
   ._                ._
 COPY     REN     DEL      EDIT
```

Select the desired program by marking *.

Step 2. **Select EDIT**

```
<RGA80A : DIR>
 * 0. TEST1          8   STEP
   ._                ._
   ._                ._
   ._                ._
   ._                ._
   ._                ._
 COPY     REN     DEL      EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
 JOB NAME = TEST1



 PROG     POINT    PLC
```

Select PROG

F1

Step 4. **Move to position to be deleted**

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
*0001    VEL 500
 0002    JMOV P1
 0003    JMOV P2
 0004    JMOV P3
INS      EDIT    JUMP   QUIT
```

Move cursor

Step 5. **Delete STEP**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT    JUMP   QUIT
```

F2

Select EDIT

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
COPY    MOVE    DEL   JUMP
```

F3

Select DEL

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
Delete OK? (ENT/ESC)
```

ENTER

ENTER

Step 6. **Update after change**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   JMOV P1
 0002   JMOV P2
 0003   JMOV P3
 0004   JMOV P4
COPY    MOVE    DEL   JUMP
```

ESC

Escape to update after change

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   JMOV P1
 0002   JMOV P2
 0003   JMOV P3
 0004   JMOV P4
INS     EDIT    JUMP   QUIT
```

F4

Select QUIT

Step 7. **Update completed**

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
*0001    JMOV P1
 0002    JMOV P2
 0003    JMOV P3
 0004    JMOV P4
Update OK? (ENT/ESC)
```

Select ENTER

ENTER

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
*0001    JMOV P1
 0002    JMOV P2
 0003    JMOV P3
 0004    JMOV P4
Update Complete !
```

Update

ENTER

```
<RGA80A : EDIT>
 JOB NAME = TEST1




 PROG    POINT    PLC
```

Update step deletion completed

■ **EDIT** ---- Program STEP block deletion

FILE NAME : TEST1

E.g.) 000 MAIN

001 VEL 500
002 JMOV P1 } **How to delete?**
003 JMOV P2

Step 1. **Select Program**

```
<RGA80A : DIR>
* 0. TEST1          7  STEP
  .__               .__
  .__               .__
  .__               .__
  .__               .__
  .__               .__
COPY    REN     DEL     EDIT
```

Select the desired program by marking *.

Step 2. **Select EDIT**

```
<RGA80A : DIR>
* 0. TEST1          8  STEP
  .__               .__
  .__               .__
  .__               .__
  .__               .__
  .__               .__
COPY    REN     DEL     EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
JOB NAME = TEST1




PROG    POINT   PLC
```

Select PROG

F1

Step 4. **block setting start position**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
*0001   VEL 500
 0002   JMOV P1
 0003   JMOV P2
 0004   JMOV P3
INS     EDIT   JUMP  QUIT
```

Move cursor

Step 5. **Set block**

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
*0001    VEL 500
 0002    JMOV P1
 0003    JMOV P2
 0004    JMOV P3
INS      EDIT    JUMP   QUIT
```

Select EDIT

F2

■ How to set block

```
STEP   INS   OP1   OP2   i
 0000    MAIN
*0001  <VEL 500

COPY    MOVE    DEL      JUMP
```

⇩

```
STEP   INS   OP1   OP2   i
 0001  <VEL 500
*0002  <JMOV P1

COPY    MOVE    DEL      JUMP
```

⇩

```
STEP   INS   OP1   OP2   i
*0002  <JMOV P1
 0003    JMOV P2

COPY    MOVE    DEL      JUMP
```

① Using the left key, move the cursor (*) to a first step of a block to be set.

② SHIFT SPACE B Pressing the left key, the first step of the block is set. ("<" displayed)

③ If moving up to the last step of the block setting, by using the left key, "<" appears continuously.

④ At the last step, if SPACE B key is pressed once again, block setting is completed.

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
*0001  < VEL 500
 0002    JMOV P1
 0003    JMOV P2
 0004    JMOV P3
COPY     MOVE     DEL   JUMP
```

Start block setting

SHIFT

SPACE B

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
 0000    MAIN
 0001  < VEL 500
*0002  < JMOV P1
 0003    JMOV P2
 0004    JMOV P3
COPY     MOVE     DEL   JUMP
```

Move step

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
 0000    MAIN
 0001  < VEL 500
*0002  < JMOV P1
 0003    JMOV P2
 0004    JMOV P3
COPY     MOVE     DEL   JUMP
```

SPACE
B

Finish block setting

Step 6. **Delete the set block**

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
 0000    MAIN
 0001  < VEL 500
*0002  < JMOV P1
 0003    JMOV P2
 0004    JMOV P3
COPY     MOVE     DEL   JUMP
```

F3

Select DEL

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
 0000    MAIN
 0001    JMOV P2
*0002    JMOV P3
 0003    JMOV P4
 0004    EOP
Delete OK? (ENT/ESC)
```

ENTER

Delete

■ If copy or move is executed without setting the block?

```
STEP     INS   OP1   OP2
*0000    MAIN
 0001    VEL 500
No Source Block
```

■ Block-wise COPY, MOVE, or DEL is executed after the block is set.

■ Unsetting the block 1)

```
STEP     INS    OP1   OP2
 0001  < VEL 500
*0002  < JMOV P1

COPY     MOVE   DEL      JUMP
```

SPACE B

```
STEP     INS    OP1   OP2
 0001    VEL 500
*0002    JMOV P1

COPY     MOVE   DEL      JUMP
```

■ Unsetting the block 2)

```
STEP     INS    OP1   OP2
 0001  < VEL 500
*0002  < JMOV P1

COPY     MOVE   DEL      JUMP
```

ESC

```
STEP     INS    OP1   OP2
 0001    VEL 500
*0002    JMOV P1

 INS     EDIT  JUMP  QUIT
```

Step 7. **Move to previous screen**

```
<RGA80A : EDIT>
 STEP    INS    OP1   OP2
 0000    MAIN
 0001    JMOV P2
*0002    JMOV P3
 0003    JMOV P4
 0004    EOP
COPY     MOVE      DEL   JUMP
```

ESC

Move to previous screen

Step 8. **Update**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   JMOV P2
*0002   JMOV P3
 0003   JMOV P4
 0004   EOP
INS     EDIT   JUMP   QUIT
```

Select QUIT

F4

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   JMOV P2
*0002   JMOV P3
 0003   JMOV P4
 0004   EOP
Update OK? (ENT/ESC)
```

Select ENTER

ENTER

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   JMOV P2
*0002   JMOV P3
 0003   JMOV P4
 0004   EOP
Update Complete !
```

Update

ENTER

```
<RGA80A : EDIT>
 JOB NAME = TEST1



 PROG    POINT    PLC
```

Update is completed after step block is deleted.

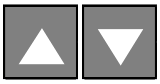
■ **EDIT** ---- Program STEP block copy

```
File name : TEST1

0000 MAIN
0001 VEL 100
0002 LABL A0
0003 JMOV P0
0004 JMOV P1
0005 DLAY 100
0006 GOTO A0
0007 EOP
```

```
File name : TEST1

0000 MAIN
0001 VEL 100
0002 LABL A0
0003 JMOV P0
0004 JMOV P1
0005 DLAY 100
0006 JMOV P0
0007 JMOV P1
0008 GOTO A0
0009 EOP
```

Step 1. **Select Program**

```
<RGA80A : DIR>
* 0. TEST1          8   STEP
  .__               .__
  .__               .__
  .__               .__
  .__               .__
  .__               .__
COPY    REN     DEL     EDIT
```



Select the desired program by marking *.

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
* 0. TEST1          8   STEP
  .__               .__
  .__               .__
  .__               .__
  .__               .__
  .__               .__
COPY    REN     DEL     EDIT
```

F4

Select EDIT

Step 3. **Select PROG**

```
<RGA80A : EDIT>
JOB NAME = TEST1



PROG    POINT   PLC
```

F1

Select PROG

Step 4. **EDIT**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
INS     EDIT    JUMP   QUIT
```

F2

Select EDIT

Step 5. **Set block**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
COPY    MOVE    DEL   JUMP
```

EDIT screen

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003   JMOV P0
 0004   JMOV P1
COPY    MOVE    DEL   JUMP
```

▼

Move a cursor (*) to a first step of block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003 < JMOV P0
 0004   JMOV P1
COPY    MOVE    DEL   JUMP
```

SHIFT
SPACE
B

Start block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE    DEL   JUMP
```

▼

Move

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE     DEL   JUMP
```

SPACE
B

Block setting completed

Step 6. **Copy**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003 < JMOV P0
 0004 < JMOV P1
 0005   DLAY 100
*0006   GOTO A0
COPY    MOVE     DEL   JUMP
```

Move a cursor (*) to the position to be copied

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   GOTO A0
COPY    MOVE     DEL   JUMP
```

F1

Select COPY

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
COPY    MOVE     DEL   JUMP
```

ENTER

Confirm COPY

Step 7. **Copy completed**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
COPY    MOVE     DEL   JUMP
```

COPY completed

▶ For Program updating after COPY is completed, refer to 3-60 page, "(3) Program updating".

■ **EDIT** ---- Program STEP block copy using JUMP

※ Step block copy using JUMP

```
FILE name : TEST1

0000 MAIN
0001 VEL 100
0002 LABL A0
0003 JMOV P0
0004 JMOV P1
0005 DLAY 100
0006 GOTO A0
0007 EOP
```

⇨

```
FILE name : TEST1

0000 MAIN
0001 VEL 100
0002 LABL A0
0003 JMOV P0
0004 JMOV P1
0005 DLAY 100
0006 GOTO A0
0007 EOP
0008 -
0009 JMOV P0
0010 JMOV P1
```

Step 1. **Select Program**

```
<RGA80A : DIR>
 * 0. TEST1          8   STEP
   ._                ._
   ._                ._
   ._                ._
   ._                ._
   ._                ._
 COPY    REN      DEL      EDIT
```

Select the desired program by marking *.

▲ ▼

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
 * 0. TEST1          8   STEP
   ._                ._
   ._                ._
   ._                ._
   ._                ._
   ._                ._
 COPY    REN      DEL      EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
 JOB NAME = TEST1



 PROG    POINT    PLC
```

Select PROG

F1

Step 4. **EDIT**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
INS     EDIT    JUMP   QUIT
```

F2

Select EDIT

Step 5. **Set block**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

EDIT screen

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003   JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

Move a cursor (*) to a first step of block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003 < JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

SHIFT
SPACE
B

Start block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE     DEL   JUMP
```

Move

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE    DEL   JUMP
```

Block setting is completed

SPACE B

Step 6. **STEP Jump**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE    DEL   JUMP
```

Select JUMP

F4

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
STEP # = 9 A
```

Enter Position to jump to

9 I ENTER

Step 7. **COPY**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0009   —
 0010   —
 0011   —
 0012   —
 0013   —
COPY    MOVE    DEL   JUMP
```

Select COPY

F1

Step 8. **Completed**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0009   JMOV P0
 0010   JMOV P1
 0011   —
 0012   —
 0013   —
COPY    MOVE    DEL   JUMP
```

COPY completed

■ **EDIT** ---- Program STEP block move

```
FILE name : TEST1

000 MAIN
001 VEL 100
002 LABL A0
003 JMOV P0
004 JMOV P1
005 DLAY 100
006 GOTO A0
007 EOP
```

⇨

```
FILE name : TEST1

000 MAIN
001 VEL 100
002 LABL A0
003 DLAY 100
004 JMOV P0
005 JMOV P1
006 GOTO A0
007 EOP
```

Step 1. **Select Program**

```
<RGA80A : DIR>
 * 0. TEST1           8   STEP
   .__                .__
   .__                .__
   .__                .__
   .__                .__
   .__                .__
 COPY    REN      DEL      EDIT
```

Select the desired program by marking *.

▲ ▼

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
 * 0. TEST1           8   STEP
   .__                .__
   .__                .__
   .__                .__
   .__                .__
   .__                .__
 COPY    REN      DEL      EDIT
```

Select EDIT

F4

Step 3. **Select PROG**

```
<RGA80A : EDIT>
 JOB NAME = TEST1



 PROG    POINT    PLC
```

Select PROG

F1

Step 4. **EDIT**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
INS     EDIT    JUMP   QUIT
```

F2

Select EDIT

Step 5. **Set block**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
*0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

EDIT screen

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003   JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

▼

Move a cursor (*) to a first step of block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
*0003 < JMOV P0
 0004   JMOV P1
COPY    MOVE     DEL   JUMP
```

SHIFT

SPACE
B

Start block setting

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0000   MAIN
 0001   VEL 100
 0002   LABL A0
 0003 < JMOV P0
*0004 < JMOV P1
COPY    MOVE     DEL   JUMP
```

▼

Move

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0000    MAIN
 0001    VEL 100
 0002    LABL A0
 0003  < JMOV P0
*0004  < JMOV P1
COPY     MOVE     DEL   JUMP
```

SPACE B

Block setting completed

Step 6. **Move to STEP BLOCK**

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0002    LABL A0
 0003  < JMOV P0
 0004  < JMOV P1
 0005    DLAY 100
*0006    GOTO A0
COPY     MOVE     DEL   JUMP
```

▼

Move a cursor (*) to position to move to

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0002    LABL A0
 0003  < JMOV P0
 0004  < JMOV P1
 0005    DLAY 100
*0006    GOTO A0
COPY     MOVE     DEL   JUMP
```

F2

Execute MOVE

Step 7. **MOVE completed**

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0002    LABL A0
 0003    DLAY 100
*0004    JMOV P0
 0005    JMOV P1
 0006    GOTO A0
COPY     MOVE     DEL   JUMP
```

ENTER completed

▶ For Program update after MOVE is completed, refer to 3-60 page, "(3) Program updating".

3.4.5 **Program updating**

▶ How to update Program after Program Edit is completed. (COPY as an example)

Step 1. **Copy completed screen**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
COPY    MOVE     DEL   JUMP
```

Copy completed

Step 2. **Update fashion**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
INS     EDIT    JUMP   QUIT
```

Select ESC

ESC

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
INS     EDIT    JUMP   QUIT
```

Select QUIT

F4

Step 3. **Update**

```
<RGA80A : EDIT>
 STEP   INS    OP1   OP2
 0002   LABL A0
 0003   JMOV P0
 0004   JMOV P1
 0005   DLAY 100
*0006   JMOV P0
Update OK? (ENT/ESC)
```

Select ENTER

ENTER

```
<RGA80A : EDIT>
 STEP    INS     OP1    OP2
 0002    LAVL A0
 0003    JMOV P0
 0004    JMOV P1
 0005    DLAY 100
*0006    JMOV P0
Update Complete !
```

ENTER

Update

```
<RGA80A : EDIT>
 JOB NAME = TEST1




 PROG    POINT    PLC
```

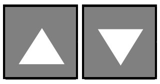
Update completed

3.4.6 Point Edit (position change)

▶ E.g.) Editing Point of "GOLD" file

Step 1. **Select Program**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .__             .__
    .__             .__
    .__             .__
    .__             .__
 COPY    REN      DEL      EDIT
```



Select the desired program by marking *.

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .__             .__
    .__             .__
    .__             .__
    .__             .__
 COPY    REN      DEL      EDIT
```

F4

Select EDIT

Step 3. **Select POINT**

```
<RGA80A : EDIT>
 JOB NAME = GOLD



 PROG    POINT    PLC
```

F2

Select POINT

Step 4. **MODE select screen**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B   L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI      CURR       EDIT   QUIT
```

Mode selection screen

Step 5-1. **MDI Mode**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B   L
A: 89.69             B : 2.83
Z: 59.72             W: 22.84


MDI      CURR      EDIT   QUIT
```

If in MDI Mode:

F1

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B   L
A: A 89.69           B : 2.83
Z: 59.72             W: 22.84


EXCH     CORD PJUMP FWRD
```

MDI Mode screen

Step 5-2. **CURR Mode**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B   L
A: 89.69             B : 2.83
Z: 59.72             W: 22.84


MDI      CURR      EDIT   QUIT
```

If in CURR Mode:

F2

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B   L
A: 0                 B : 0
Z: 0                 W: 0


EXCH     CORD PJUMP FWRD
```

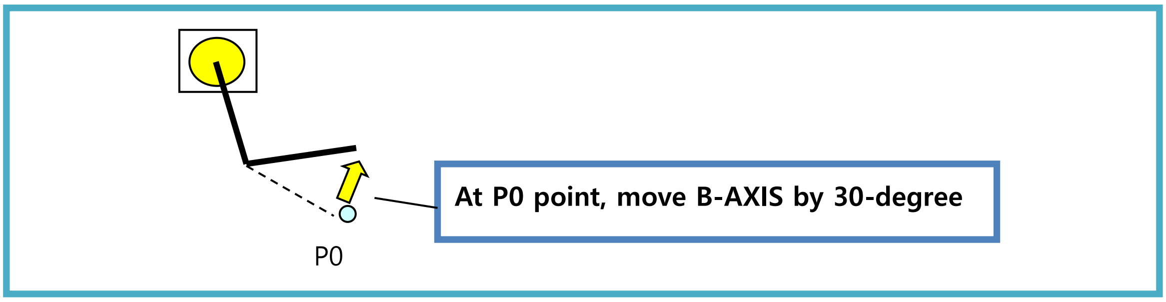
CURR Mode screen

- ▶ For MDI Mode, see 3.4.6.1 MDI (Manual Direct Input) Mode (3-64 page)
- ▶ For CURR Mode, see 3.4.6.2 CURR (Current) Mode (3-69 page)

3.4.6.1 MDI (Manual Direct Input) Mode

1. Point editing in ANGLE coordinate system
2. Point editing in Cartesian coordinate system

① Point editing in Angle coordinate system



Step 1. **Point teaching screen**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B L
A: 89.69             B : 2.83
Z: 59.72             W: 22.84


MDI      CURR       EDIT   QUIT
```

Open Point teaching screen

Step 2. **MDI Mode**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B L
A: 89.69             B : 2.83
Z: 59.72             W: 22.84


MDI      CURR       EDIT   QUIT
```

F1

Select MDI

Step 3. **Change MDI point**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B L
A: A 89.69           B : 2.83
Z: 59.72             W: 22.84


EXCH     CORD PJUMP FWRD
```

Open MDI setup screen

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : A 2.83
Z: 59.72            W: 22.84

EXCH    CORD PJUMP FWRD
```

Move right

▶

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 30.83A
Z: 59.72            W: 22.84

EXCH    CORD PJUMP FWRD
```

Enter B-AXIS value to be changed

3 S | 0 V | . W | 8 H | 3 S

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 30.83
Z: A59.72           W: 22.84

EXCH    CORD PJUMP FWRD
```

Select ENTER

ENTER

Step 4. **Temporary update of point**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 30.83
Z: A59.72           W: 22.84

EXCH    CORD PJUMP FWRD
```

Select EXCH

F1

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 30.83
Z: A59.72           W: 22.84

Update OK? (ENT/ESC)
```

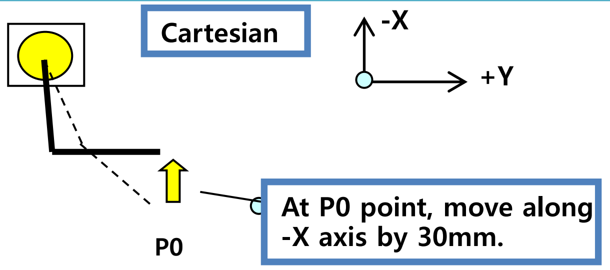
Temporarily update the changed point

ENTER

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: A89.69           B : 30.83
Z: 59.72            W: 22.84

EXCH    CORD PJUMP FWRD
```

Temporary update is completed

② Point editing in Cartesian coordinate system



Step 1. **Point teaching screen**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI      CURR       EDIT   QUIT
```

Open Point teaching screen

Step 2. **MDI Mode**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI      CURR       EDIT   QUIT
```

F1

Select MDI

Step 3. **Set up MDI point**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: A 89.69          B : 2.83
Z: 59.72            W: 22.84


EXCH     CORD PJUMP FWRD
```

Open MDI setup screen

Step 4. **Convert to Cartesian coordinate system**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0     US   B L
A: A 89.69          B : 2.83
Z: 59.72            W: 22.84


EXCH     CORD PJUMP FWRD
```

F2

Select CORD. (Convert to Cartesian coordinate system)

Step 5. **MDI point change**

```
<RGA80A : EDIT> V :          50
F: GOLD     P: 0        US    B L
X: 483.62               Y : 309.82
Z: 59.72                W: 55.35

EXCH     CORD PJUMP FWRD
```

Enter X-AXIS value to be changed

[X-AXIS value=513.62+(-30)=483.62]

4	8	3
L	H	S
.	6	2
W	N	R

```
<RGA80A : EDIT> V :          50
F: GOLD     P: 0        US    B L
X: 483.62               Y : A309.82
Z: 59.72                W: 55.35

EXCH     CORD PJUMP FWRD
```

Select ENTER

ENTER

Step 6. **Temporary update of point**

```
<RGA80A : EDIT> V :          50
F: GOLD     P: 0        US    B L
X: 483.62               Y : A309.82
Z: 59.72                W: 55.35

EXCH     CORD PJUMP FWRD
```

Select EXCH

F1

```
<RGA80A : EDIT> V :          50
F: GOLD     P: 0        US    B L
X: 483.62               Y : A309.82
Z: 59.72                W: 55.35

Update OK? (ENT/ESC)
```

Temporarily update the changed point

ENTER

Step 7. **Select robot form**

```
<RGA80A : EDIT> V :     50L
F: GOLD     P: 0      US     B     L
X: 483.62             Y : A309.82
Z: 59.72              W: 55.35

LEFT Form OK?
```

Select robot form (see the following description)

ROBOT ARM : LEFT form

```
<RGA80A : EDIT> V :    50R
F: GOLD    P: 0       US   B L
X: 483.62             Y : A309.82
Z: 59.72              W: 55.35

RIGHT Form OK?
```

Select robot form (see the following description)

ROBOT ARM : RIGHT form

Step 8. **Select robot form**

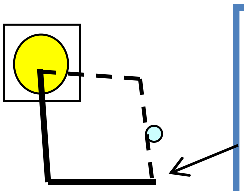
```
<RGA80A : EDIT> V :         50
F: GOLD    P: 0       US   B L
X: 483.62             Y : A309.82
Z: 59.72              W: 55.35

EXCH    CORD PJUMP FWRD
```

ENTER

Select RIGHT form

■ Comment

▷ Selection of robot form



- In SCARA, there are 2 forms for 1 point.
  Dashed line: LEFT form
  Solid line: RIGHT form
- To use Cartesian coordinate in MDI Point teaching, ARM shape (LEFT or RIGHT form) is to be set up.

Important

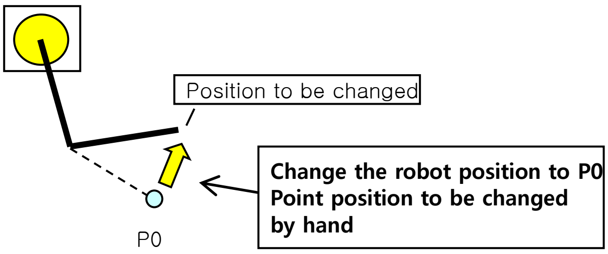
- EXCH is that P0 point is temporarily updated in a point file.
- If the current screen is powered out, the previous value before update is stored in P0 point.

3.4.6.2 CURR (Current) Mode

3.4.6.3 Joint coordinate system and XY coordinate system

1. Move the robot to the position to be changed by hand, and update it.
2. Move the robot to the position to be changed by using Jog or Inch key, and update it.

① Move the robot to the position to be changed by hand, and update it.



▶ Keys involved when is moved by hand:

BRK / F	Used to forcefully turn on/off a brake for each axis. (Going down along Z-axis is possible.
SVON / E	Used to forcefully turn on/off a servo motor for each axis. (Checkable by an indication lamp of teaching pendant.)

Initial position

```
F:GOLD P:0 US B L
A:29.69          B:2.83
Z:59.72          W:22.84
EXCH CORD PJUMP FWRD
```

Changed position

```
F:GOLD P:0 US B L
A:29.69          B:10.83
Z:59.72          W:22.84
EXCH CORD PJUMP FWRD
```

- In CURR Mode, the current Position of the robot is always displayed.
- The coordinate value of a teaching pendant is changed by manipulating JOG key of the teaching pendant

Step 1. **Point teaching screen**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0     US   B L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI     CURR     EDIT   QUIT
```

Open Point teaching screen

Step 2. **CURR Mode**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0     US   B L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI     CURR     EDIT   QUIT
```

F2

Select CURR

Step 3. **Change CURR point**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0        US    B L
A: 89.69               B : 2.83
Z: 59.72               W: 22.84


EXCH    CORD PJUMP FWRD
```

CURR point screen before changed

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0        US    B L
A: 89.69               B : 10.83
Z: 59.72               W: 22.84


EXCH    CORD PJUMP FWRD
```

SVON
E

If a robot is moved by Servo OFF (to move it by hand), the coordinate value changes.

Step 4. **Temporary update of point**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0        US    B L
A: 89.69               B : 10.83
Z: 59.72               W: 22.84


EXCH    CORD PJUMP FWRD
```

F1

Select EXCH

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0        US    B L
A: 89.69               B : 10.83
Z: 59.72               W: 22.84


Update OK? (ENT/ESC)
```

ENTER

Temporarily update the changed point.

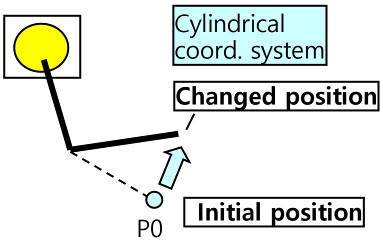
```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0        US    B L
A: 89.69               B : 10.83
Z: 59.72               W: 22.84


EXCH    CORD PJUMP FWRD
```

Temporary update is completed

▶ For final updating the point, refer to 3.4.8 Point Updating (3-93**오류! 책갈피가 정의되어 있지 않습니다.** page)

② Move the robot to the position to be changed by using Jog or Inch key, and update it.

JOG key based motion



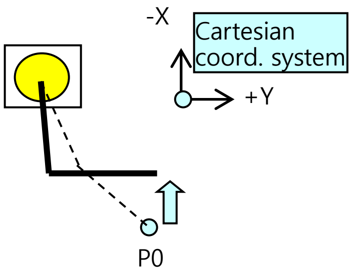
Initial position

F:GOLD P:0 US B M
A:29.69 B:2.83
Z:59.72 W:22.84
EXCH ORD PJUMP FWRD

Changed position

F:GOLD P:0 US B M
A:29.69 B:10.83
Z:59.72 W:22.84
EXCH CORD PJUMP FWRD

← B →
O P



Initial position

F:GOLD P:0 US B M
X:513.62 Y:509.82
Z:59.72 W:55.35
EXCH CORD PJUMP FWRD

Changed position

F:GOLD P:0 US B M
X:483.62 Y:509.82
Z:59.72 W:55.35
EXCH CORD PJUMP FWRD

← A →
J K

▶ **Manipulation keys related to JOG & INCH:**

▶ Axis(A,B,Z,W) motion keys:

← A →		← B →		← Z →		← W →	
J	K	O	P	T	U	Y	Z

- ANGLE coordinate :
Move for a unit distance along the relevant axis while the key is being pressed one time (PTP motion).
- Cartesian coordinate :
Interpolated motion for a unit distance along the relevant axis when the key is pressed one time
(★ for SCARA robot only)

▶ Servo ON/OFF switching key:

Servo OFF

F:GOLD P:0 NW B L
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

SVON
E

Servo ON

F:GOLD P:0 NW L
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

`

VEL
C

F:GOLD P:0 NW B L
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

5% velocity of Jv
25 rpm

F:GOLD P:0 NW B M
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

50% velocity of Jv
250 rpm

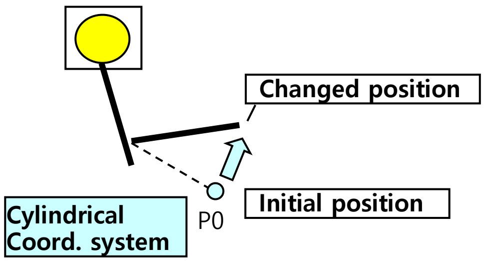
F:GOLD P:0 NW B H
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

100% velocity of Jv
500 rpm
10- degree

Joint Motion

JOINT MOTION: X
Mv:3000 At:0.30
Jv :500 Ip:10
It :10 Jk: 50

INCH key based motion



Initial position

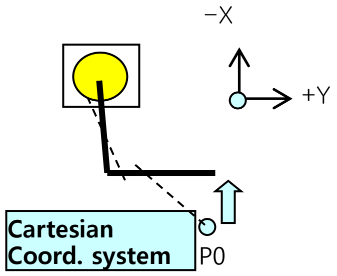
F:GOLD P:0 US B m
A:29.69 B:0.83
Z:59.72 W:22.84
EXCH ORD PJUMP FWRD

Changed position

F:GOLD P:0 US B m
A:29.69 B:1.83
Z:59.72 W:22.84
EXCH CORD PJUMP FWRD

← B	
O	P

Unit moving distance(It): 10
l : 5% It → 0.5-degree
m: 50% It → 5-degree
h: 100% It → 10-degree



Initial position

F:GOLD P:0 US B m
X:513.62 Y:509.82
Z:59.72 W:55.35
EXCH CORD PJUMP FWRD

Changed position

F:GOLD P:0 US B m
X:483.62 Y:509.82
Z:59.72 W:55.35
EXCH CORD PJUMP FWRD

A →	
J	K

Unit moving distance(It): 10
l : 5% Ia→ 0.5mm
m: 50% Ia → 5mm
h: 100% Ia→ 10mm

▶ Manipulation keys related to JOG & INCH:

▶ Axis(A,B,Z,W) moving keys:

← A →		← B →		← Z →		← W →	
J	K	O	P	T	U	Y	Z

- ANGLE coordinate :
Move for a unit distance along the relevant axis while the key is being pressed one time (PTP motion).
- Cartesian coordinate :
Interpolated motion for a unit distance along the relevant axis when the key is pressed one time
(★ for SCARA robot only)

▶ Servo ON/OFF switching key:

Servo OFF

F:GOLD P:0 NW B l
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

SVON
E

Servo ON

F:GOLD P:0 NW l
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP FWRD

▶ Unit distance changing key for INCH Mode:

Cylindrical coord. system

VEL
C

F:GOLD P:0 NW B l
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP

5% distance of It
0.5 degree

F:GOLD P:0 NW B m
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP

50% distance of It
5 degree

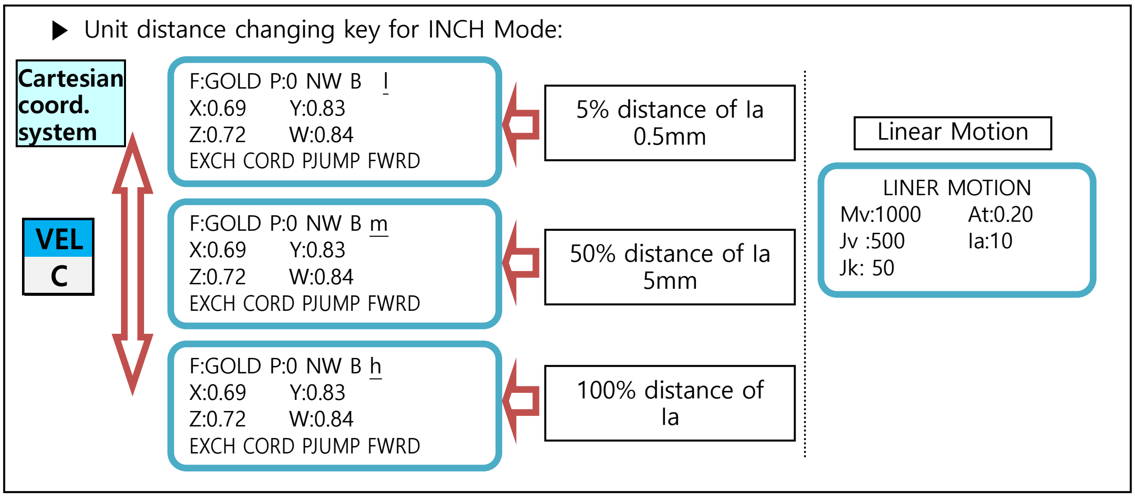
F:GOLD P:0 NW B h
A:0.69 B:0.83
Z:0.72 W:0.84
EXCH CORD PJUMP

100% distance of It

Joint Motion

JOINT MOTION: X
Mv:3000 At:0.30
Jv :500 Ip:10
It :10 Jk: 50

INCH key based motion



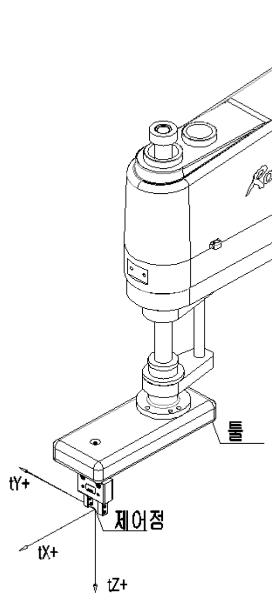
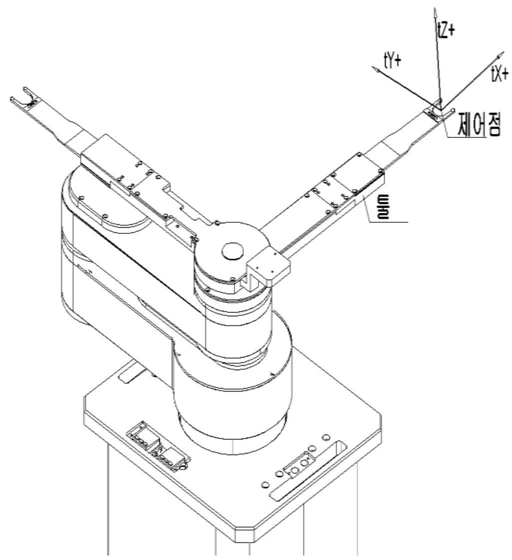
CAUTION

- The robot mechanism may slightly shake during Servo ON/OFF.
- In INCH motion for moving the fine distance, if the robot is continuously driven without Servo ON not activated, the positional error may occur.
- Be sure to perform INCH motion under the state of Servo ON activated.

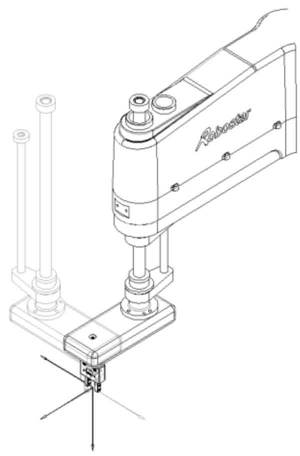
3.4.6.4 TOOL Coordinate System

With a TOOL parameter configured by user, select a TOOL coordinate system from the teaching pendant and it is consistent with the control point of the actually-installed TOOL. When pressing the jog key, the robot operates with the control point at the center.

<This function is available for use only in version that came after ROSEP Robot and Ver 03.00.0.>

	
SCARA ROBOT TYPE	ROSEP ROBOT TYPE

In case an error in the control point is great when rotating with the control point of TOOL at the center in the TOOL coordinate system, it is necessary to identify the set value of TOOL parameter.



[W-axis Rotation on TOOL Coordinate System]

Step 1. **Point Teaching Screen**

```
<ROSEP200: EDIT>            V: 50
F: GOLD     P: 0      US     B   L
A   : -27.08          B   : 124.04
Z   : 9.995           W1: -6.955
W2: -6.955

MDI      CURR      EDIT      QUIT
```

Open the point teaching screen.

Step 2. **CURR Mode**

```
<ROSEP200: EDIT>           V: 50
F: GOLD    P: 0     US   B   L
A  : -27.08         B  : 124.04
Z  : 9.995          W1: -6.955
W2: -6.955

MDI     CURR     EDIT     QUIT
```

F2

Select CURR.

Step 3. **CURR Mode**

```
<ROSEP200: EDIT>           V: 50
F: GOLD    P: 0     US   B   L
A  : -27.08         B  : 124.04
Z  : 9.995          W1: -6.955
W2: -6.955

MDI     CURR     EDIT     QUIT
```

Check CURR point.

Step 4. **Change XY coordinate system**

```
<ROSEP200: EDIT>           V: 50
F: GOLD P: 0        US   B   L
X : 215.36          Y : 550.45
Z : 59.72          RW : 22.84
W: 22.84

EXCH    CORD    PJUMP    FWRD
```

F2

Change XY coordinate system

Step 5. **XY coordinate system**

```
<ROSEP200: EDIT>           V: 50
F: GOLD P: 0        US   B   L
X : 215.36          Y : 550.45
Z : 59.72          RW : 22.84
W: 22.84

EXCH    CORD    PJUMP    FWRD
```

CURR XY coordinate system screen

Step 6. **Change TOOL JOG mode**

```
<ROSEP200: EDIT>              V: 50
F: GOLD P: 0          US     B   L
X   : 215.36          Y   : 550.45
Z   : 59.72           RW : 22.84
W2: 22.84


EXCH     CORD   PJUMP   FWRD
```

F6

Change CURR TOOL JOG mode

Step 7. **TOOL JOG mode**

```
<ROSEP200: EDIT>              V: 50
F: GOLD      P: 0     US     B   L
X   : 215.36          Y   : 550.45
Z   : 59.72           RW : 22.84
W2: 22.84
          < TOOL JOG MODE>

EXCH     CORD   PJUMP   FWRD
```

CURR TOOL JOG mode screen

Step 8. **TOOL JOG operation**

```
<ROSEP200: EDIT>              V: 50
F: GOLD      P: 0     US     B   L
X   : 215.36          Y   : 550.45
Z   : 59.72           RW : 22.84
W2: 22.84
          < TOOL JOG MODE>

EXCH     CORD   PJUMP   FWRD
```

← **J1** →

~

← **J6** →

Move to the desired position by pressing JOG KEY on TOOL coordinate system.

Caution

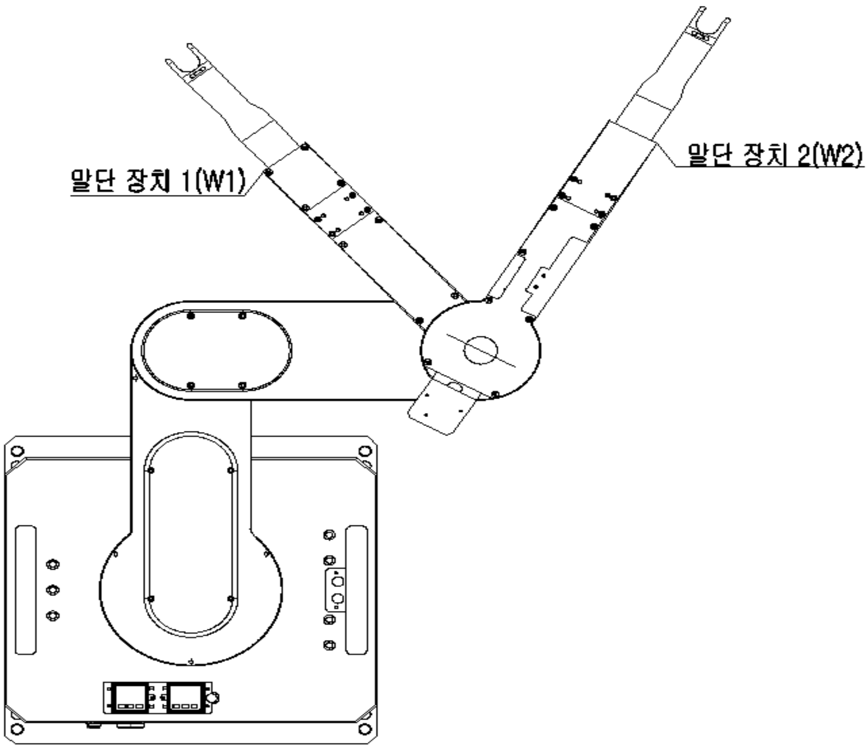
▶ If an accurate JOG operation is found on TOOL coordinate system, check if actually-mounted TOOL and parameter set value are consistent.

▶ The TOOL number applied when saving the current position during JOG operation on the TOOL coordinate system should be identical to the one applied to AUTO MODE and FORWARD. If not identical, the robot's location and posture do not move to the location and posture when saved.

3.4.6.5 End Effector Change

Where there are several end effectors while running JOG, it is necessary to determine which end effector is to primarily be used for moving them in XY direction on the TOOL coordinate system before making subsequent changes.

<This function is applicable only in version that came after ROSEP Robot and Ver 03.00.00.>



On the TOOL coordinate system, use **INT D** key to modify the end effector.

The modified axis has a 'R' letter added to it on the teaching pendant.

<ROSEP200: EDIT> V: 50 F: GOLD P: 0 US B L X : 215.37 Y : 550.45 Z : 9.99 RW : 72.96 W2: 90.00 < TOOL JOG MODE> EXCH CORD PJUMP FWRD	<ROSEP200: EDIT> V: 50 F: GOLD P: 0 US B L X : 98.14 Y : 568.01 Z : 9.99 W1 : 72.96 RW: 90.00 < TOOL JOG MODE> EXCH CORD PJUMP FWRD
when selecting end effector 1(W1)	When selecting end effector 2(W2)

Step 1. **Point Teaching Screen**

```
<ROSEP200: EDIT>          V: 50
F: GOLD     P: 0      US    B   L
A   : -9.96          B   : 139.33
Z   : 9.99          W1: -56.407
W2: -39.36

MDI      CURR     EDIT     QUIT
```

Open point teaching screen.

Step 2. **CURR Mode**

```
<ROSEP200: EDIT>          V: 50
F: GOLD     P: 0      US    B   L
A   : -9.96          B   : 139.33
Z   : 9.99          W1: -56.407
W2: -39.36

MDI      CURR     EDIT     QUIT
```

F2

Select CURR.

Step 3. **CURR Mode**

```
<ROSEP200: EDIT>          V: 50
F: GOLD     P: 0      US    B   L
A   : -9.96          B   : 139.33
Z   : 9.99          W1: -56.407
W2: -39.36

MDI      CURR     EDIT     QUIT
```

CURR point screen

Step 4. **XY Coordinate System Change**

```
<ROSEP200: EDIT>          V: 50
F: GOLD     P: 0      US    B   L
A   : -9.96          B   : 139.33
Z   : 9.99          W1: -56.407
W2: -39.36

MDI      CORD     EDIT     QUIT
```

F2

Select XY coordinate system.

Step 5. **XY Coordinate System**

```
<ROSEP200: EDIT>          V: 50
F: GOLD     P: 0      US    B   L
X : 215.37            Y : 550.45
Z : 9.99            RW : 72.96
W2: 90.00

EXCH     CORD    PJUMP    FWRD
```

CURR XY point screen

Step 6. **TOOL JOG Mode Change**

```
<RGA80A : EDIT> V :        50
F: GOLD     P: 0       US     B    L
X   : 215.37          Y   : 550.45
Z   : 9.99            RW: 72.96
W2 : 90.00


EXCH    CORD   PJUMP   FWRD
```

F6

Change TOOL JOG mode.

Step 7. **TOOL JOG Mode**

```
<RGA80A : EDIT> V :        50
F: GOLD     P: 0       US     B    L
X : 215.37              Y : 550.45
Z : 9.99            RW : 72.96
W2: 90.00
        < TOOL JOG MODE>

EXCH    CORD   PJUMP   FWRD
```

Screen with selected end effector 1.

Step 8. **End Effector Change**

```
<RGA80A : EDIT> V :        50
F: GOLD     P: 0       US     B    L
X   : 215.37          Y   : 550.45
Z   : 9.99            RW : 72.96
W2 : 90.00
        < TOOL JOG MODE>

EXCH    CORD   PJUMP   FWRD
```

INT
D

Change end effector.

Step 9. **End Effector 2 Selection**

```
<ROSEP200: EDIT>              V: 50
F: GOLD     P: 0       US     B    L
X   : 98.14             Y   : 568.01
Z   : 9.99              W1 : 72.96
RW: 90.003
        < TOOL JOG MODE>

EXCH    CORD   PJUMP   FWRD
```

Screen with selected end effector 2.

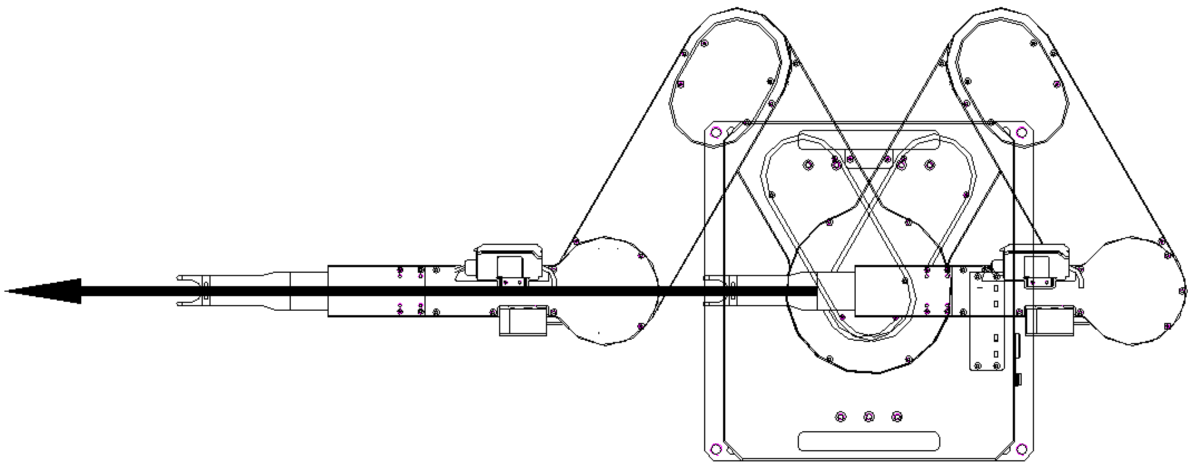
Caution

▶ The end effector applied when saving the current position should be identical to the one applied to AUTO MODE and FORWARD. If not identical, the robot's location and posture do not move to the location and posture when saved.

3.4.6.6 JMOV Synchronous Movement

When making a linear interpolation motion (LMOV) near a singularity, an overspeed alarm occurs due to the sudden speed change near the singularity. To resolve this problem, a ROSEP robot type operates PTP motion (JMOV) with a synchronization to enable a linear motion passing through the singularity.

<This function is available for use only in version that came after ROSEP Robot and Ver 03.00.00.>



[ROSEP ROBOT TYPE Passing through Singularity]

Step 1. **Point Teaching Screen**

```
<ROSEP200: EDIT>            V: 50
F: GOLD     P: 0      US    B   L
A   : 45.00           B   : 90.00
Z   : 10.0            W1: -45.00
W2:-45.00

MDI      CURR      EDIT      QUIT
```

Open point teaching screen.

Step 2. **CURR Mode**

```
<ROSEP200: EDIT>            V: 50
F: GOLD     P: 0      US    B   L
A   : 45.00           B   : 90.00
Z   : 10.0            W1: -45.00
W2:-45.00

MDI      CURR      EDIT      QUIT
```

F2

Select CURR.

Step 3. **CURR Mode**

```
<ROSEP200: EDIT>              V: 50
F: GOLD     P: 0        US    B   L
A   : 45.00             B   : 90.00
Z   : 10.0              W1: -45.00
W2:-45.00


EXCH    CORD   PJUMP   FWRD
```

CURR point screen

Step 4. **JNTSYN MODE Change**

```
<ROSEP200: EDIT>              V: 50
F: GOLD     P: 0        US    B   L
A   : 45.00             B   : 90.00
Z   : 10.0              W1: -45.00
W2:-45.00

EXCH    CORD   PJUMP   FWRD
```

F6

Change JNTSYN mode

Step 5. **JNTSYN MODE**

```
<ROSEP200: EDIT>              V: 50
F: GOLD     P: 0        US    B   L
A   : 45.00             B   : 90.00
Z   : 10.0              W1: -45.00
W2:-45.00
       <J1 SYNC JOG MODE>

EXCH    CORD   PJUMP   FWRD
```

JNTSYN mode screen

Step 8. **JNTSYNC JOG Operation**

```
<ROSEP200: EDIT>              V: 50
F: GOLD     P: 0        US    B   L
A   : 45.00             B   : 90.00
Z   : 10.0              W1: -45.00
W2:-45.00

       <J1 SYNC JOG MODE>
EXCH    CORD   PJUMP   FWRD
```

←	J1	→

Press JOG KEY to move to the desired position.

Caution

▶ In JNTSYC mode, jog can move only in "J1 " direction.

3.4.7 **Point Copy, Move & Delete**

■ COPY ---- Copy the saved point

- Copy as new point number (See page 3-82)
- Copy as existing point number (See page 3-86)

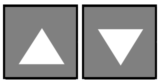
3.4.7.1 Copy as new point number

▶ E.g.) Copy P0 (USE) → P1 (NEW)

Step 1. **Select Program**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
   ._               ._
   ._               ._
   ._               ._
   ._               ._
 COPY     REN      DEL      EDIT
```

Select the desired program by marking *.



Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
   ._               ._
   ._               ._
   ._               ._
   ._               ._
 COPY     REN      DEL      EDIT
```

Select EDIT

F4

Step 3. **Move to POINT**

```
<RGA80A : EDIT>
 JOB NAME = GOLD



 PROG    POINT    PLC
```

Select POINT

F2

Step 4. **Edit POINT**

```
<RGA80A : EDIT> V :        50
F: GOLD    P: 0      US   B L
A: 89.69             B : 2.83
Z: 59.72             W: 22.84


MDI      CURR       EDIT   QUIT
```

F3

Select EDIT

Step 5. **Select saved point**

```
<RGA80A : EDIT>
F: GOLD
*P0        US   A   P1        ——
 P2        ——       P3        ——
 P4        ——       P5        ——
 P6        ——       P7        ——
 P8        ——       P9        ——
COPY     MOVE      DEL   PJUMP
```

ENTER

Select P0

Step 6. **Select copy**

```
<RGA80A : EDIT>
F: GOLD
*P0        US   A   P1        ——
 P2        ——       P3        ——
 P4        ——       P5        ——
 P6        ——       P7        ——
 P8        ——       P9        ——
COPY     MOVE      DEL   PJUMP
```

F1

Select COPY

Step 7. **Select copy target**

```
<RGA80A : EDIT>
F: GOLD
*P0        US       P1        ——
 P2        ——       P3        ——
 P4        ——       P5        ——
 P6        ——       P7        ——
 P8        ——       P9        ——
Point # = 1 A
```

1 Q ENTER

Copy to point No. 1

```
<RGA80A : EDIT>
F: GOLD
*P0        US       P1        ——
 P2        ——       P3        ——
 P4        ——       P5        ——
 P6        ——       P7        ——
 P8        ——       P9        ——
Copy Complete
```

ENTER

Confirm completed copy

```
<RGA80A : EDIT>
F: GOLD
*P0      US   A   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
COPY    MOVE      DEL   PJUMP
```

Copy to P1 is completed

Step 8. **Final update**

```
<RGA80A : EDIT>
F: GOLD
*P0      US   A   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
COPY    MOVE      DEL   PJUMP
```

Final update to point file

ESC

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0    US    B L
A: 89.69           B : 2.83
Z: 59.72           W: 22.84



MDI      CURR      EDIT   QUIT
```

Select QUIT

F4

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0    US    B L
A: 89.69           B : 2.83
Z: 59.72           W: 22.84



Update OK? (ENT/ESC)
```

Execute update

ENTER

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0    US    B L
A: 89.69           B : 2.83
Z: 59.72           W: 22.84



Update Complete !
```

Update completed

ENTER

```
<RGA80A : EDIT>
 JOB NAME  =  GOLD





 PROG    POINT    PLC
```

Update completed

■ Setting the point block

```
<RBSA80A   :EDIT>
 F:GOLD
■P0 US      P1 US
  P2 __      P3 __
  P4 __      P5 __
  P6 __      P7 __
  P8 __      P9 __
COPY MOVE DEL PJUMP
```

```
<RBSA80A   :EDIT>
 F:GOLD
 *P0 US     ■P1 US
  P2 __      P3 __
  P4 __      P5 __
  P6 __      P7 __
  P8 __      P9 __
COPY MOVE DEL PJUMP
```

```
<RBSA80A   :EDIT>
 F:GOLD
 *P0 US      *P1 US
■P2 __       P3 __
  P4 __       P5 __
  P6 __       P7 __
  P8 __       P9 __
COPY MOVE DEL PJUMP
```

① Using the left key, move ■ to a first point of the block to be set. (Use PJUMP)

② ENTER If Enter key is pressed, * is displayed at the front of the point number.
Select continuously the necessary points.

▶ The set block can be repetitively used before it is unset.

3.4.7.2 Copy as existing point number

▶ E.g.) Copy P0 (USE) → P1 (USE)

Step 1. **Program**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .—              .—
    .—              .—
    .—              .—
    .—              .—
 COPY    REN     DEL     EDIT
```

Select the desired program by marking *.

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .—              .—
    .—              .—
    .—              .—
    .—              .—
 COPY    REN     DEL     EDIT
```

F4

Select EDIT

Step 3. **Move to POINT**

```
<RGA80A : EDIT>
 JOB NAME  =  GOLD



 PROG    POINT    PLC
```

F2

Select POINT

Step 4. **Edit POINT**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0    US    B L
A: 89.69           B : 2.83
Z: 59.72           W: 22.84


MDI     CURR     EDIT   QUIT
```

F3

Select EDIT

Step 5. **Select the saved point**

```
<RGA80A : EDIT>
F: GOLD
*P0     US  A  P1      US
 P2     ――     P3      ――
 P4     ――     P5      ――
 P6     ――     P7      ――
 P8     ――     P9      ――
COPY   MOVE    DEL  PJUMP
```

ENTER

Select P0

Step 6. **Select copy**

```
<RGA80A : EDIT>
F: GOLD
*P0     US  A  P1      US
 P2     ――     P3      ――
 P4     ――     P5      ――
 P6     ――     P7      ――
 P8     ――     P9      ――
COPY   MOVE    DEL  PJUMP
```

F1

Select COPY

Step 7. **Select the position & Copy**

```
<RGA80A : EDIT>
F: GOLD
*P0     US     P1      US
 P2     ――     P3      ――
 P4     ――     P5      ――
 P6     ――     P7      ――
 P8     ――     P9      ――
Point # = 1 A
```

1 Q ENTER

Copy to point No. 1

```
<RGA80A : EDIT>
F: GOLD
*P0     US     P1      US
 P2     ――     P3      ――
 P4     ――     P5      ――
 P6     ――     P7      ――
 P8     ――     P9      ――
Delete OK? (ENT/ESC)
```

ENTER

Delete the existing P1 point

```
<RGA80A : EDIT>
F: GOLD
*P0     US     P1      US
 P2     ――     P3      ――
 P4     ――     P5      ――
 P6     ――     P7      ――
 P8     ――     P9      ――
Copy Complete !
```

ENTER

Execute copy

Step 8. **Copy completed**

```
<RGA80A : EDIT>
F: GOLD
*P0     US      P1      US
 P2     ——      P3      ——
 P4     ——      P5      ——
 P6     ——      P7      ——
 P8     ——      P9      ——
COPY    MOVE    DEL   PJUMP
```

Copy completed

▶ For final updating the point, refer to 3.4.8 Point Updating (3-93**오류! 책갈피가 정의되어 있지 않습니다.** page)

■ MOVE ---- Move point

E.g.) Move P0, P1 (USE) → P2, P3

Step 1. **Select Program**

```
<RGA80A : DIR>
  0. TEST1         8   STEP
 *1. GOLD          5   STEP
   .__             .__
   .__             .__
   .__             .__
   .__             .__
 COPY    REN     DEL      EDIT
```

Select the desired program by marking *.

Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
  0. TEST1         8   STEP
 *1. GOLD          5   STEP
   .__             .__
   .__             .__
   .__             .__
   .__             .__
 COPY    REN     DEL      EDIT
```

F4

Select EDIT

Step 3. **Move to POINT**

```
<RGA80A : EDIT>
 JOB NAME = GOLD




 PROG    POINT    PLC
```

F2

Select POINT

Step 4. **Edit POINT**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0    US     B L
A: 89.69           B : 2.83
Z: 59.72           W: 22.84



MDI      CURR      EDIT   QUIT
```

F3

Select EDIT

Step 5. **Set point block**

```
<RGA80A : EDIT>
F: GOLD
*P0      US   *   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
COPY    MOVE     DEL   PJUMP
```

ENTER ENTER

Refer to “Point block setting”

Step 6. **Select move target & Execute**

```
<RGA80A : EDIT>
F: GOLD
*P0      US   *   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
COPY    MOVE     DEL   PJUMP
```

F2

Select MOVE

```
<RGA80A : EDIT>
F: GOLD
*P0      US   *   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
Point # = 2 A
```

2 R ENTER

Enter point number to be moved

```
<RGA80A : EDIT>
F: GOLD
*P0      US   *   P1       US
 P2      ——       P3       ——
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
Move Complete !
```

ENTER

Execute move

Step 7. **Move completed**

```
<RGA80A : EDIT>
F: GOLD
 P0      ——       P1       ——
 P2      US       P3       US
 P4      ——       P5       ——
 P6      ——       P7       ——
 P8      ——       P9       ——
COPY    MOVE     DEL   PJUMP
```

Move is completed

▶ For updating point after Move, refer to page 3-93, "3.4.8 Point Updating".

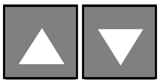
■ DEL ---- Delete point

E.g.) Delete P0, P1 (USE)

Step 1. **Select Program**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .__             .__
    .__             .__
    .__             .__
    .__             .__
 COPY    REN     DEL      EDIT
```

Select the desired program by marking *.



Step 2. **Select JOB EDIT**

```
<RGA80A : DIR>
  0. TEST1          8   STEP
 *1. GOLD           5   STEP
    .__             .__
    .__             .__
    .__             .__
    .__             .__
 COPY    REN     DEL      EDIT
```

Select EDIT

F4

Step 3. **Move to POINT**

```
<RGA80A : EDIT>
 JOB NAME = GOLD



 PROG    POINT    PLC
```

Select POINT

F2

Step 4. **Edit POINT**

```
<RGA80A : EDIT> V :    50
F: GOLD    P: 0     US     B L
A: 89.69            B : 2.83
Z: 59.72            W: 22.84


MDI      CURR      EDIT   QUIT
```

Select EDIT

F3

Step 5. **Set point block**

```
<RGA80A : EDIT>
F: GOLD
*P0      US  *  P1       US
 P2      ——     P3       ——
 P4      ——     P5       ——
 P6      ——     P7       ——
 P8      ——     P9       ——
COPY    MOVE    DEL   PJUMP
```

ENTER ENTER

See “Point block setting”

Step 6. **Select DEL & Execute**

```
<RGA80A : EDIT>
F: GOLD
*P0      US  *  P1       US
 P2      ——     P3       ——
 P4      ——     P5       ——
 P6      ——     P7       ——
 P8      ——     P9       ——
COPY    MOVE    DEL   PJUMP
```

F3

Select DEL

```
<RGA80A : EDIT>
F: GOLD
*P0      US  *  P1       US
 P2      ——     P3       ——
 P4      ——     P5       ——
 P6      ——     P7       ——
 P8      ——     P9       ——
Delete OK? (ENT/ESC)
```

ENTER

Execute delete

Step 7. **Delete completed**

```
<RGA80A : EDIT>
F: GOLD
 P0      ——     P1       ——
 P2      ——     P3       ——
 P4      ——     P5       ——
 P6      ——     P7       ——
 P8      ——     P9       ——
COPY    MOVE    DEL   PJUMP
```

Delete completed

▶ For updating point after Delete, refer to 3-93 page, "3.4.8 Point updating”.

3.4.8 Point Updating

Be sure to update the point before teaching the robot (MDI, CURR) or performing copy, move, or delete on a basis of block.

▶ **How to update the point when in robot teaching, and block basis copy, move, or delete**

Step 1. **Final update of point**

<RGA80A : EDIT> V : 50
F: GOLD P: 0 US B L
A: 89.69 B : 2.83
Z: 59.72 W: 22.84

MDI CURR EDIT QUIT

F4

In USED point file, select QUIT

<RGA80A : EDIT> V : 50
F: GOLD P: 0 US B L
A: 89.69 B : 2.83
Z: 59.72 W: 22.84

Update OK? (ENT/ESC)

ENTER

Execute update

<RGA80A : EDIT> V : 50
F: GOLD P: 0 US B L
A: 89.69 B : 2.83
Z: 59.72 W: 22.84

Update Complete !

ENTER

Update message is displayed

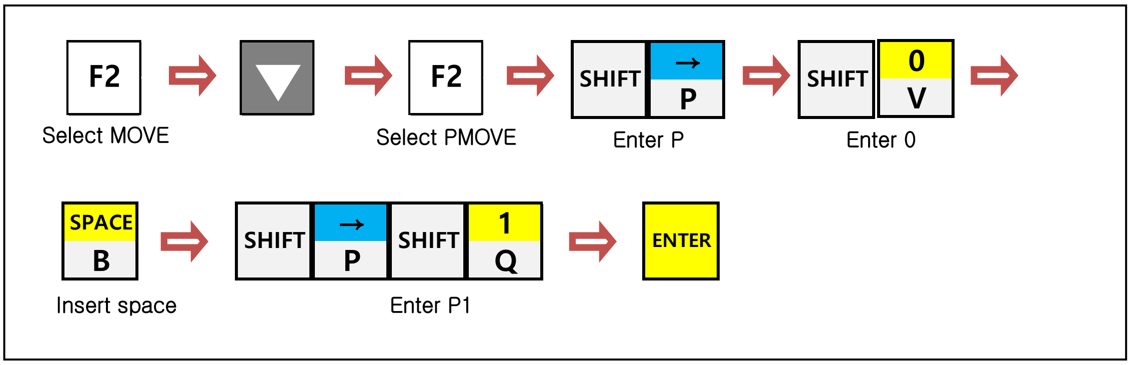
<RGA80A : EDIT>
JOB NAME = GOLD

PROG POINT PLC

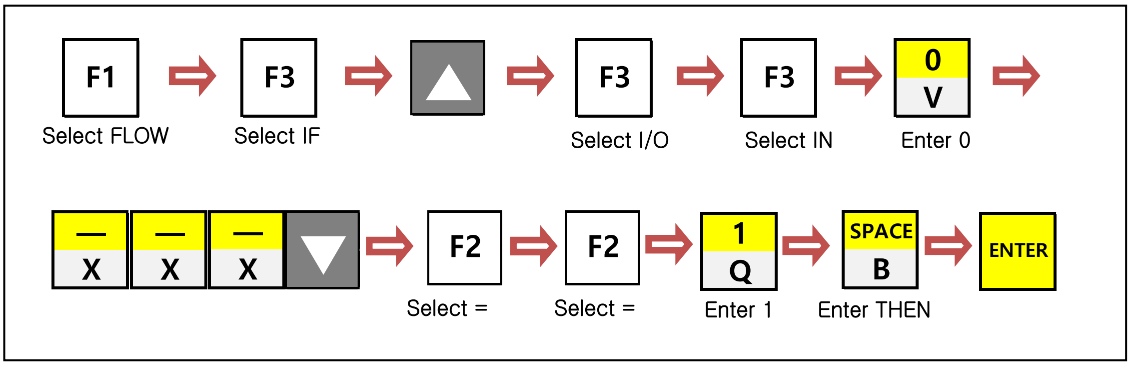
Update completed

3.5 How to enter complicated commands

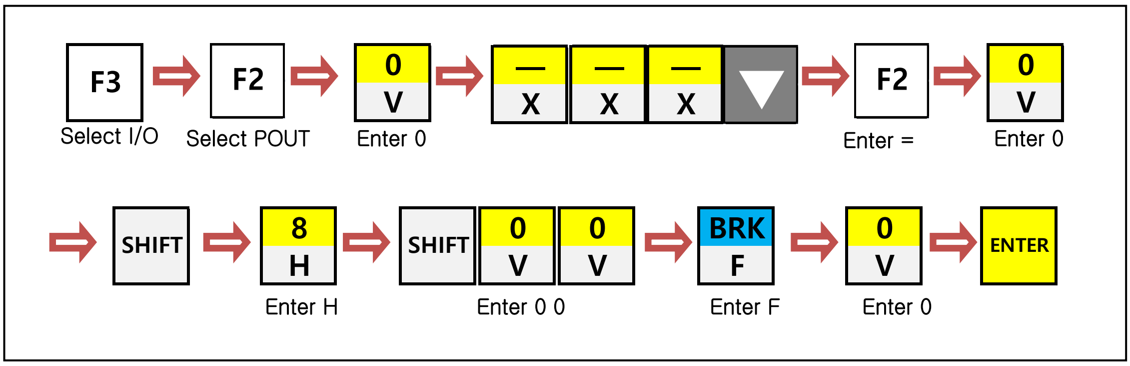
(1) PMOV command   E.g.) PMOV P0 P1



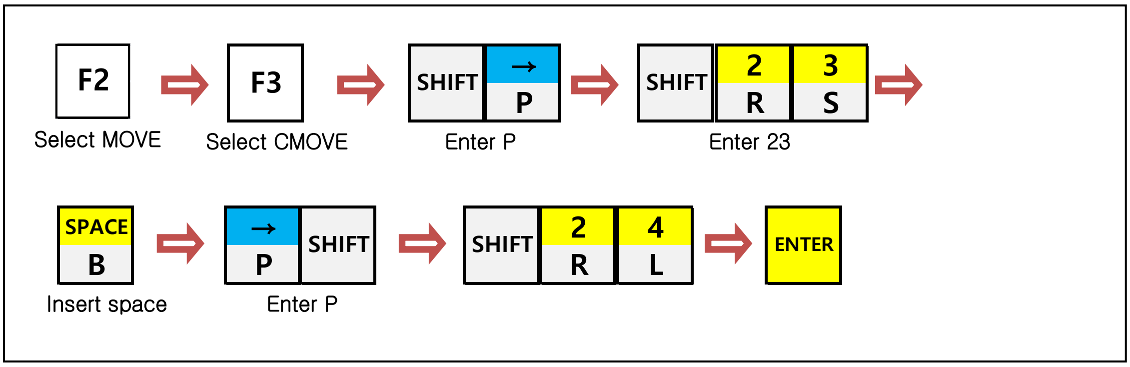
(2) IF command   E.g.) IF IN0==1 THEN



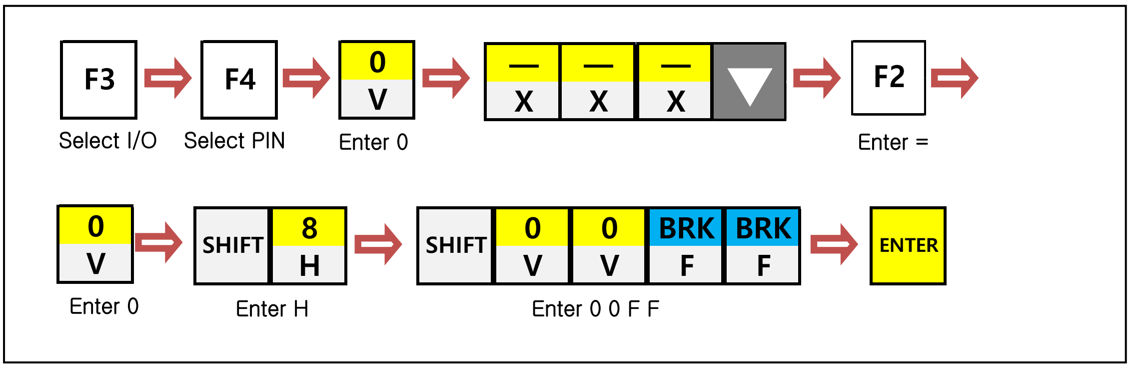
(3) POUT command   E.g.) POUT0=0H00F0



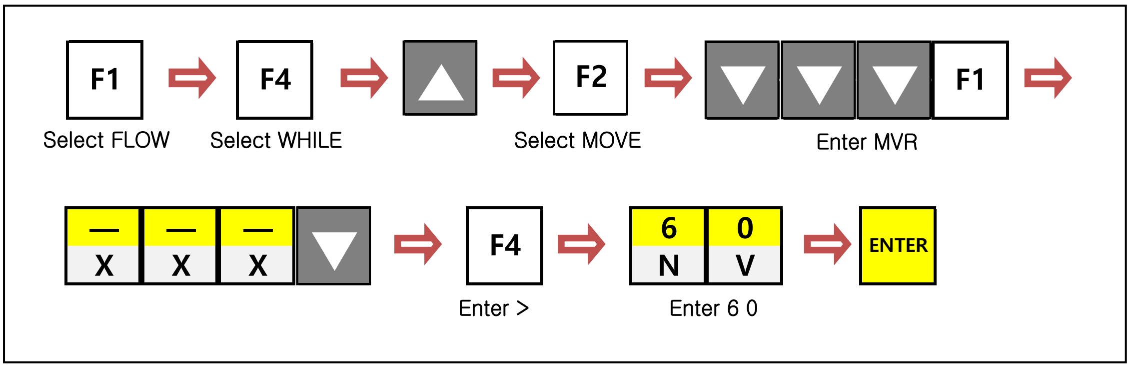
(4) CMOV command E.g.) CMOV P23 P24



(5) PIN command E.g.) PIN0=0H00FF



(6) WHILE command E.g.) WHILE MVR>60



3.6 Global Position Parameter Editing

Similar to conventional point editing (MDI, CURR).

Step 1. **Select GLOBAL POINT**

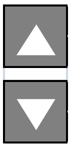
```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                  SELECT #
```

8
H

On MAIN MENU screen, select "8.GPNT"

Step 2. **Select channel to be used**

```
<GLBPNT MAIN>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST



SEL   INFO              EXIT
```



Select a robot by using "robot select " ▲▼ key.

Step 3. **Select POINT**

```
<GLBPNT MAIN>
 NO             TYPE
*CH1            XYZW
 CH2            XY_TEST



SEL   INFO              EXIT
```

F1

Select a robot by marking * and then "F1" key.

Step 4-1. **MDI Mode**

```
<GLBPNT EDIT>   V:      50
GLBPNT  GP0  NW  B L
A: 0            B : 0
Z: 0            W: 0



MDI    CURR             QUIT
```

F1

For MDI Mode

Step 4-2. **CURR Mode**

```
<GLBPNT EDIT>    V:          50
GLBPNT   GP0   NW   B L
A: 0                 B : 0
Z: 0                 W: 0


MDI     CURR                QUIT
```

F2

For CURR Mode

▶ The following sequences of above table are identical to the conventional point editing. (Teaching, Update, Copy, Delete, etc)

3.7 Global Inter & Floating point number Parameter

Step 1. **Select INT / FLT**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                 SELECT #
```

9
I

On MAIN MENU screen, select “9.INT/FLT”

Step 2. **INT EDIT MODE**

```
<INT EDIT>
I0      0
I1      0
I2      0
I3      0
I4      0
I5      0
BACK  NEXT  JUMP  FLOAT
```

F4

Global Integer Parameter editing screen

Step 3. **FLOAT EDIT MODE**

```
<FLOAT EDIT>
F0      0
F1      0
F2      0
F3      0
F4      0
F5      0
BACK  NEXT  JUMP  INT
```

F4

Global Float Parameter editing screen

■ Comment

▷ Setup of the global integer & floating point number parameters

DATA TYPE	Description
INT	Global integer parameter (0≤number≤499).
FLOAT	Global floating point number parameter (0≤number≤499).

CAUTION

▶ For the integer type parameter, -32766 to 32767 is supported.

▶ For the floating point number type parameter, up to the third place of decimal is supported, and the valid data place is totally 7. E.g. 1234.456: 4 places of mantissa and 3 places of decimal

Chapter 4 Job Run Mode

4.1 Overview

■ This mode executes JOB file (Robot JOB) written in JOB Mode.

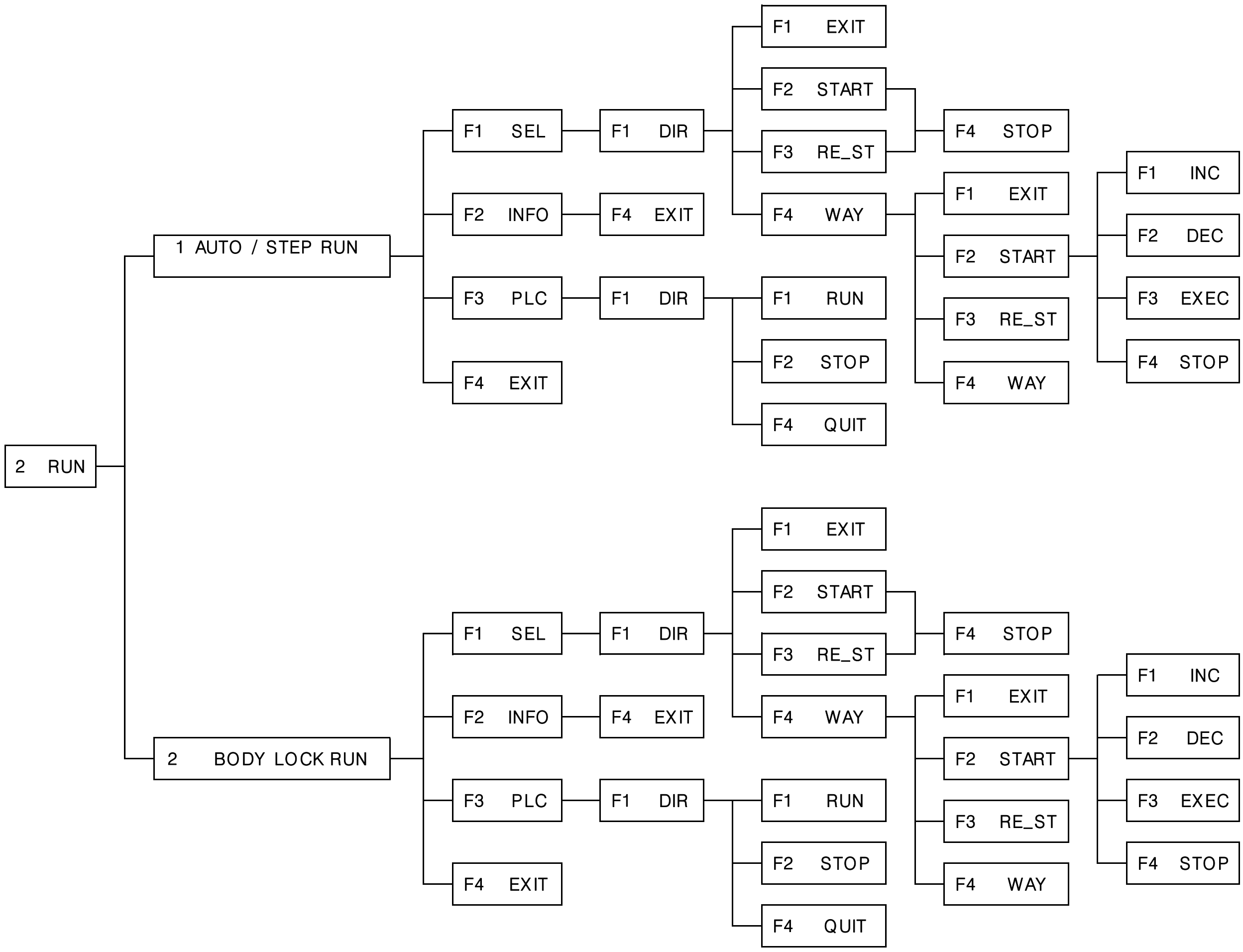
■ Before executing JOB file, be sure to perform Origin searching.

■ How to select JOB file to be executed includes the selection in a menu(DIR) and the selection by directly entering the file name.

■ Job Run Mode includes AUTO/STEP RUN and BODY LOCK RUN.

<table>
<tr><th colspan="2">Types</th><th>Function</th><th>Remarks</th></tr>
<tr><td rowspan="2">Internal Auto Run Mode</td><td>AUTO/STEP RUN</td><td>STEP : Program is executed by 1 step (1 line) and then stopped.
AUTO : All steps of Program are continuously executed.</td><td></td></tr>
<tr><td>BODY LOCK RUN</td><td>Identical to the current AUTO / STEP RUN MODE</td><td></td></tr>
<tr><td colspan="2">External Auto Run Mode</td><td>A mode for which Robot Program is executed by the external machines, such as a sequence.</td><td>Auto run mode for normal machines</td></tr>
</table>

4.2 Job Run Mode Flowchart



4.3 How to Manipulate

■ Write JOB intial and then execute STEP RUN Mode. Then check the abnormlity of Program and execute AUTO RUN.

▶ Screen

```
  <RUN MODE>
FILES F:10
 0:AA            1:BB
 2:CC            3:DD
 4:EE            5:FF
```

Number of files

File name

1) From DIR(Directory), select Program (E.g.: In the case of 10 registered Job Programs)

Step 1. **Initial main screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                  SELECT #
```

Open intial menu screen

Step 2. **Move to RUN screen**

```
<MAIN MENU>
1. JOB          2. RUN
3. HOST         4. PARA
5. ORIGIN       6. I/O
7. SYSTEM       8. GPNT
9. INT/FLT      A. ALARM

                  SELECT #
```

Select 2.RUN

2
R

Step 3. **Select RUN MODE**

```
<RUN>
  SELECT RUN MODE
 1. AUTO / STEP RUN
 2. BODY LOCK RUN



                INPUT NO.
```

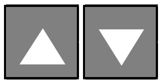
Select 1.AUTO / STEP RUN

1
Q

Step 4. **Select robot to be run**

```
<RUN MODE>
NO       TYPE               STE
 *1      XYZW           IDL *
 2       XY_ _          IDL


SEL   INFO    PLC    EXIT
```

Select '*' for a robot type to be run



```
<RUN MODE>
NO       TYPE               STE
 *1      XYZW           IDL *
 2       XY_ _          IDL


SEL   INFO    PLC    EXIT
```

Select SEL

F1

Step 5. **Select JOB to be run**

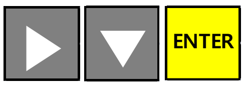
```
          <RUN MODE>

       INPUT FILE NAME
            : AA


DIR
```

Select DIR

F1

```
<RUN MODE>
0 AA              1 BB
2 CC              3 DD
4 EE              5 FF
6 GG              7 HH
8 II              9 TEST1
```

Using direction keys, select "TEST1"



Step 6. **RUN screen**

```
<RUN MODE : INS>
CH1: XYZW          A RDY
F: TEST1    S: 0   V: 300
0000:    MAIN
0001:    POS A
0002:    INT PP
0003:    A=P0
EXIT     START     RE-ST     WAY
```

RUN screen

CAUTION

■ **Be cautious before executing Program**

Prepare for any accidentr when executing Job after the Job Program is loaded.

1. A worker must move outside the robot working area.
2. Be ready to press EMERGENCY switch.
3. Check once again the job point and moving path of the robot.
4. After checking the job run at a low speed, graduall increase the speed.

4.3.1 Execution of AUTO/STEP RUN

4.3.1.1 Execution of STEP RUN

Step 1. **RUN Mode**

```
<RUN MODE : INS>
CH1: XYZW        S RDY
F: TEST1   S: 0  V: 300
0000:    MAIN
0001:    POS A
0002:    INT PP
0003:    A=P0
EXIT     START   RE_ST    WAY
```

On initial execution screen, check STEP MODE

Step 2. **Execute RUN**

```
<RUN MODE : INS>
CH1: XYZW        S RDY
F: TEST1   S: 0  V: 300
0000:    MAIN
0001:    POS A
0002:    INT PP
0003:    A=P0
EXIT     START   RE-ST    WAY
```

Select START / RESTART (Job starts to run)

F2 or **F3**

Step 3. **Execute STEP**

```
<RUN MODE : INS>
CH1: XYZW        S STP
F: TEST1   S: 0  V: 300
0000:    MAIN
0001:    POS A
0002:    INT PP
0003:    A=P0
INC      DEC     EXEC     STOP
```

EXEC (step-by-step execution)

F3

```
<RUN MODE : INS>
CH1: XYZW        S STP
F: TEST1   S: 1  V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
INC      DEC     EXEC     STOP
```

To continue execution, press F3 key.

F3

CAUTION

▶ When RE-ST is executed after the robot stops, the previous velocity is remained.

■ Comment

■ Explanation for screen

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:1     V:300
0001 : POS A
0002 : INT PP
0003 : A=P0
0004 : JMOV A
INC   DEC   EXEC   STOP
```

▶ S RDY : Now STEP RUN is being performed

▶ F: TEST1 → Job file name

▶ S: 1 → Number of step to be executed.

▶ V: 300 → Job execution velocity

★ If there is no velocity setup in Job Program:
→ 30% of maximum velocity (30% of Parameter Mv set-up value)

★ Using ▲ ▼ keys, the velocity can be adjusted. (Velocity adjustment while in Job run is possible.)

▶ Position of Job Program executed when 0004:JMOV A→ **F3** key is pressed.

▶ INC:=If Increment key is pressed, JobProgram increases 1 step
(Program is not executed, but only the step is increased.)

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:1     V:300
0001 : POS A
0002 : INT PP
0003 : A=P0
0004 : JMOV A
INC   DEC   EXEC   STOP
```

F1 ⇨

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:2     V:300
0002 : INT PP
0003 : A=P0
0004 : JMOV A
0005 : JMOV P1
INC   DEC   EXEC   STOP
```

F1 ⇨

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:3     V:300
0003 : A=P0
0004 : JMOV A
0005 : JMOV P1
0006 : JMOV P2
INC   DEC   EXEC   STOP
```

▶ DEC:= If Decrement key is pressed, JobProgram decreases 1 step
(Program is not executed, but only the step is decreased.)

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:3     V:300
0003 : A=P0
0004 : JMOV A
0005 : JMOV P1
0006 : JMOV P2
INC   DEC   EXEC   STOP
```

F2 ⇨

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:2     V:300
0002 : INT PP
0003 : A=P0
0004 : JMOV A
0005 : JMOV P1
INC   DEC   EXEC   STOP
```

F2 ⇨

```
<RUN MODE: INS>
CH1:XYZW        S STP
F:TEST1   S:1     V:300
0001 : POS A
0002 : INT PP
0003 : A=P0
0004 : JMOV A
INC   DEC   EXEC   STOP
```

▶ EXEC: 1 step (JMOV P1) of Job Program displayed on a teaching pendant screen is executed.

▶ STOP: STEP RUN is stopped.

■ Restart after Job Run stops

Step 1. **STEP RUN screen**

```
<RUN MODE : INS>
CH1: XYZW        S STP
F: TEST1   S: 1  V: 300
0001:   POS A
0002:   INT PP
0003:   A=P0
0004:   JMOV A
INC     DEC      EXEC     STOP
```

On STEP RUN screen,

Step 2. **Stop STEP RUN**

```
<RUN MODE : INS>
CH1: XYZW        S RDY
F: TEST1   S: 1  V: 300
0001:   POS A
0002:   INT PP
0003:   A=P0
0004:   JMOV A
INC     DEC      EXEC     STOP
```

F4

To stop STEP RUN

Step 3. **Restart STEP RUN**

```
<RUN MODE : INS>
CH1: XYZW        S RDY
F: TEST1   S: 1  V: 300
0001:   POS A
0002:   INT PP
0003:   A=P0
0004:   JMOV A
EXIT    START    RE-ST    WAY
```

F3

Restart after stop (start with No. 3 step)

```
<RUN MODE : INS>
CH1: XYZW        S RDY
F: TEST1   S: 1  V: 300
0001:   POS A
0002:   INT PP
0003:   A=P0
0004:   JMOV A
INC     DEC      EXEC     STOP
```

Restart screen

■ Change STEP RUN → AUTO RUN

Step 1. **STEP RUN screen**

```
<RUN MODE : INS>
CH1: XYZW           S STP
F: TEST1    S: 1    V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
INC      DEC        EXEC      STOP
```

On STEP RUN screen,

Step 2. **Stop STEP RUN**

```
<RUN MODE : INS>
CH1: XYZW           S RDY
F: TEST1    S: 1    V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
INC      DEC        EXEC      STOP
```

F4

Stop STEP RUN

Step 3. **Change STEP → AUTO**

```
<RUN MODE : INS>
CH1: XYZW           S RDY
F: TEST1    S: 1    V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
EXIT     START      RE_ST     WAY
```

F4

Change STEP RUN → AUTO RUN

```
<RUN MODE : INS>
CH1: XYZW           A RDY
F: TEST1    S: 1    V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
EXIT     START      RE_ST     WAY
```

Changed to AUTO RUN (S → A)

4.3.1.2 AUTO RUN execution

Step 1. **AUTO RUN screen**

```
<RUN MODE : INS>
CH1: XYZW          A RDY
F: TEST1    S: 1   V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
EXIT     START     RE_ST     WAY
```

F2

On execution start screen에서, select START

Step 2. **AUTO RUN is being executed**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
                             STOP
```

Start Job Run

■ Job Run after stop

Step 1. **Stop AUTO RUN**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1    S: 3   V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
                             STOP
```

F4

Stop Job Run

Step 2. **RESTART**

```
<RUN MODE : INS>
CH1: XYZW          A RDY
F: TEST1    S: 3   V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
EXIT     START     RE-ST     WAY
```

F3

Restart after stop (start with No. 3 step)

4.3.2 BODY LOCK RUN execution

Step 1. **BODY LOCK RUN screen**

```
<RUN MODE : INS>
CH1: XYZW          A RDY
F: TEST1    S: 1   V: 300
0001:     POS A
0002:     INT PP
0003:     A=P0
0004:     JMOV A
EXIT      START    RE_ST    WAY
```

F2

On run start screen, select START

Step 2. **BODY LOCK RUN is being executed**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
0001:     POS A
0002:     INT PP
0003:     A=P0
0004:     JMOV A
                            STOP
```

Job run start

★ Currently, BODY LOCK RUN is executed identical to AUTO/STEP RUN, but a motor does not operate.

■ Comment

■ Screen Explanation 1)

<AUTO RUN>

```
<RUN MODE: INS>
CH1:XYZW        A RDY
F:TEST1   S:1   V: 300
0001 : POS A
0002 : INT PP
0003 : A=P0
0004 : JMOV A
EXIT START RE-ST   WAY
```

F4

<STEP RUN>

```
<RUN MODE: INS>
CH1:XYZW        S RDY
F:TEST1   S:1   V: 300
0001 : POS A
0002 : INT PP
0003 : A=P0
0004 : JMOV A
EXIT START RE-ST   WAY
```

▶ A RDY : Now AUTO RUN is being operated

▶ F: TEST1 → Job file name

▶ S: 1 → Number of step to be executed

▶ V: 300 → Job execution velocity

★If there is no velocity setup in Job Program:
→ 30% of maximum velocity (30% of Parameter Mv set-up value)

★ Using ▲ ▼ keys, the velocity can be adjusted. (Velocity adjustment while in Job run is possible.)

▶ 0001 :POS A→ Command of the step to be executed

▶ START : Auto Run starts from No. 0 step of Job Program

▶ RE-ST : Restart; Auto Run starts from the stopped step.

▶ WAY : AUTO RUN → Changes STEP RUN

4.3.3 Monitoring while RUN is being executed

■ **While in Job Run (AUTO/STEP RUN), the followings can be monitored.**

- I/O signal ---- Only user I/O can be monitored.
- Coordinate value of each axis
- Velocity (rpm) of each axis
- Torque (%) of each axis

(However, in BODY LOCK RUN, I/O monitoring is available, but POS,TRQ, and RPM monitoring is not available.)

■ **I/O signal checking**

Checking sequence is displayed in the order of OUT → IN → PLC → OUT.

Step 1. **AUTO RUN is being executed**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
0001:    POS A
0002:    INT PP
0003:    A=P0
0004:    JMOV A
                         STOP
```

While in RUN status,

Step 2. **User output Signal**

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
INS:     JMOV P2
OUT      0123456789ABCDEF
000      0000000000000000

                         STOP
```

I/O
A

Check user output signal (0 to 15)

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
INS:     JMOV P2
OUT      0123456789ABCDEF
016      0000000000000000

                         STOP
```

Pg Up

Check user output signal (16 to 31)

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
INS:     JMOV P2
OUT      0123456789ABCDEF
032      0000000000000000

                         STOP
```

Pg Dn

Check user output signal (32 to 47)

Step 3. **User input signal**

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
IN         0123456789ABCDEF
000        0000000000000000
                        STOP
```

Check user input signal (0 to 15)

I/O A

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
IN         0123456789ABCDEF
016        0000000000000000
                        STOP
```

Check user input signal (16 to 31)

Pg Up

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
IN         0123456789ABCDEF
032        0000000000000000
                        STOP
```

Check user input signal (32 to 47)

Pg Dn

Step 4. **PLC signal**

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
PLC        0123456789ABCDEF
096        0000000000000000
                        STOP
```

Check PLC signal (96 to 111)

I/O A

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
PLC        0123456789ABCDEF
112        0000000000000000
                        STOP
```

Check user input signal (112 to 127)

Pg Up

```
<RUN MODE : IO>
CH1: XYZW          A RUN
F: TEST1    S: 1    V: 300
INS:       JMOV P2
PLC        0123456789ABCDEF
128        0000000000000000
                        STOP
```

Check user input signal (128 to 143)

Pg Dn

■ Checking coordinate value of each axis

▶ Auto Run state → Monitor the coordinate value of each axis

Step 1. **POSITION**

```
<RUN MODE : POS>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:      JMOV P2
X: -0.46           Y : 48.78
Z: 286.26          W: 360

                          STOP
```

Pg Dn

Current coordinate value of each axis is displayed

■ Checking velocity (rpm) / torque (TRQ) / max TRQ of each axis

▶ Auto Run state → Monitor the velocity value of each axis

Step 1. **RPM**

```
<RUN MODE : RPM>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:      JMOV P2
X: 1500            Y : 2500
Z: 3000            W: 2700

                          STOP
```

VEL C

Velocity value (unit: RPM) of each axis is displayed

Step 2. **TRQ**

```
<RUN MODE : TRQ>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:      JMOV P2
X: 15              Y : 25
Z: 30              W: 27

                          STOP
```

VEL C

Torque (unit: %) of each axis is displayed

Step 3. **MAX TRQ**

```
<RUN MODE : MAX TRQ>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:      JMOV P2
X: 50              Y : 80
Z: 120             W: 95

                          STOP
```

VEL C

MAX TRQ (unit: RPM) of each axis is displayed

Step 4. **LOAD**

```
<RUN MODE : LOAD>
CH1: XYZW          A RUN
F: TEST1    S: 1   V: 300
INS:      JMOV P2
X: 17              Y : 15
Z: 30              W: 10

                         STOP
```

VEL C

LOAD of each axis is displayed

▶ Auto Run state → Monitor the items

Group	RPM	TRQ	MAX TRQ	LOAD
Selection key	VEL C	VEL C	VEL C	VEL C
Item moving Pg Up Pg Dn	RPM			LOAD
	LINEAR MOTION			SIGLE TURN
	JOINT MOTION			MULTI TURN
				VDC
				POSITON ERROR
◀ ▶	MAX RPM			

- **RPM**: velocity value of each axis
- **LINEAR MOTION**: velocity value when in axis interpolation
- **JOINT MOTION:** velocity value when in point to point motion
- **MAX RPM:** maximum velocity value of each axis
- **TRQ:** torque value of each axis
- **MAX TRQ:** maximum torque value of each axis
- **LOAD:** load rate of each axis
- **SINGLE TURN:** single-turn position of the encoder of each axis
- **MULTI TURN:** multi-turn position of the absolute motor
- **VDC:** voltage value at DC link stage
- **POSITION ERROR:** Difference between the reference position and the real position of each axis (unit: pulse)

■ Integer/Float parameter monitoring

▶ Auto Run state → Monitor Integer/Float

Step 1. **INTEGER**

```
<RUN MODE : INTEGER>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:       JMOV P2
I000:      4
I001:      1
I002:      2
                      STOP
```

INT D

INTEGER parameter is displayed

Step 2. **FLOAT**

```
<RUN MODE : FLOAT>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:       JMOV P2
F000:       0.91
F001:       0
F002:      59.63
                      STOP
```

INT D

FLOAT parameter is displayed

■ Global Point monitoring

▶ Auto Run state → Monitor Global point

Step 1. **GLOBAL POINT**

```
<RUN MODE : GPNT#0>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
INS:       JMOV P2
X: 50              Y : 27.78
Z: 58.33           W: 56.33

                      STOP
```

BRK F

Monitor Global Point

▶ How to return from Monitoring screen → Auto Run screen

Step 1. **AUTO RUN**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
0001:      POS A
0002:      INT PP
0003:      A=P0
0004:      JMOV A
                      STOP
```

ESC

Return to AUTO RUN screen

4.3.4 **Skip during Run**

■ **When, during Auto Run, JobProgram is not performed because of input-conditioned Wait command (e.g., IN1=1), the Wait command can be skipped.**

■ **How to manipulate**

Step 1. **Signal waiting state**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
0001:     IN1=1
0002:     JMOV P1
0003:     A=P0
0004:     JMOV A
                          STOP
```

In the waiting state, until IN1 signal is turned on,

Step 2. **Neglect signal**

```
<RUN MODE : INS>
CH1: XYZW          A RUN
F: TEST1   S: 1    V: 300
0002:     JMOV P1
0003:     A=P0
0004:     JMOV A
0005:     JMOV P2
                          STOP
```

ESC

To skip by neglecting the signal

CAUTION

▶ When during Job run, a robot is in the waiting state due to Wait command, be careful of handling the teaching pendant to stop the robot.

▶ Be careful of confusing Stop key (F4) and SKIP key (ESC) because they are close to each other. If ESC key is pressed mistakenly, the robot may execute the next step which may cause the unexpected accidents.

Chapter 5 I/O Mode

☞ Sufficient PLC I/O ports are to be secured by well estimating I/O ports required for the robot.
☞ System I/O ports are to be secured, since most of them are reuired.
☞ The required number of User I/O is determined according to the robot Job process.
☞ In Chapter 5, the input-output ports and their monitoring method are explained.

5.1 Overview

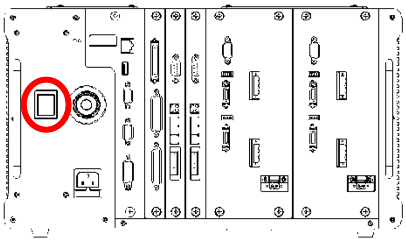
■ I/O mode is for displaying the input and output port status on a screen.
- The output port can be turned on/off by manipulating keys of the teaching pendant.
- The status of 32 points of User I/O and 32 points of Option I/O) is displayed.

■ Operation of peripherals(PLC, sensor, solenoid, etc.) can be checked, when connected to the peripheral.

5.2 How to manipulate

■ How to check the input/output signal when the robot is in stationery state, not in Run mode.
(*For checking the input/output signal when the robot is in Run Mode, refer to "Monitoring while in run".)

Step 1. **Apply the power**



Apply the power
(If the error occurs, settle it according to the error code.)

Step 2. **EXIT System Mode**

SYSTEM MODE
Auto Run Mode

1R 2R EXIT

F4

Exit the system mode

Step 3. **Move to MAIN MENU**

<MAIN MENU>

1. JOB	2. RUN
3. HOST	4. PARA
5. ORIGIN	6. I/O
7. SYSTEM	8. GPNT
9. INT/FLT	A. ALARM

SELECT #

6
N

Select I/O

Step 4. **Move to I/O Mode**

<I/O MAIN>

1. DIO	2. PLC
3. FBUS	

Input #

Move in I/O Mode

5.2.1 Digital IO

5.2.1.1 Digital IO OUTPUT Monitoring

The number of DIO Ports depends on the setup of Digital Board.

▶ **Explanation for User output screen**

```
<DIO(OUT)>
OUT   0123456789ABCDEF
PO1  :1111111111111111
PO1  :0000000000000000
PO2  :0000000000000000
PO3  :0000000000000000
PO4  :0000000000000000
OUT   IN   NEW
```

OUT : Output screen status

0 : Signal OFF

1 : Signal ON

0 to F : Output port number

PO# : User output port number (0 to 19)

SS# : System output port number (0 to 1),

For information on the assigned pins, refer to SYSIO Input Parameter Menu.

▶ **System output port screen**

```
<DIO(OUT)>
OUT   0123456789ABCDEF
PO2  :0000000000000000
PO3  :0000000000000000
PO4  :0000000000000000
SS0  :1100001110001110
SS1  :0000
OUT   IN   NEW
```

▶ **How to turn on User output port signal**

E.g.) In the case that OUT6 is turned ON(=1)/OFF(=0) :

```
OUT  0123456789ABCDEF
PO0: 0000000000000000
```

6
N

```
OUT  0123456789ABCDEF
PO0: 0000001000000000
```

E.g.) In the case that OUT20 is turned ON(=1)/OFF(=0):

First, move the cursor (:), USR : → OP0 : by using PgDn key.

```
OUT  0123456789ABCDEF
PO0  0000000000000000
PO1: 0000000000000000
```

5
M

```
OUT 0123456789ABCDEF
PO0 0000000000000000
PO1:0000010000000000
```

5.2.1.2 Digital IO INPUT Monitoring

▶ **User input screen**

```
<DIO(IN)>
I N   0 1 2 3 4 5 6 7 8 9 A B C D E F
P I 0 : 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
P I 1 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P I 2 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P I 3 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P I 4 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
OUT   IN   NEW
```

▶ **System input port screen**

```
<DIO(IN)>
I N   0 1 2 3 4 5 6 7 8 9 A B C D E F
P I 2 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P I 3 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P I 4 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
S S 0 : 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 0
S S 1 : 0 0 0 0
OUT   IN   NEW
```

IN : DIO input screen status

0 : Signal OFF

1 : Signal ON

0 to F : Input port number

PI# : User input port number (0 to 19)

SS# : System input port number (0 to 1),

For information on the assigned pins, refer to SYSIO Input Parameter Menu.

5.2.2 PLC Monitoring

1. Monitoring the internal ports of PLC
2. Ports are assigned to 96 to 255.
3. Input monitoring and Output is available. (Identical to DIO output.)

▶ **Explanation for User output screen**

```
<PLC>
PLC   0 1 2 3 4 5 6 7 8 9 A B C D E F
096 : 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
112 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
128 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
144 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
160 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

PLC : PLC internal port input/output screen
0 : Signal OFF
1 : Signal ON
0 to F : Output port number
Assigned internal port number (96 to 255)

5.2.3 LATCH

1. Monitoring input contact of servo module
2. Consists of 2 input contacts per servo module

Step 1. **LATCH Movement**

```
<I/O MAIN>
1. DIO            2. PLC
3. FBUS           4. LATCH

                  Input  #
```

4
L

Select LATCH.

▶ Description of Latch Input Screen

```
<LATCH(IN)>
IN   0123456789ABCDEF
SV0 00
SV1 00
SV2 00

                    ESC
```

SV0: 1st servo module

SV1: 2nd servo module

SV2: 3rd servo module

0 : Signal OFF

1 : Signal ON

Chapter 6 Simple PLC functions

> ☞ Most simple functions of PLC are built in the controller.
> ☞ If the controller power is shut down, all the PLC signals being output are reset.

6.1 Overview

- Basic PLC functions can be used in a robot controller.
- When powering on (or rebooting), the selected PLC JOB can be automatically run.
- Up to 10 PLC Programs can be written.
- In 1 PLC Program, up to 1000 steps can be written.
- Scanning time of PLC Program is 30ms.
- Writing Program can be done by the teaching pendant or the dedicated software (Unihost).

- T/P based Job Sequence

Parameter Setup

↓

Program Write

↓

Program Run

6.2 PLC Parameter setup

E.g.) Before PLC_9 JOB is operated in AUTO RUN, Parameter must be set up.
Parameter setting is as 1), 2), 3) below.

1) PLC JOB No. to be auto run is determined.

Step 1. **Move to PUBLIC PARAMETER**

```
<PUBLIC PARAMETER>
1: HW CONF        2: PALLET
3. PLC            4: ETC



                  group #
```

On MAIN MENU screen,
Move 4.PARA → PUB

Step 2. **Move to PLC**

```
<PUBLIC PARAMETER>
1: HW CONF        2: PALLET
3. PLC            4: ETC



                  group #
```

3 / S

Select PLC

Step 3. **Move to and Set up PNO**

```
<PUBLIC-PLC>
      GROUP : PLC
1: PNO            2: AUTO
3: IO ENB         4: BCD
5: OCL


                  item #
```

1 / Q

Select 1: PNO

```
<PLC-PNO>
      PLC JOB NUMBER

CHANNEL = 1
JOB NUM = 0
```

1 / Q or 3 / S

Set the robot channel to be used.

<PLC-PNO>
PLC JOB NUMBER

CHANNEL = 1
JOB NUM = 9

9
I

Enter PLC_9 JOB → 9

Step 4. **Set-up value updating**

<PLC-PNO>
PLC JOB NUMBER

CHANNEL = 1
JOB NUM = 9

Update OK? (ENT/ESC)

ESC

Select ESC

<PLC-PNO>
PLC JOB NUMBER

CHANNEL = 1
JOB NUM = 9

Update OK? (ENT/ESC)

ENTER

Select update

<PUBLIC-PLC>
GROUP : PLC

1: PNO	2: AUTO
3: IO ENB	4: BCD
5: OCL	

item #

Update completed

2) Auto run of PLC JOB for the selected number is determined.

Step 1. **Move to PLC screen**

```
<PUBLIC-PLC>
            GROUP : PLC
1: PNO                2: AUTO
3: IO ENB             4: BCD
5: OCL


                         item #
```

Open PLC screen

Step 2. **Move to and Set up AUTO**

```
<PUBLIC-PLC>
            GROUP : PLC
1: PNO                2: AUTO
3: IO ENB             4: BCD
5: OCL


                         item #
```

2: AUTO

2
R

```
<PLC-AUTO>
            AUTO RUN : YES
1: YES                2: NO



                         input #
```

1: YES

1
Q

Step 3. **Set-up value updating**

```
<PLC-AUTO>
            AUTO RUN : YES
1: YES                2: NO



Update OK? (ENT/ESC)
```

Select ESC

ESC

```
<PLC-AUTO>
            AUTO RUN : YES
1: YES                2: NO



Update OK? (ENT/ESC)
```

Select update

ENTER

```
<PUBLIC-PLC>
       GROUP : PLC
1: PNO            2: AUTO
3: IO ENB         4: BCD
5: OCL

                  item #
```

Update completed

3) Determine to use I/O in PLC.

Step 1. **Move to PLC screen**

```
<PUBLIC-PLC>
       GROUP : PLC
1: PNO            2: AUTO
3: IO ENB         4: BCD
5: OCL

                  item #
```

Open PLC screen

Step 2. **Move to & Set up IO ENB**

```
<PUBLIC-PLC>
       GROUP : PLC
1: PNO            2: AUTO
3: IO ENB         4: BCD
5: OCL

                  item #
```

3
S

Select 3: IO ENB

```
<PLC-ENB>
       PLC I/O ENABLE
OP1: NO           OP2: -
OP3: -            OP4: -
OP5: -            OP6: -
```

ENTER

Select ENTER

```
<PLC-ENB>
       PLC I/O ENABLE
OP1: YES          OP2: -
OP3: -            OP4: -
OP5: -            OP6: -
```

Change OP1: NO → YES

Step 3. **Set-up value updating**

```
<PLC-ENB>
           PLC I/O ENABLE
OP1: YES              OP2: -
OP3: -                OP4: -
OP5: -                OP6: -

Update OK? (ENT/ESC)
```

ESC

Select ESC

```
<PLC-ENB>
           PLC I/O ENABLE
OP1: YES              OP2: -
OP3: -                OP4: -
OP5: -                OP6: -

Update OK? (ENT/ESC)
```

ENTER

Select update

```
<PUBLIC-PLC>
           GROUP : PLC
1: PNO                2: AUTO
3: IO ENB             4: BCD
5: OCL

                         item #
```

Update completed

6.3 PLC JOB editing

Step 1. **Move to JOB EDIT screen**

```
<XYZW : JOB EDIT>



DIR      JEDIT
```

On higher menu screen, select 1.JOB

Step 2. **Move to JOB EDIT**

```
<XYZW : JOB EDIT>



DIR      JEDIT
```

Select JEDIT

F2

Step 3. **Set up FILE NAME**

```
<XYZW : EDIT>
 FILE NAME = 0
```

Enter “0” as FILE NAME to be written (PLC_0)

0/V ENTER

Step 4. **Set up FILE NUMBER**

```
<XYZW : EDIT>
 FILE NUM = 0
```

Enter “0” as FILE Number to be written

0/V ENTER

Step 5 **Set up FILE NUMBER**

```
<XYZW : EDIT>
 FILE NAME= 0



PROG    POINT    PLC
```

F3 Select PLC

Step 6. **Enter PLC command**

```
<XYZW : EDIT>
STEP    INS    OP1   OP2
*0000   —
 0001   —
 0002   —
 0003   —
 0004   —
INS     EDIT     JUMP     QUIT
```

F1 Enter command

```
<XYZW : EDIT>
STEP    INS    OP1   OP2   i
*0000   —
 0001   —
 0002   —
 0003   —
 0004   —
FLOW    CTRL
```

Input screen

※ Command Screen display

Group Selection key / Move screen.	FLOW	CTRL
	F1	F2
▲ ▼	AND OR NOT START	TMR CTR MCS MCSC
	OUT PULS SET RESET	DATA BLK EOP

6.4 PLC JOB RUN

Step 1. **Initial main screen**

```
<MAIN MENU>
1. JOB              2. RUN
3. HOST             4. PARA
5. ORIGIN           6. I/O
7. SYSTEM           8. GPNT
9. INT/FLT          A. ALARM

                      SELECT #
```

Open intial menu screen

Step 2. **Move to RUN screen**

```
<MAIN MENU>
1. JOB              2. RUN
3. HOST             4. PARA
5. ORIGIN           6. I/O
7. SYSTEM           8. GPNT
9. INT/FLT          A. ALARM

                      SELECT #
```

2 / R

Select 2.RUN

Step 3. **Select RUN MODE**

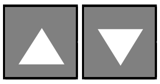
```
<RUN>
      SELECT RUN MODE
 1. AUTO/STEP RUN
 2. BODY LOCK RUN


                    INPUT NO.
```

1 / Q

Select 1.AUTO / STEP RUN

Step 4. **Select robot**

```
<RUN MODE>
NO      TYPE              STE
 *1     XYZW          IDL *
 2      XY_ _         IDL


SEL   INFO    PLC    EXIT
```



Select a robot type to be run by marking *.

Step 5. **Move to PLC and then select JOB**

```
<RUN MODE>
NO      TYPE          STE
 *1     XYZW        IDL *
 2      XY_ _       IDL


SEL  INFO  PLC  EXIT
```

F3

Select PLC

```
        <RUN MODE>

     INPUT FILE NAME
           : 0


DIR
```

F1

Select DIR

```
<RUN MODE>
 FILES   F : 4
 0 PLC_0        1 PLC_1
 2 PLC_2        3 PLC_3
```

■ Select PLC JOB number to be executed mark (PLC_1)

```
<RUN MODE>
 FILES   F : 4
0 PLC_0     1 PLC_1
2 PLC_2     3 PLC_3
```

ENTER

① Select PLC_1 JOB

Step 6. **RUN PLC**

```
      PLC RUN MODE
   No Job is Running



RUN    STOP          QUIT
```

F1

② Select RUN .

PLC RUN MODE
No Job is Running

Complie OK !

Start? (ENT/ESC)

ENTER

Select ENTER

PLC RUN MODE
PLC_1 is Running

RUN STOP QUIT

③ During running

■ Comment

① If there is no PLC Job being currently run, "No Job is Running" is displayed.

* If there is a PLC Job being currently run, the name of the PLC JOB is displayed.

PLC RUN MODE
PLC_1 is Running

RUN STOP QUIT

When PLC_1 is already being run.

* To execute the selected JOB, press F1 (RUN).
* If F1 (RUN) key is pressed, currently running PLC JOB is stopped, and the selected PLC JOB is run.
* To stop the running PLC JOB, press F2 (STOP) key.
* To escape from PLC RUN MODE, press F4 (QUIT) key.

② On the previous screen, if F1 (RUN) key is pressed, PLC JOB is compiled.

* If, due to miss-writing Program, the compiling cannot be performed, the following screen appears.
* Step number of the erroneously written program is displayed aside #.

PLC RUN MODE
No Job is Running

Compile Fail : #4

* Press ENTER key to return to the previous screen.
* If ESC key is pressed, PLC JOB is not executed and the previous screen is recovered.

③ To stop the running JOB, press F2 (STOP) key.

Important

▶ Comiling is required: Although PLC Program has been change, if it is not compiled in RUN screen, the JOB before change is run.

6.5 PLC command

■ PLC command consists basically of commands and input-output ports.

■ Input-output port is identified by the suffix of port number, such is as follows.
X : External port input, Y : External port output, M : Internal port input-output, S : system port input

■ In the external port input-output, the relation between the port number and the real I/O is as follows.

Port number	Real I/O
0 to 15	User I/O
16 to 31	Option 0 I/O
32 to 47	Option 1 I/O
48 to 63	Option 2 I/O
64 to 79	Option 3 I/O

■ Internal ports are all 470 points, and Internal port number can have values between 320 and 799.
* For example,
1) START X0 : Reads External port No. 0, i.e., PI0 input port value.
2) AND X32 : Reads External port No. 32, i.e., PI2 Input port value.
3) AND X65 : Reads External port No. 65, i.e., PI4 Input port value.
4) OR Y0 : Reads External port No. 0, i.e., PO0 output port value.
5) AND M91 : Reads Internal port No. 91 value.
6) OUT Y0 : Write External port No. 0, i.e., PO0 output port.
7) OUT M82 : Write Internal port No. 82 value.

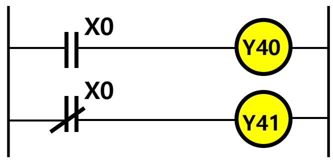
6.5.1 PLC command list

Command	Function	Format
START	Start the logic operation (Aport)	START X(or Y,M)n
STARTNOT	Start the logic operation (Bport)	STARTNOT X(or Y,M)n
OUT	Output the operation result to the designated port.	OUT Y(or M)n
AND	Serial connection of the logic operation (Aport)	AND X(or Y,M)n
ANDNOT	Serial connection of the logic operation (Bport)	ANDNOT X(or Y,M)n
ANDBLK	ANDs two Blocks.	ANDBLK X(or Y,M)n
OR	Parallel connection of the logic operation (Aport)	OR X(or Y,M)n
ORNOT	Parallel connection of the logic operation (Bport)	ORNOT X(or Y,M)n
ORBLK	ORs two Blocks.	ORBLK X(or Y,M)n
PULS	Output for 1 Scan Time, from when the input port value is changed from Off to On (1 Scan Time = 30ms)	PULS Y(or M)n
PULSNOT	Output for 1 Scan Time, from when the input port value is changed from On to Off (1 Scan Time = 30ms)	PULSNOT Y(or M)n
SET	If the input is On, the designated output port is remained to On state.	SET Y(or M)n
RESET	If the input is On, the designated output port is remained to Off state.	RESET Y(or M)n
TMR	Decreases according to input condition, and outputs the data when the current value is 0. (Unit of TMR is 0.1s)	See Exercise
CTR	Decreases according to Enable condition and Count Pulse, and outputs the data when the current value is 0. If Enable is turned Off, the current value is reset to the prescribed value. Input after the count is completed is neglected.	See Exercise
MCS	Master Control Set (Common Interlock Set)	See Exercise
MCSC	Master Control Reset (Common Interlock Reset)	See Exercise
DATA	Data is designated to CTR and TMR	DATA Integer (Integer: 1 to 255)

6.6 Exercise of PLC Program Writing

6.6.1 START / STARTNOT

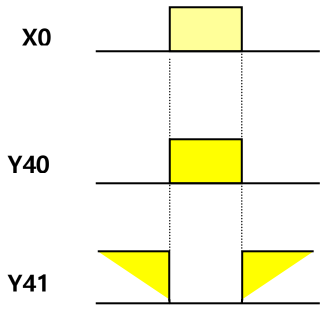
▶ **Sequence Diagram**



▶ **Program**

```
0   START X0
1   OUT Y40
2   STARTNOT X0
3   OUT Y41
4   EOP
```

▶ **Time Chart**



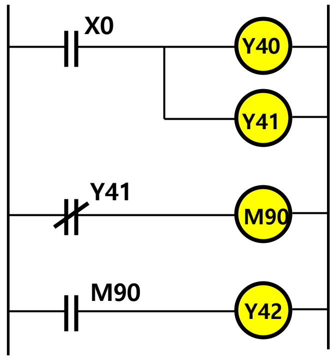
⚠ 주의

*** CAUTION**
STARTNOT (O)
START NOT (X)

6.6.2 OUT

OUT

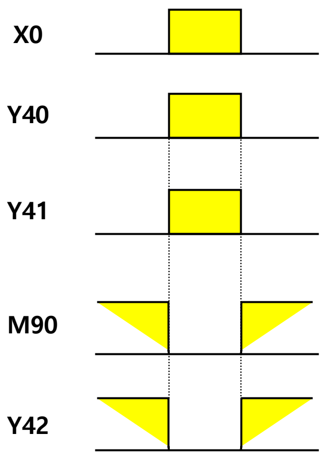
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 OUT Y40
2 OUT Y41
3 STARTNOT Y41
4 OUT M90
5 START M90
6 OUT Y42
7 EOP

▶ **Time Chart**

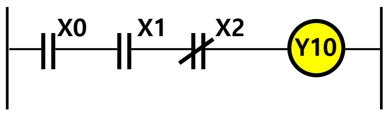


주의

*** CAUTION**
For checking I/O being used in Program, refer to Chapter 5 I/O Mode. – Corresponds to PLC port monitoring.

6.6.3 AND / ANDNOT

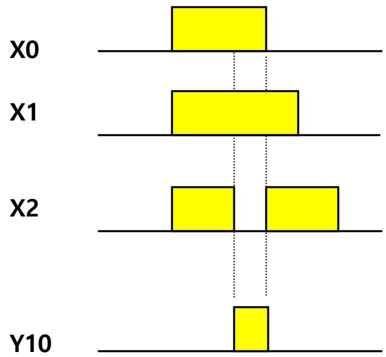
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 AND X1
2 ANDNOT X2
3 OUT Y10
4 EOP

▶ **Time Chart**

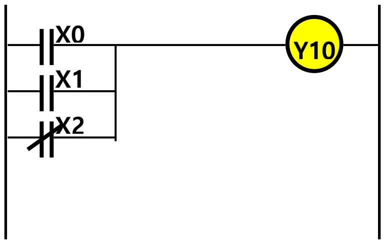


주의

ANDNOT (O)
AND NOT (X)

6.6.4 OR / ORNOT

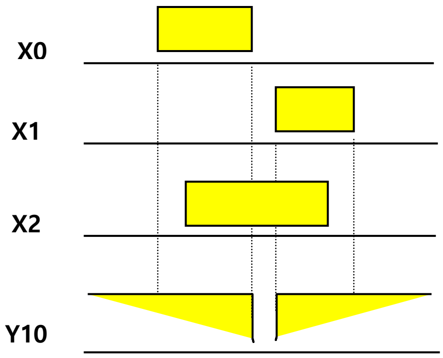
▶ **Sequence Diagram**



▶ **Program**

```
0   START X0
1   OR X1
2   ORNOT X2
3   OUT Y10
4   EOP
```

▶ **Time Chart**



주의

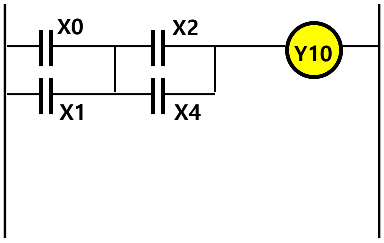
ORNOT (O)

OR NOT (X)

6.6.5 **ANDBLK (= ANDBLOCK)**

ANDBLOCK

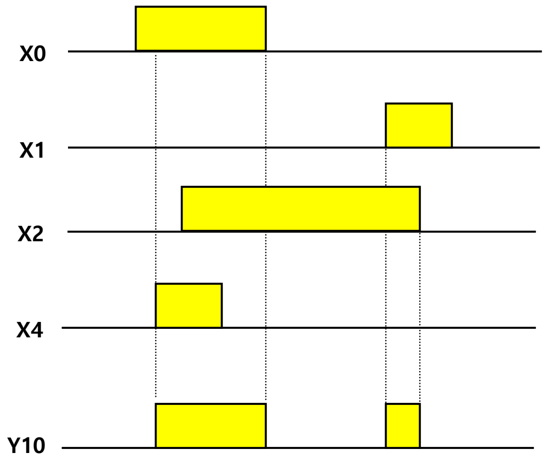
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 OR X1
2 START X2
3 OR X4
4 ANDBLK
5 OUT Y10
6 EOP

▶ **Time Chart**



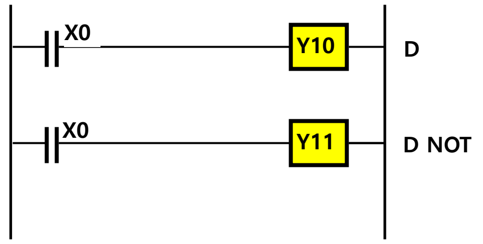
주의

ANDBLK (O)
AND BLK (X)

6.6.6 PULS (=PULSE) / PULSNOT (=PULSENOT)

PULSE / PULSENOT

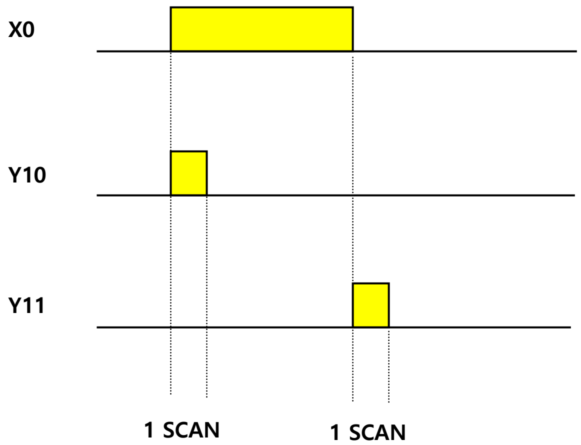
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 PULSE Y10
2 START X0
3 PULSNOT Y11
4 EOP

▶ **Time Chart**

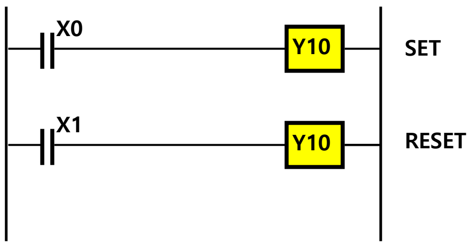


⚠ 주의

PULSNOT (O)
PULS NOT (X)

6.6.7 SET / RESET

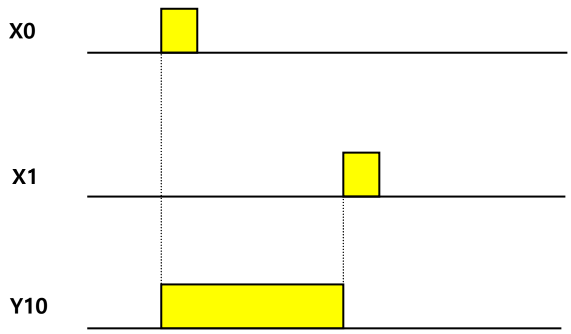
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 SET Y10
2 START X1
3 RESET Y10
4 EOP

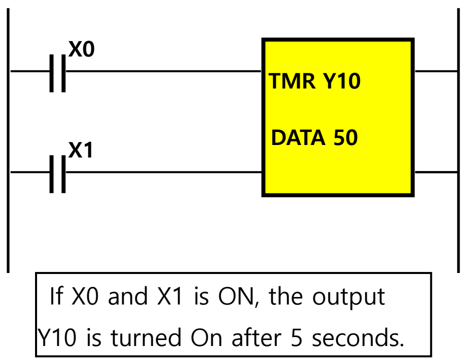
▶ **Time Chart**



6.6.8 **TMR (=TIMER)**

TMR

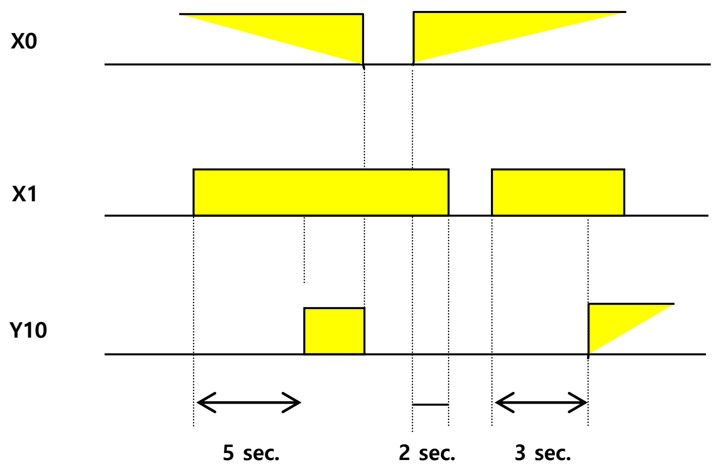
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 START X1
2 TMR Y10
3 DATA 50
4 EOP

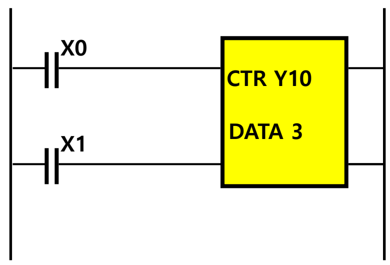
▶ **Time Chart**



6.6.9 CTR (=COUNTER)

CTR

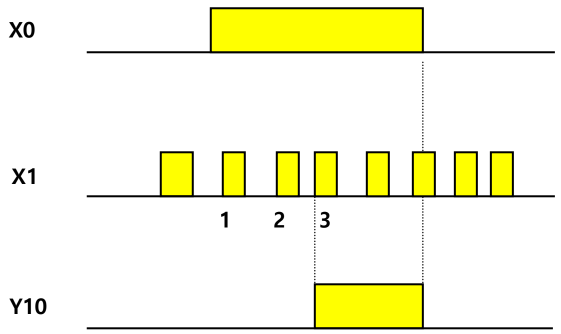
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 START X1
2 CTR Y10
3 DATA 3
4 EOP 0

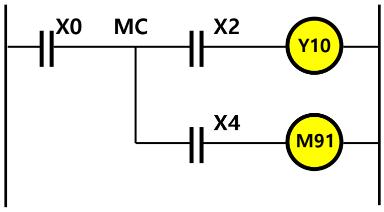
▶ **Time Chart**



6.6.10 MCS / MCSC

MCS / MCSC

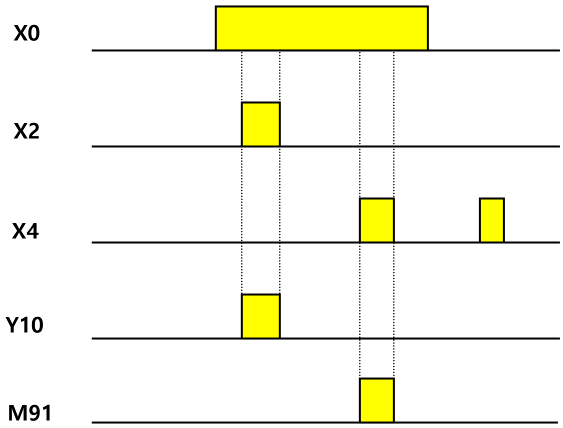
▶ **Sequence Diagram**



▶ **Program**

0 START X0
1 MCS
2 START X2
3 OUT Y10
4 START X4
5 OUT M91
6 MCSC
7 EOP

▶ **Time Chart**



6.7 Use Example between PLC JOB ↔ Robot JOB

Use Example

■ **PLC_0.JOB**

```
START X0
OUT Y15
START X1
OUT Y20
START X2
OUT Y129
START X3
OUT M91
START M110
OUT M234
EOP
```

■ **TEST1.JOB**

```
MAIN
VEL 30
LABL A0
IF OUT15==1 THEN
JMOV P1
OUT110=1
ELSE
IF OUT20==1 THEN
JMOV P2
ELSE
IF IN91==1 THEN
JMOV P3
ELSE
IF IN129==1 THEN
JMOV P4
ELSE
GOTO A0
ENDIF
ENDIF
ENDIF
ENDIF
EOP
```

IF IN15==1 THEN (X)

OUT20=1 (X) Overlapped with OUT Y20 of PLC

IF OUT91==1 THEN (X)
IF M91==1 THEN (X)

IF OUT129==1 THEN (X)
IF M129==1 THEN (X)

■ Comment

① In a simple PLC, 00 to 79 are assigned to User I/O (Of course, in I/O ENABLE, only the setup range is possible). Therefore, taking output signal(Y) as an example, PLC uses them as Y00 to Y79, and a robot program uses the port used in PLC as OUT00 to 79 and cannot use them as IN15(X) and IN20(X). In addition, the output signals having the same number cannot be used. (One of either PLC or robot Program can use them.)

② In a simple PLC, the values 80 to 271 are for Internal port. Therefore, PLC uses them as M80 to M271, and the robot program can use the port used in PLC as IN80 to IN271 only and cannot use them as M80 to M271(X).

③ OUT110 used in the robot Program can be used as M110 by PLC because it is for Internal port.

Use Example

■ **Robot Program**

```
MAIN
VEL 50
WHILE 1
IN0=1
JMOV P10
OUT0=1
JMOV P12
DLAY 100
PMOV P1 P15
OUT1=1
DLAY 100
ENDWL
```

■ **Simple PLC Program**

```
START Y0
AND Y1
OUT Y2
EOP
```

```
START X0 (X)
START M0 (X)
```

```
AND X1 (X)
AND M1 (X)
```

■ Comment

▷ PLC recognizes OUT0=1, OUT1=1 of robot Program and outputs OUT2=1.
(★ wherein OUT0, 1 is User I/O.)

■ **Robot Program**

```
MAIN
VEL 50
WHILE 1
IN0=1
JMOV P1
OUT100=1
DLAY 100
IN1=1
JMOV P2
OUT101=1
DLAY 100
ENDWL
```

■ **Simple PLC Program**

```
START Y100
AND Y101
OUT Y2
EOP
```

```
START X100 (X)
START M100 (O)
```

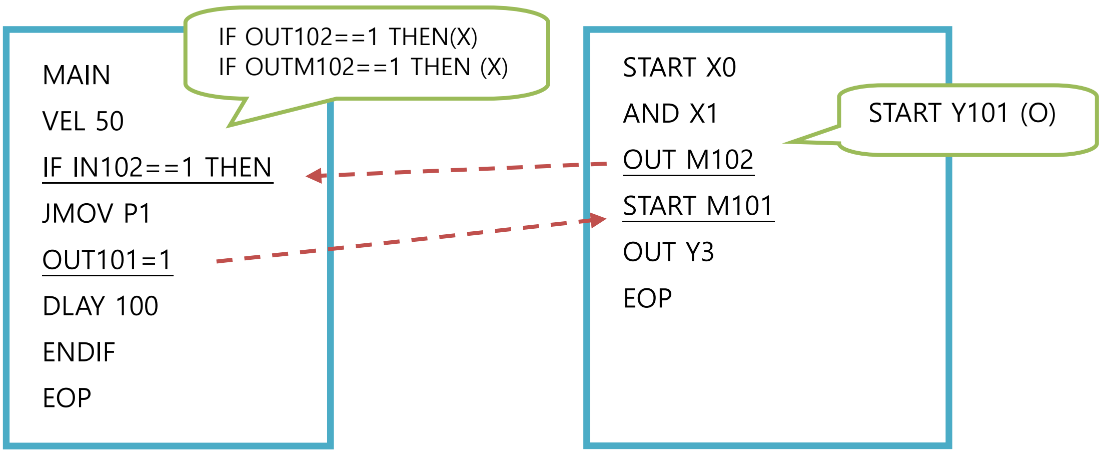
```
AND X101 (X)
AND M101 (O)
```

■ Comment

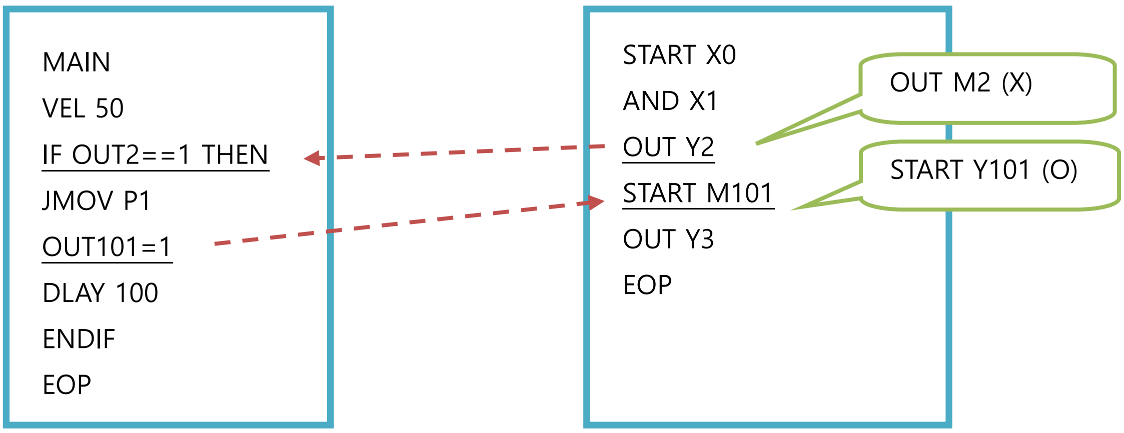
▷ PLC recognizes OUT100=1, OUT101=1 of robot Program and outputs OUT2=1.
(★ wherein OUT100, 101 is Option I/O.)

Use Example

■ **Robot Program** ■ **Simple PLC Program**



■ **Robot Program** ■ **Simple PLC Program**



Rev.	Revised Date	Description	Reviser	**S/W Version**
V.1	2012.07.30	First Edition Printed		
V.2	2012.12.31	Add content	Kimjs	

N1 ROBOT CONTROLLER

CONTROLLER MANUAL

FIRST EDITION JULY 2012

ROBOSTAR CO, LTD

ROBOT R&D CENTER