NSK

Robot Module System

- P Series Module Main Unit
- R Series Module Main Unit
- EXEA Controller

User's Manual 2

= Programming and Operation of EXEA Controller =

1 Installation and Maintenance of EXEA Controller

1. Introduction
2. Safety Precautions
3. System Configuration
4. Glossary
5. Reference Number • Specifications
6. Unpacking and Installation
7. Wiring
8. Startup
9. Initial Setting
10. Trial Running
11. Protection and Safety
12. Maintenance • Checking
13. Alarms
14. Troubleshooting

Appendix

2 Programming and Operation of EXEA Controller

15. Programming
16. Description of Function
17. Operation of Robot Module
18. Remote Control Operation

3 Installation and Maintenance of Module Main Unit

19. Reference Number • Specifications
20. Unpacking
21. Installation
22. Maintenance • Checking

M–E099XE0K2–023

NSK Ltd.

Document Number: K20079-01

EC-T

Limited Warranty

NSK Ltd. warrants its products to be free from defects in material and/or workmanship which NSK Ltd. is notified of in writing within, which comes first, one (1) year of shipment or 2400 total operation hours. NSK Ltd., at its option, and with transportation charges prepaid by the claimant, will repair or replace any product which has been proved to the satisfaction of NSK Ltd. to have a defect in material and/or workmanship.

This warranty is the sole and exclusive remedy available, and under no circumstances shall NSK Ltd. be liable for any consequential damages, loss of profits and/or personal injury as a result of claim arising under this limited warranty. NSK Ltd. makes no other warranty express or implied, and disclaims any warranties for fitness for a particular purpose or merchantability.

Robot Module System
EC Directives Conformity

NSK Ltd. declares that "Robot Module System" conforms to EC Directive (CE Marking).

However, please note that the following conditions are added for conformity to the EC directive.

⊙ EC Declaration of Incorporation

- NSK Ltd. declares that the Robot Module System is a machine component which is to be incorporated into the machine. (EC Declaration of Incorporation)
- The Robot Module System must not be operated until it is incorporated to the machine.
- The Robot Module System, as the machine component, conforms with following EC Directives.
  - ◊ EC Machinery Directive 89/392 as amended 94/368 and 93/44.
  - ◊ EC Low Voltage Directive 73/23 as amended 93/68.
- The customer has to take appropriate measures to its machine to conform to Electromagnetic Compatibility Directive. The Robot Module must not put into service until the machinery into which it to be incorporated has been declared in conformity with the provisions of EC Directives.
- Our declaration becomes invalid if technical or operational modifications are introduced without the consent of Mechatronics Technology Department of NSK Ltd.

⊙ Remaining Hazards (Following notes should be observed for your safety.)

- EXEA controller shall be put into the enclosure conforming to relevant European standard in terms of fire protection and electrical shock protection. The protection grade of the enclosure must be IP 54 or better. EXEA controller shall not be exposed to water or oil.
- Just after the power is turned on and off, there will be the hazardous voltage on the parts of EXEA controller, such as the power input terminal, motor connector and connector for an external regenerative dump resistor. Put covers on those parts to protect from touching when operating the machine or doing maintenance work.
  Furthermore, provide appropriate protection from disconnecting the motor connector accidentally.
- An isolation transformer must be used to prevent electrical shock. The isolation transformer must have enough capacity for the Robot Module System power consumption.
- Install noise filter in the primary AC power line as a measure for Electromagnetic Compatibility Directive.
- A circuit breaker must be installed to the primary AC power line of Robot Module System.
- Ground earthing must be provided to EXEA controller.
- Wiring inside of EXEA controller is simply internal wirings and the grounding wire is not distinguished by color as the protective grounding conductive.
- Secure the controller cables and motor cables firmly so that those cables do not break or have loose contact.
- Surround the machine, to which the Robot Module System is incorporated, with safety fence to prevent any personnel from entering its moving range.

⊙ Unit Limitation

- Units of Robot Module System which conform to EC Directives are limited to the following reference number only.

  1. EXEA controller

     Reference No. : M-EXEA □ – □ □ □ □ T □ □

                                                  ↑

                                  T : Indicates conformity with the Directive

  2. Teaching Box

     Reference No. : M-EXTB 04

- However, all robot module main units are compatible with the EC Directives. If you require to build the Robot Module System that complies to the EC Directives, the EXEA controller and the Teaching Box must be compatible with the EC Directives.

Contents

(Blank Page)

15. Programming

- It is necessary to make operation programs prior to move robot module automatically. On receiving a start command, the robot module executes the operation in accordance with the program.

- Programming is referred to as making instructions by converting sequential operations into the combination of program command (robot language) provided for the EXEA controller.

- The motion commands that execute positioning specify the point register as a destination of motion. Setting the coordinates data to a point register is called "Teaching." The teaching decides the destinations of positioning operation of the robot module.

[Example] MOV P0000 : Linear motion to point P0000

Motion command (linear) = Move linearly to the point of which coordinate data is written on the point register P0000.

Specify a position register P0000 as the destination.

15.1. Teaching

- Teaching is referred to as a procedure to determine a coordinates data of point register (= data of point register) of a destination point of the motion command in the programmed operation.

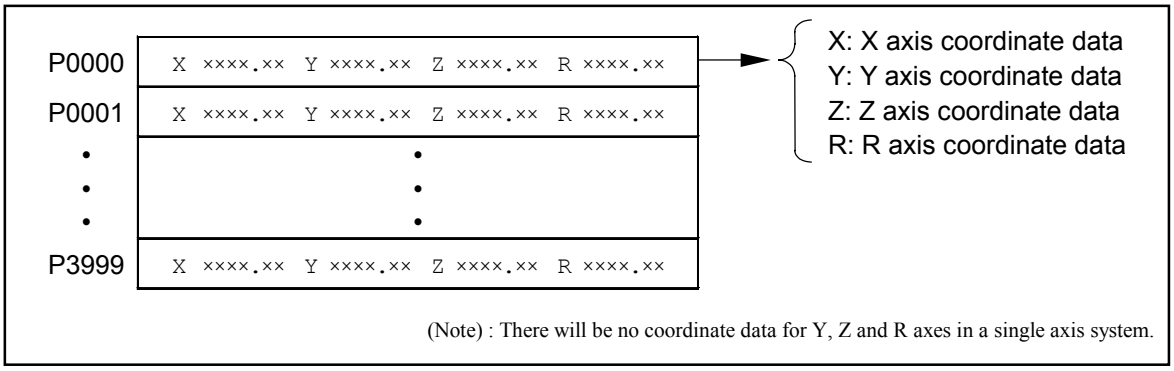
- The unit of data is millimeter and its resolution is 0.01 mm.
  [Example] X0200.05 • • • X axis coordinate = 200.05 mm

  * "××××.××" denotes that the robot does not move. When the teaching gives "××××.××," the robot does not move even the motion command is inputted. (Holds current position.)

- Number of point register is available in P0000 to P3999, allowing you to make up to 4000 points.

Caution : *After the teaching, be sure to save the data to the memory. All data will be lost if the power is turned off before data is stored. Refer to "15.1.4. Saving Point Data."*

Figure 15-1: Construction of point register



- There are three ways to make the coordinate data.
  1) Move the slider to the desired position by jog operation, then set the point register. (Actual setting)
  2) Input numerical data manually through the teaching box. (Manual data input)
  3) Find out the coordinates arithmetically by a program operating command while programming.
     This is used when changing the coordinate data of positioning point register while programming.

- When it is necessary to change the teaching data, specify the point register number and perform teaching again. New data will be overwritten.

- Two data shown below may be set as the additional information of teaching.
  1) Unit number (U∗) (Only U1 can be set for a single axis system.)
  2) Coordinate format: A (Absolute position) or I (Relative position)
     (Relative position is not applicable when setting the point register by Jog operation.)

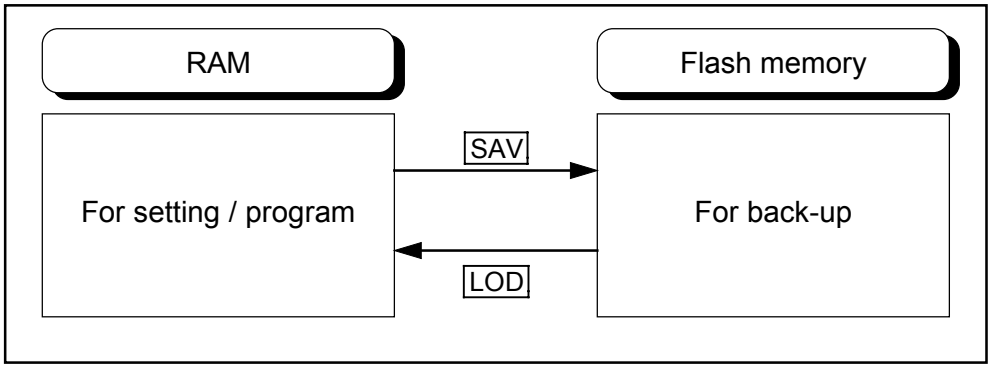
  These data are used for the direct operation. Refer to "16.2. Direct Operation" for detail.

- Teaching data won't be stored automatically. They are lost if the power is turned off before saving them. For saving data, select [SAV].

- Select [LOD] to read out the stored data.

Data of the point registers are common to all programs. They won't be lost after the main power is turned off, if they are saved in advance.

Figure 15-2: Construction of Memory

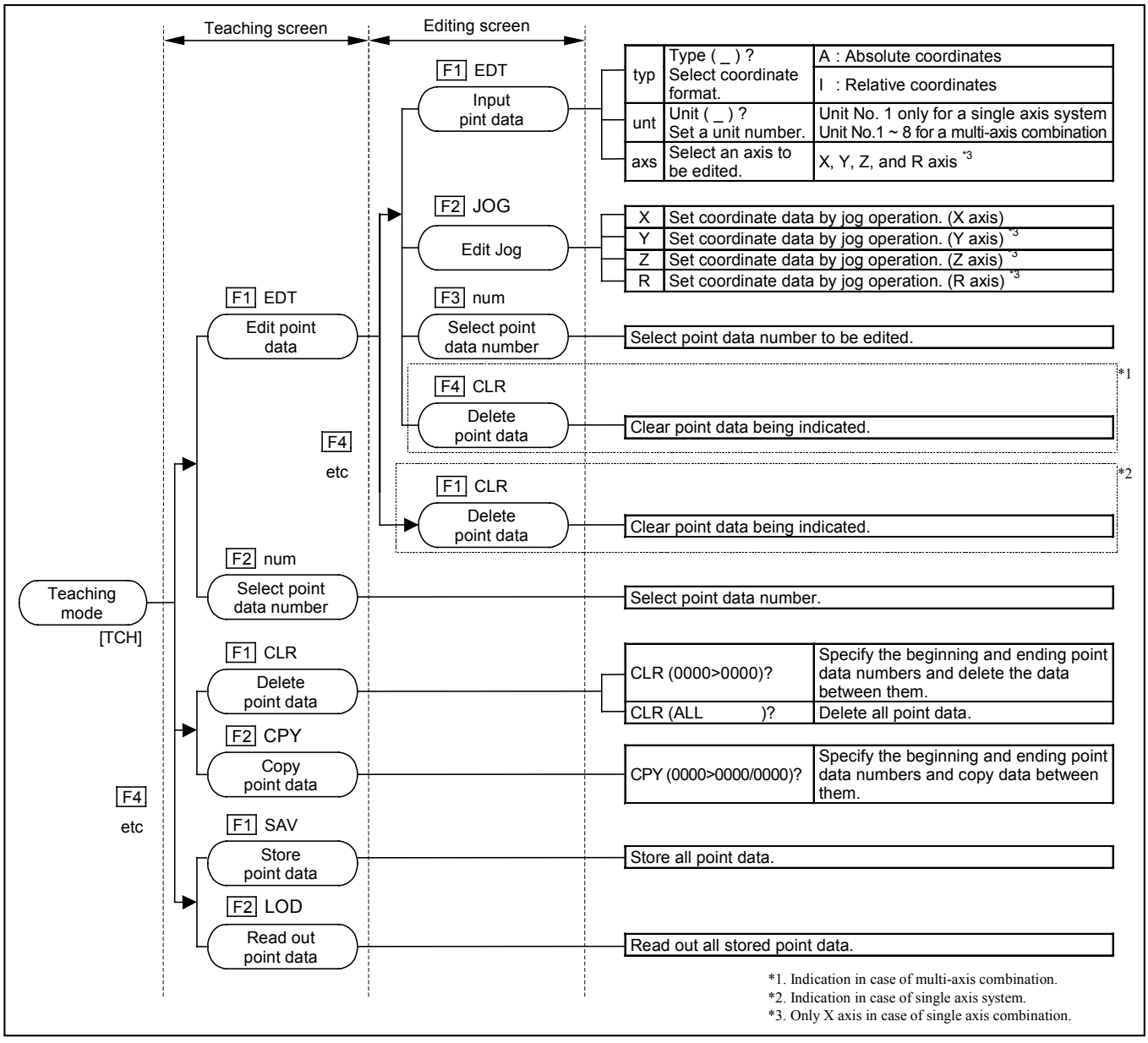


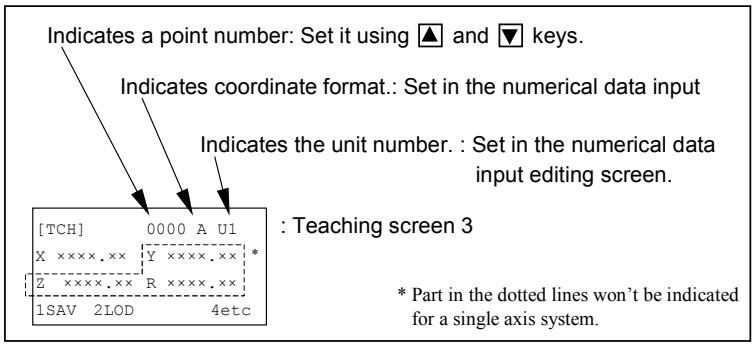
For simplification in the description hereafter, "point register data" will be referred to as "Point data" and "point register number" will be referred to as "point number."

15.1.1. List of Teaching Function

- Selection of F2 TCH in the Teaching menu screen will make you to get into Teaching mode.
- Setting function of teaching mode is like a tree structure as shown in Figure 15-3. These functions can be set by selecting the function keys on the teaching box.

Figure 15-3: List of teaching mode





15.1.2. Teaching by Jog Operation

- Move the slider actually with jog keys (+X, -X, +Y, -Y, +Z, -Z, +R and -R) of the teaching box to specify the point data to be used for positioning.

  * Use +X and -X only for a single axis system.

- Moving speed and acceleration shall be set by "Jog speed (L)" and "Jog accel" which are described in "9.3.3. Parameters for Jog Operation."

  Danger : Do not enter working area of the robot. Such hazards exist that the robot would hit, nip or catch a person within its working area when the module main unit or EXEA controller malfunctions.

  ◊ If you conduct the teaching beyond the guard fence, be careful not to get in the working area of the robot. Stay in the area from where you can clearly see its motion. Be sure to set to safety speed. (250 mm/Sec or lower)

  * Shipping set of the jog and teaching speed is 50 mm/Sec.

  ◊ Conduct the jog operation before teaching to confirm that it is set to the safety speed or lower.

- It requires completion of the home return before the teaching by jog operation.

15.1.2.1. Teaching Procedure by Jog Operation

1) Select the editing screen in the teaching screen 1 by pressing F1 key (EDT).

2) Specify a point number using ▲ and ▼ keys.

3) Then, press F2 key (JOG) to make it possible to perform the jog operation for teaching. An alarm arises if the home return is not completed at this moment.

4) Specify a unit number using ◀ and ▶ keys. (for multi-axis combination only)

5) Use F1, F2, F3 and F4 keys to select or default an axis.

   * If an axis is selected, the screen indicated current position while "××××.××" is indicated for a defaulted axis.

   * Use F1 key only for a single axis system.

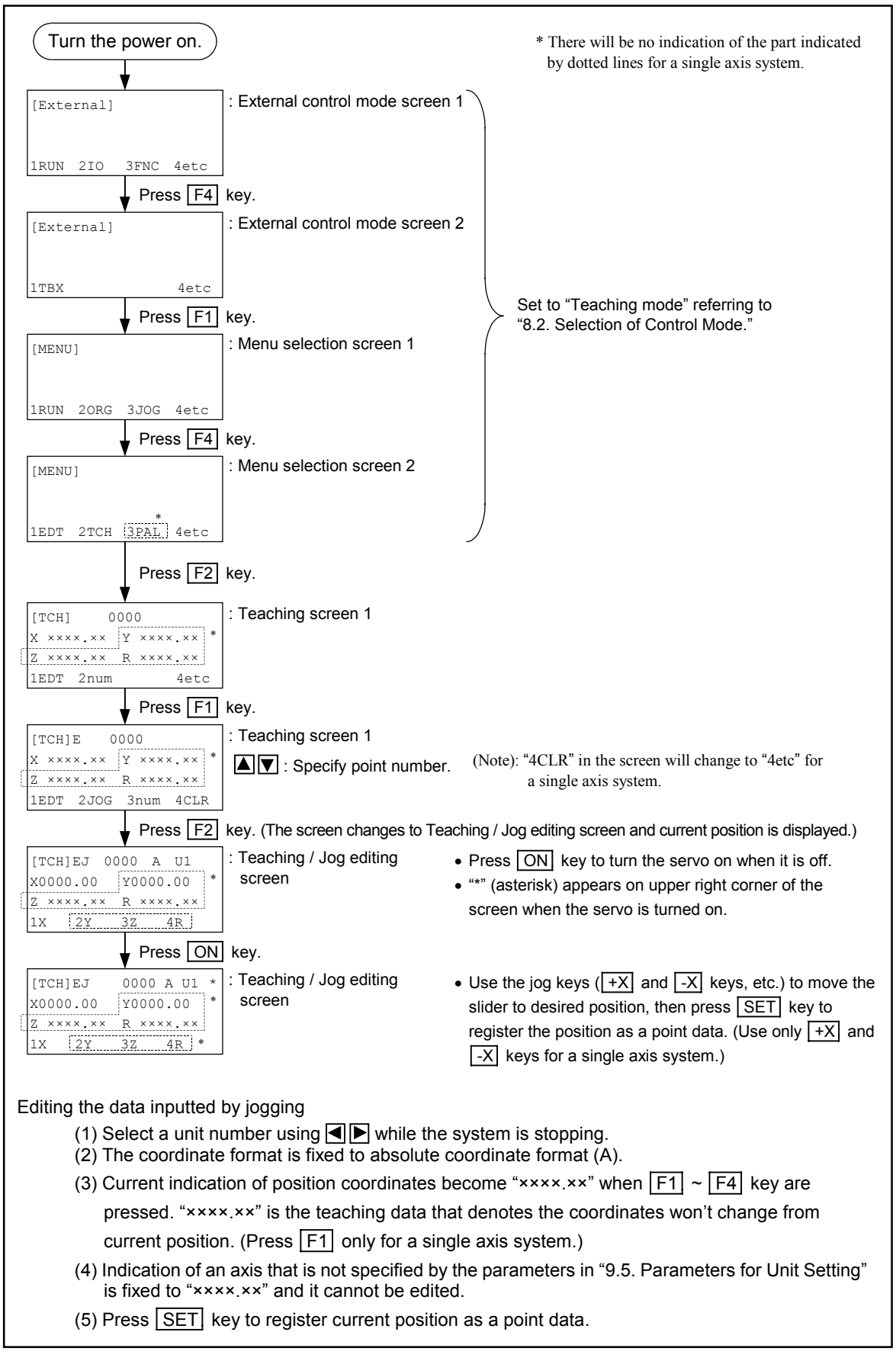
6) Turn the servo on by ON key and move the robot by the jog keys (+X and -X etc.).

   * Use +X and -X keys only for a single axis system.

7) The coordinate format of position data is absolute coordinates.

   * Refer to Figure 15-4: Teaching procedure by jog operation.

Figure 15-4: Teaching procedure by jog operation



Editing the data inputted by jogging

(1) Select a unit number using ◀ ▶ while the system is stopping.
(2) The coordinate format is fixed to absolute coordinate format (A).
(3) Current indication of position coordinates become "××××.××" when F1 ~ F4 key are pressed. "××××.××" is the teaching data that denotes the coordinates won't change from current position. (Press F1 only for a single axis system.)
(4) Indication of an axis that is not specified by the parameters in "9.5. Parameters for Unit Setting" is fixed to "××××.××" and it cannot be edited.
(5) Press SET key to register current position as a point data.

15.1.2.2. Automatic Positioning to Editing Point

- There is a function to move the robot automatically to a set point. This is to confirm position of point data to be set and to move the robot to the specified point.
- A module main unit moves to the specified point automatically. The jog keys are disabled meanwhile.
- Motion speed and acceleration is set by "Jog speed (L)" and "Jog accel" described in "9.3.3. Parameters for Jog Operation."

(1) Press F1 key (EDT) in the teaching screen 1 to get in the editing screen.

(2) Specify a point number to be the position of destination using ▲ and ▼ keys.

(3) Specify coordinates data by numeric keys when the point data is not registered yet.

(4) Press START key to go to the teaching screen and you can operate the motion to the specified position.

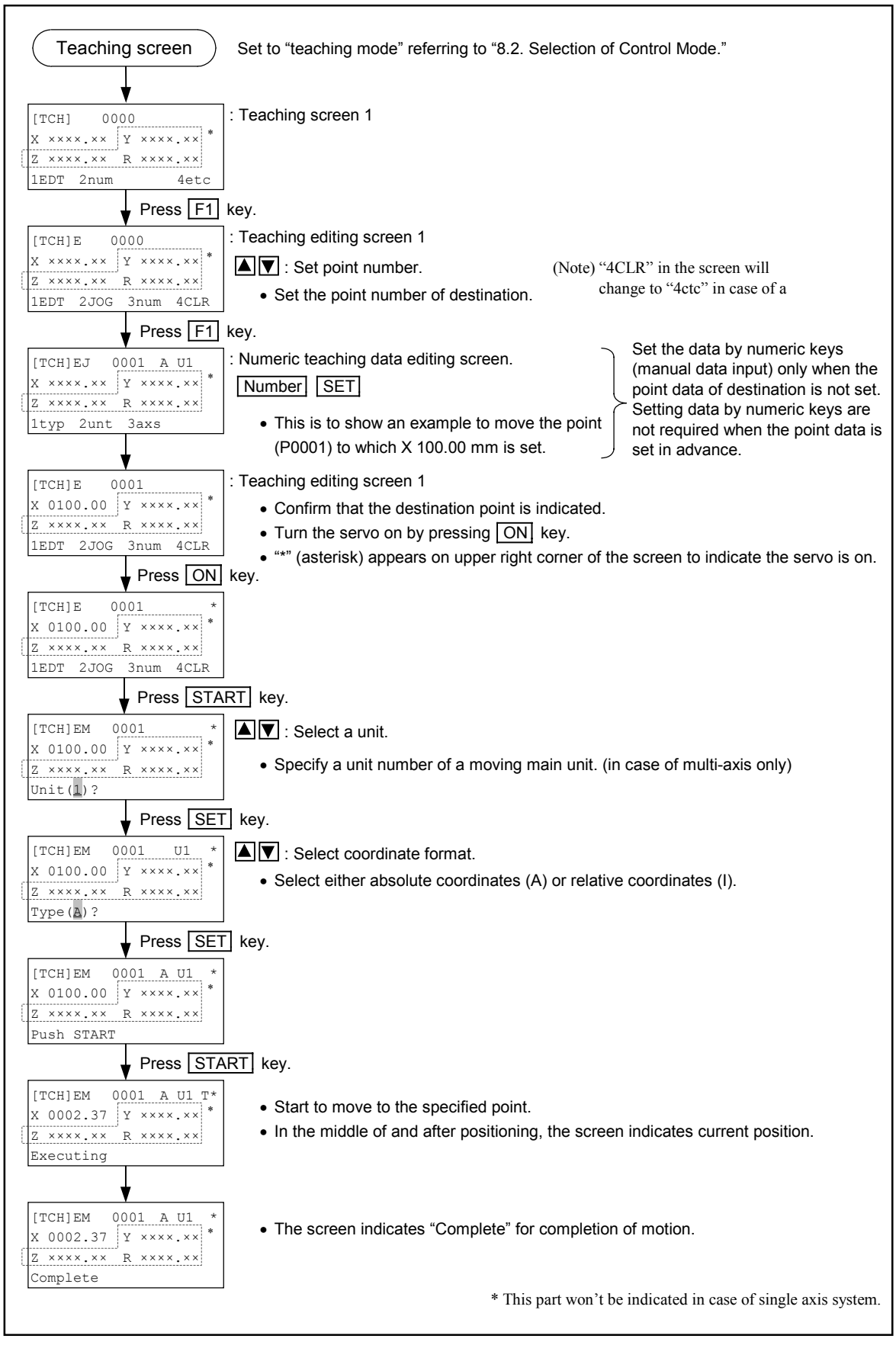
(5) Use ▲ and ▼ keys to select a Unit number and press SET key. (for multi-axis combination only)

(6) Use ▲ ▼ keys to select coordinate format and press SET key.
(A: Absolute coordinate, I: Relative coordinates)

(7) Press START key to move the robot to the specified position. An alarm may arise if the home return is not completed at this moment.

(8) "Complete" will be indicated when the motion completes.

Figure 15-5: Procedure to make to specified point

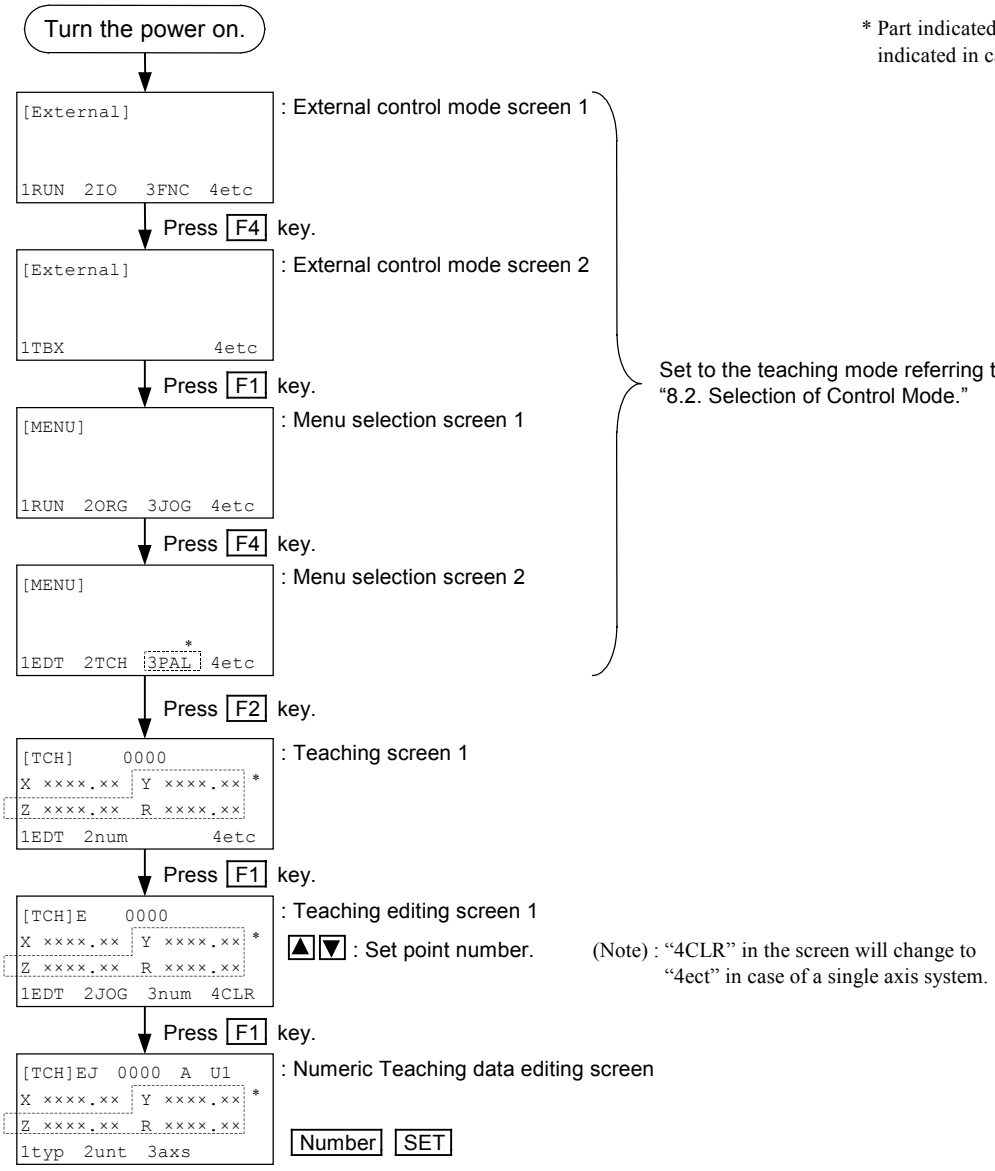


15.1.3. Teaching by Manual Data Input

- This is the teaching to input coordinate data of the designated point number using numerical keys of the teaching box.
- For manual data input, the home return is not necessarily completed in advance.

* Refer to Figure 15-6 for procedures of manual data input.

Figure 15-6: Procedure for manual data input



Editing manual data input

(1) Use ◀ and ▶ keys to select an axis. (This process is not necessary for a single axis system as the unit number is fixed to U1.)

(2) Input numerical data using numeric keys and press SET key for execution.

(3) When numeric data input is not necessary (denotes that the axis unit does not move.), press CLR and SET for confirmation. Indication of the screen will change to "××××.××"

(4) Select the coordinate format using F1 keys.
Select either absolute coordinates or relative coordinates by ▲ and ▼ keys and press SET for confirmation. Press MODE key to cancel the selection. Absolute position data or relative position data through the teaching may be used in direct operation.

(5) Set a unit number by F2 key. The unit number will be used for direct operation. Use ▲ and ▼ keys for input and press SET key to set. Press MODE key to cancel the setting. (Only a unit number U1 can be set for a single axis system.)

15.1.4. Saving Point Data

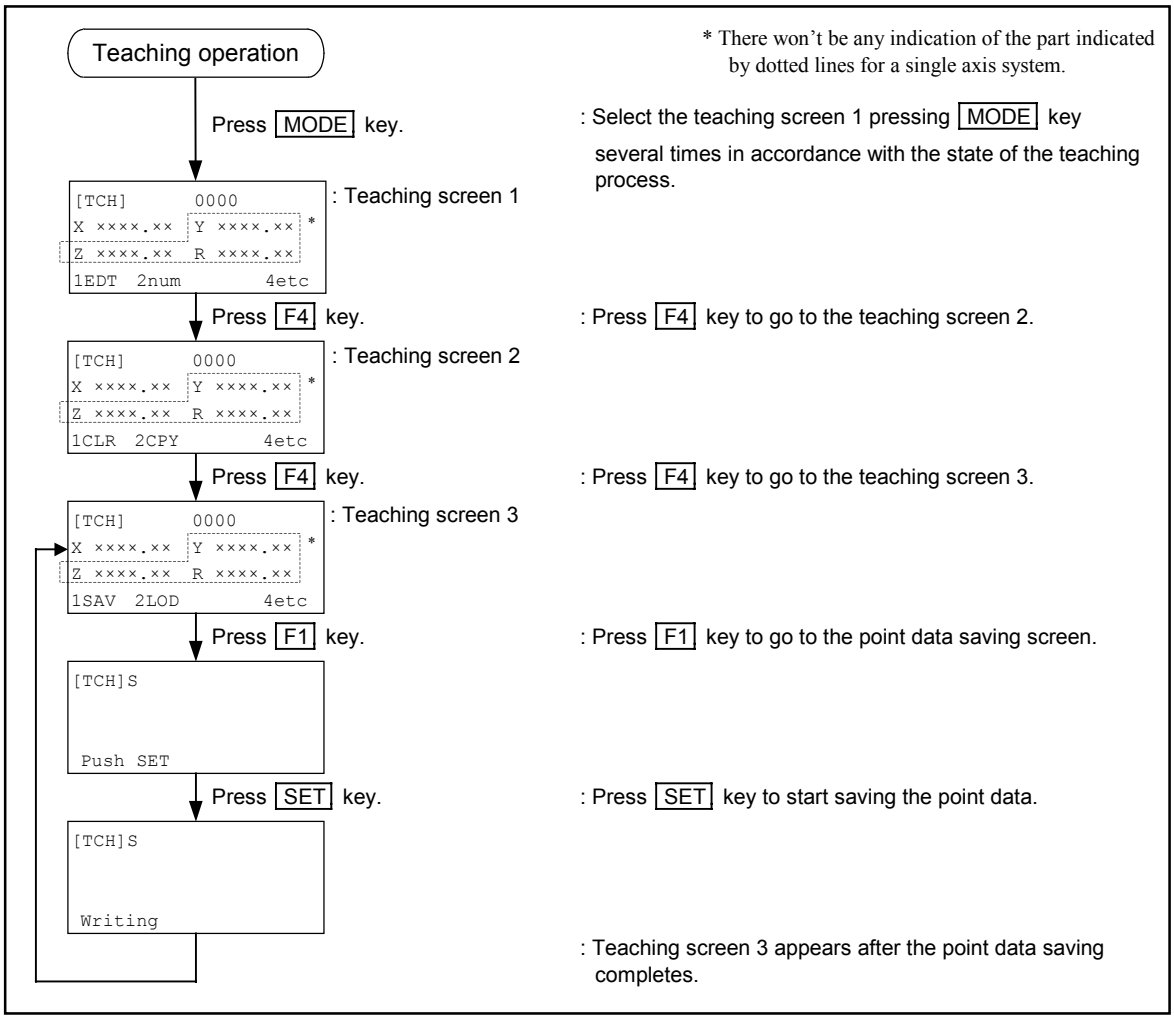
Caution : After completion of the teaching by jog operation or manual data input, be sure to save point data. All point data of 4000 point registers can be saved. The point data will be lost if the power is turned of before they are saved.

Caution : "Writing" message will appear on the screen. Do not turn off the power before all data are saved. If not, it leads to "memory error" when turn on the power again.

Procedure for saving pint data

- Press F1 key (SAV) in the teaching screen 3.

Figure 15-7: Saving point data



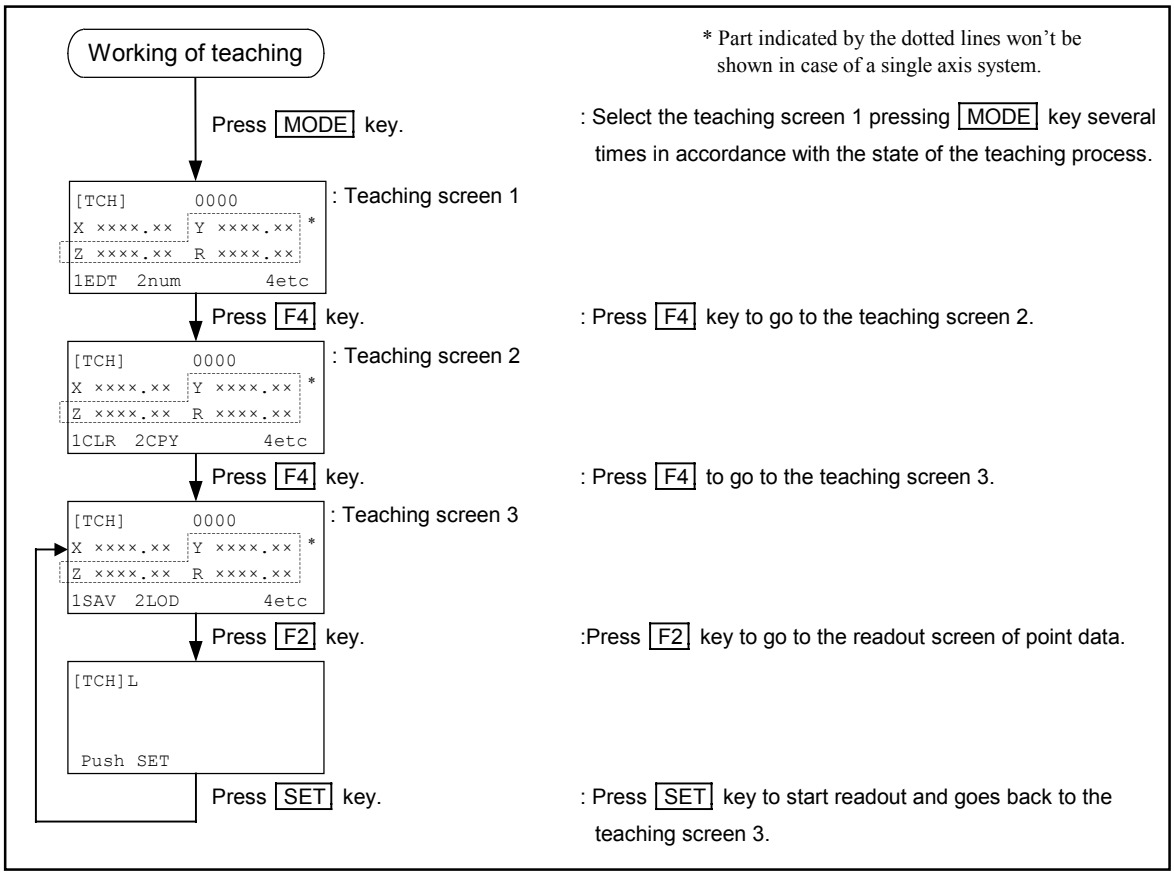
15.1.5. Readout of Point Data

- This function is to read out the stored data. Be careful that execution of readout overwrites point data currently being edited. All point data in 4000 point register can be read out.

Procedure for readout point data

- Press F2 key in the teaching screen 3.

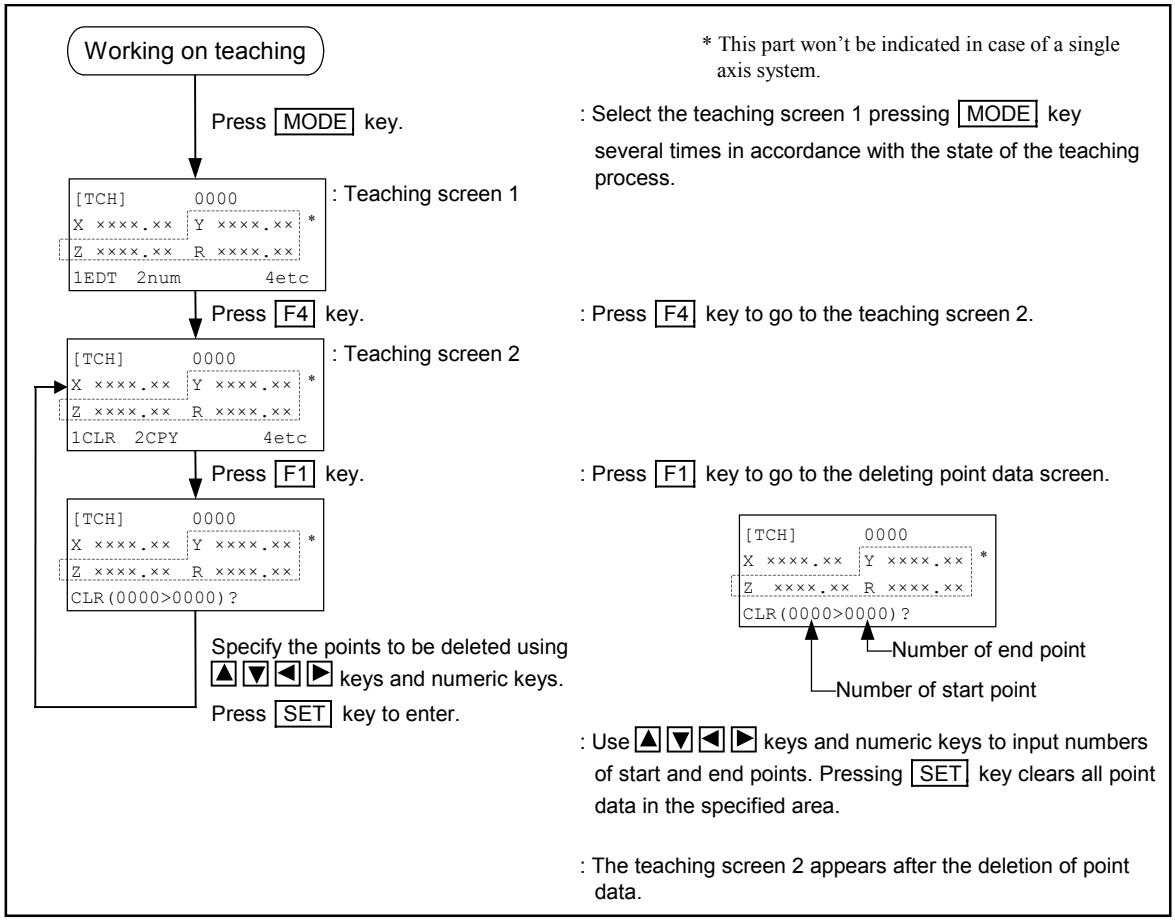
Figure 15-8: Readout of point data



15.1.6. Deleting point Data

- Point data can be deleted by pressing F1 key in the teaching screen 2.

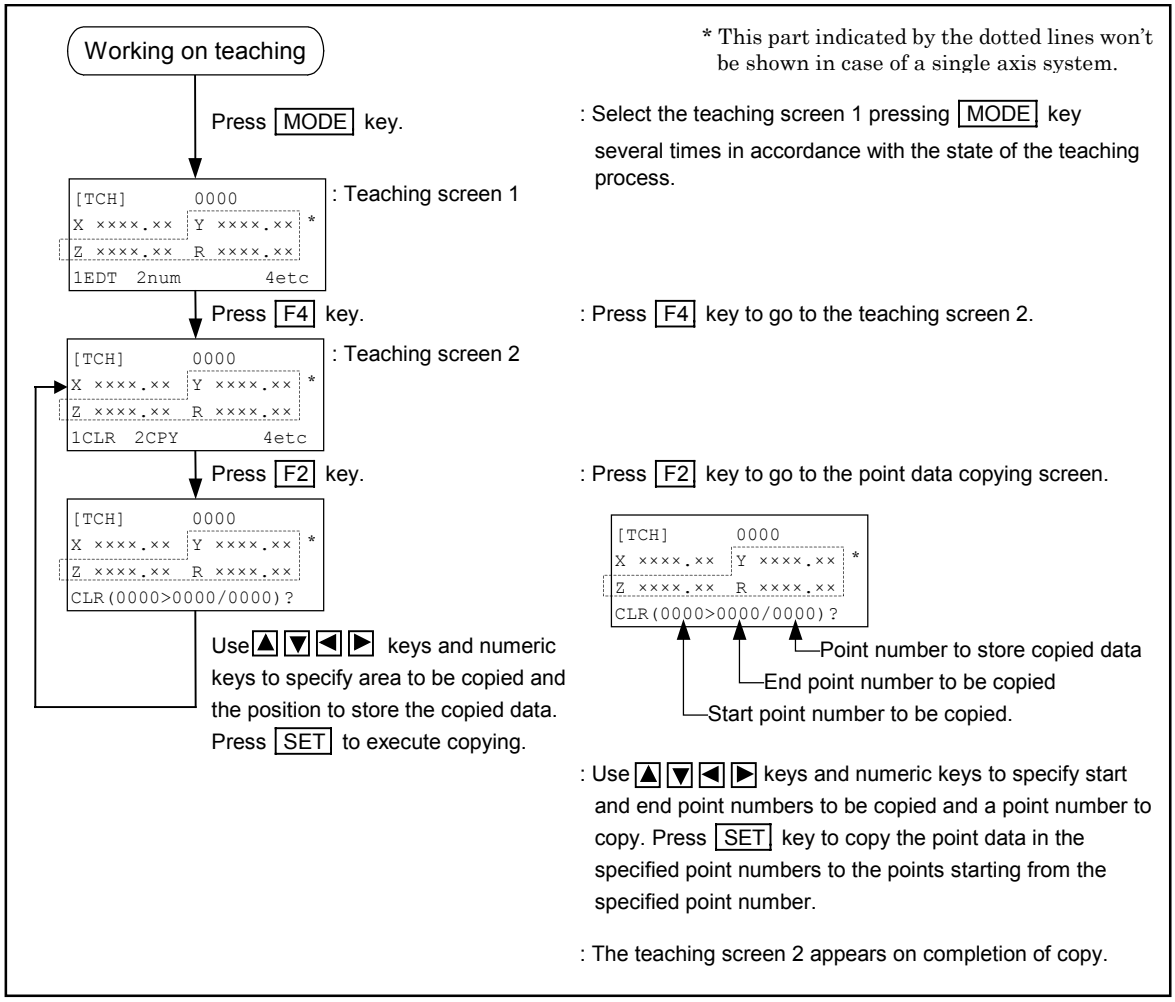
Figure 15-9: Deleting point data



15.1.7. Copying Point Data

- Pressing F2 key in the teaching screen 2 makes it possible to copy the point data.

Figure 15-10: Copying point data



15.2. Programming

- This section describes the fundamental ways of operation and examples of programming which are necessary to make and edit the program.

  Caution : *Be sure to store the program to the flash memory after editing it. All program will be lost if the power is turned off before storing them. Refer to "15.2.3.5. Procedure for Saving Program."*

15.2.1. Programming Area

- This is the area the programs are written in.
- No. 0 to 127 program channels are available. (Totally 128 channels)
- Each program can be comprised of up to 1000 (000 to 999) steps. (Refer to Figure 15-11.)
- Total number of steps that can be set to a whole program depends on the code length of programmed commands. If all steps of a program consist of only simple commands with command code length of "1," approximately 45 000 steps are the limit of total steps. (Refer to "Table 15-1: Reference of command code length."
- Attempting to set more steps to a program, which has already full of steps, arises "Out of memory" alarm and EXEA controller does not accept any steps.
- As the program area is consumed by command code length, an equivalent number of the steps for command code length shall be added to the count of total number of steps. The table below shows examples of command code length.

[Example]

Program number	Step number	Example of program	Command code length		
0	0	ABCD	3		
	1	SPD S100.0 A1.0 B2.0	8		
	2	MOV P0001	3		
	3	MOV P0002	3		
	≀	≀	⋮	300	
	101	MOV P0110	3		
	102	TIM #1.00	5		817
	103	ARC P1001 P1002	5		
	104	ARC P1003 P1004	5		
	≀	≀	⋮	500	
	202	ARC P1009 P1010	5		
	203	END	1		

* When the program "0" in the above example is copied to the program numbers 1 ~ 54, the total equivalent step length will be (817 × 55 = 44 935). Remaining program area will be (45 000 - 44 935 = 65 steps). You cannot add another program of the same size as the program "0."

* Allowable number of steps of a program is 204. However, in this example, equivalent number of steps (command code length) is 817. You can make a program which has less than 45 000 equivalent steps.

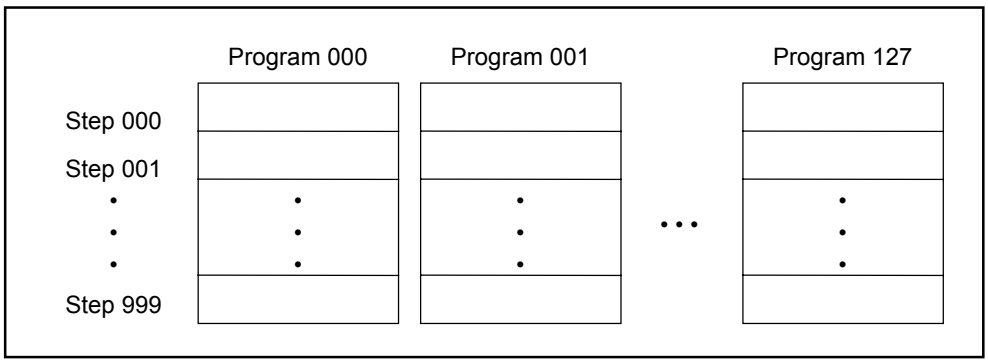
* This example is for a multi-axis combination. Estimation of code length is the same for a single axis system.

Table 15-1: Reference of command code length

Example of program command	Command code length
(Blank)	1
UNT U1	3
SPD S600.0	6
SPD S600.0 A35.0 B35.0	8
SPD U1 S600.0 A35.0 B35.0	10
TYP &A	3
TYP U1 &A	5
NOF D000	3
NOF U1 #1	6
PBS P0000	3
PBS U1 X5.00	9
* PBS U1 X5.00 Y10.00 Z15.00 R20.00	15
* ESCZ D001 D002 D003	7
* ESCZ U1 #10.00 #100.00 #200.00	15
LD D000 = #100.00	9
LD P0000 = X10.00	11
* LD P0000 = X10.00 Y20.00 Z30.00 R40.00	17
LDS D000 = SPD	7
LDS P0000 = PBS	7
* LDS PX0000 = U1 ESCZ UPR	11
CAL D000 = D000 + D001	11
CAL D000 = #100.55 + #200.55	15
CAL P0000 = X1.00 + X5.00	19
* CAL P0000 = X1.00 Y2.00 Z3.00 R4.00 + X5.00 Y6.00 Z7.00 R8.00	31
TCH P0000 = X1.00	11
* TCH P0000 = X1.00 Y2.00 Z3.00 R4.00	17
TCH P0000 = U1 X@D001	10
* TCH P0000 = U1 X@D001 Y@D002 Z@D003 R@D004	13
OUT OB000 = ON	7
OUT OP00 = ;00000000	8
INP D000 = IP00	7
LCAL D000 = D001 AND D002	11
SRV ON	3
HOM	1
HOM ALL	3
HOM U1 X CUR	7
* HOM U1 X Y Z R CUR	7
MOV P0000	3
MOV P0000 &A	5
MOV U1 X1.00	18
* MOV U1 X1.00 Y2.00 Z3.00 R4.00 S100.00 A10.0 B1.0 &A	24
MOVM P0000 P0001	5
MOVM U1 P0000 P1111 S600.0 A35.0 B35.0 &A	16
MSTP ALL	3
MSTP U1 X	5
MSTS D000	3
MSTS D000 = U1 X	9
* MSTS D000 = U1 X Y Z R	9
* ARC P0000 P0001	5
* ARC P0000 P0001 P0002 &A	9
* ARC P0000 P0001 P0002 S600.0 A35.0 B35.0 &A	16
* PALI QN00 D000	5
* PALI U1 QN00 $ABCDEF &A	12
* PALM QN00	3
* PALM QN00 QP0000	5
* PALM QN00 QP0000 S600.0 A1.0 B1.0 &A	14
* PALN QN00 D000	5
* QSTS D000 = QN00 QPM	9
* CPS	1
* CPS S10.0 A1.0 B1.0	8
* CPS U1 S10.0 A1.0 B1.0 &A	12
CPE*, RET, NXT	1
' ABCD (Four characters)	3
TAG _06	3
JEQ _00, JGE _00, JGT _00, JLE _00, JLT _00, JNE _00, JMP _00	3
END	1
END CSTP	3
CMP OP00 ;00000000	5
CMP D000 D001 JEQ _ABCD	12
TIM D000	3
TIM #1.00	5
CALL $PRGNAM	6
REP #10	4
WAIT D000 D001 EQ	7
WAIT D000 #10 EQ #5.00	13
CHG D000	3
CHG $PRGNAM CSTP	8

* Program command for multi-axis combination only.

Figure 15-11: Programing area



- One program command may be written on one step.
- Operating the robot module by the program is called "programmed operation."
- Select program numbers (0 ~ 127) to start the programmed operation through the teaching box, control I/O (CN3) or remote communication with a personal computer.
- When the programmed operation of a selected program number starts, the program executes its steps in due order from 000.

15.2.2. List of Programming Mode

- Pressing F1 key in the menu screen of teaching will lead to the programming mode.

- Programming and editing program functions comprises a tree structure as shown in Figure 15-12. You can select and set each item by the function keys of the teaching box.

Figure 15-12

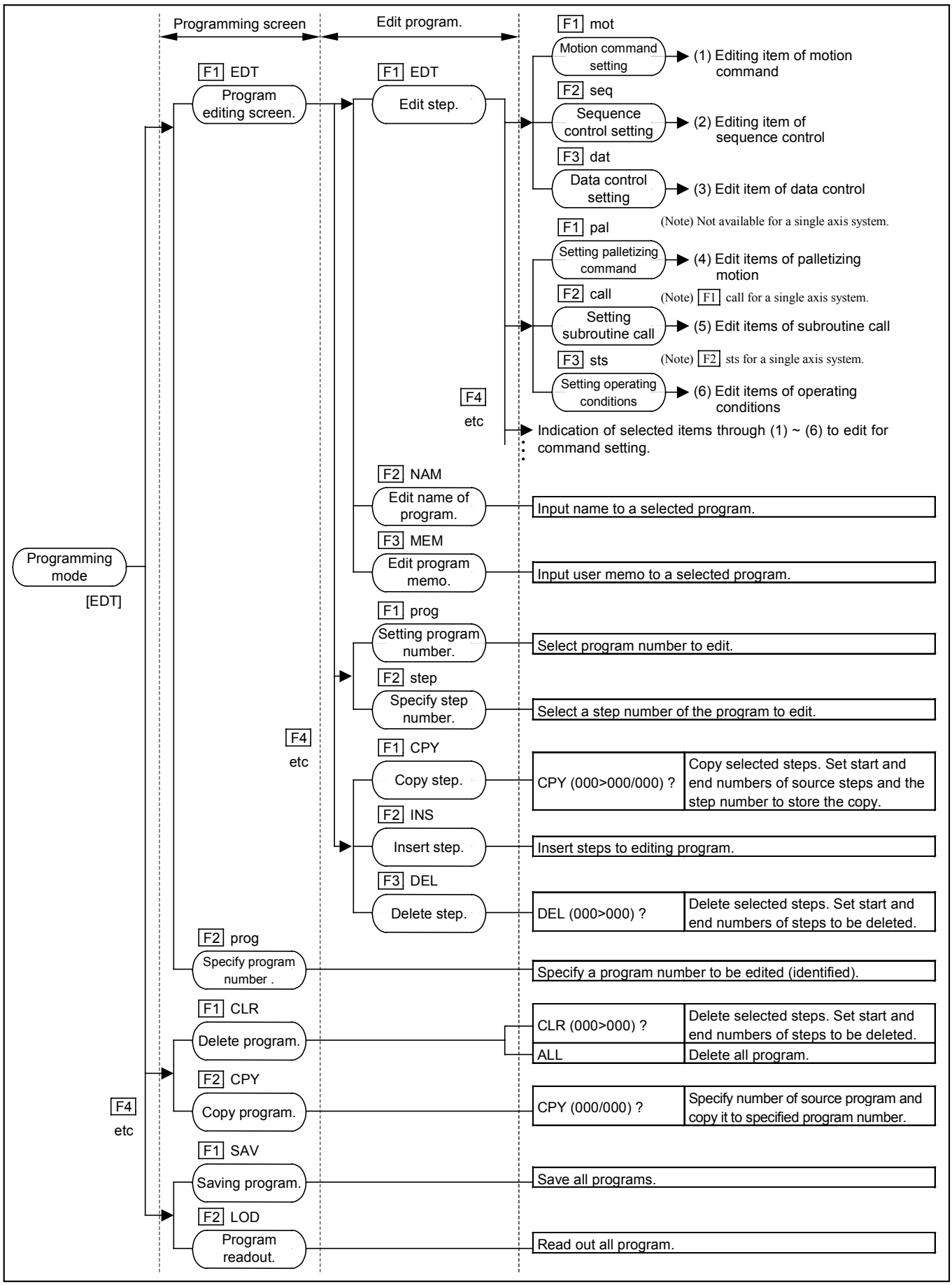

Programming screen
Edit program.
F1 EDT
Program editing screen.
F1 EDT
Edit step.
F1 mot
Motion command setting
(1) Editing item of motion command
F2 seq
Sequence control setting
(2) Editing item of sequence control
F3 dat
Data control setting
(3) Edit item of data control
(Note) Not available for a single axis system.
F1 pal
Setting palletizing command
(4) Edit items of palletizing motion
F2 call
(Note) F1 call for a single axis system.
Setting subroutine call
(5) Edit items of subroutine call
F3 sts
(Note) F2 sts for a single axis system.
Setting operating conditions
(6) Edit items of operating conditions
F4 etc
Indication of selected items through (1) ~ (6) to edit for command setting.
F2 NAM
Edit name of program.
Input name to a selected program.
F3 MEM
Edit program memo.
Input user memo to a selected program.
Programming mode
[EDT]
F1 prog
Setting program number.
Select program number to edit.
F2 step
Specify step number.
Select a step number of the program to edit.
F4 etc
F1 CPY
Copy step.
CPY (000>000/000) ?
Copy selected steps. Set start and end numbers of source steps and the step number to store the copy.
F2 INS
Insert step.
Insert steps to editing program.
F3 DEL
Delete step.
DEL (000>000) ?
Delete selected steps. Set start and end numbers of steps to be deleted.
F2 prog
Specify program number .
Specify a program number to be edited (identified).
F1 CLR
Delete program.
CLR (000>000) ?
Delete selected steps. Set start and end numbers of steps to be deleted.
ALL
Delete all program.
F2 CPY
Copy program.
CPY (000/000) ?
Specify number of source program and copy it to specified program number.
F4 etc
F1 SAV
Saving program.
Save all programs.
F2 LOD
Program readout.
Read out all program.

Figure 15-13: List of programming menu

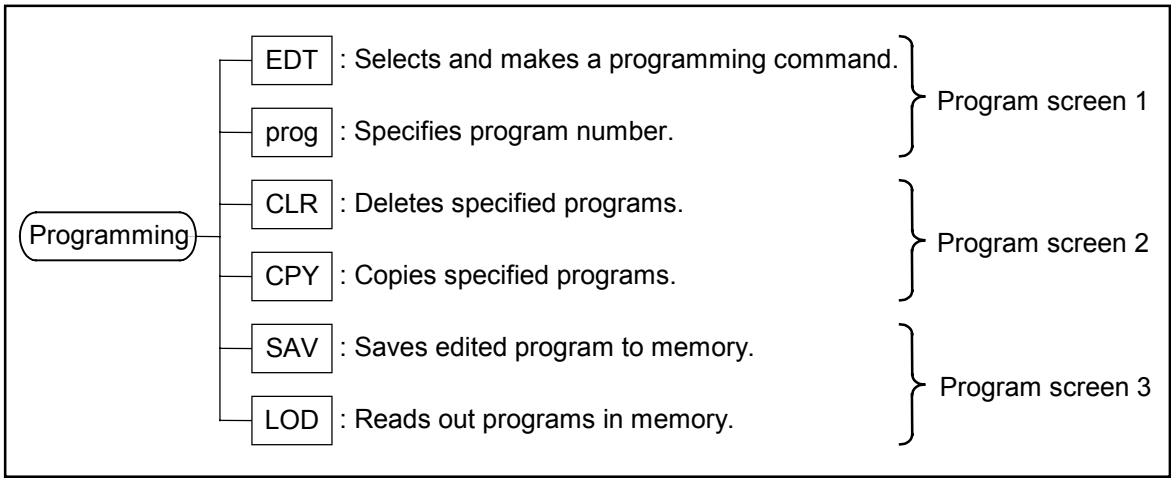
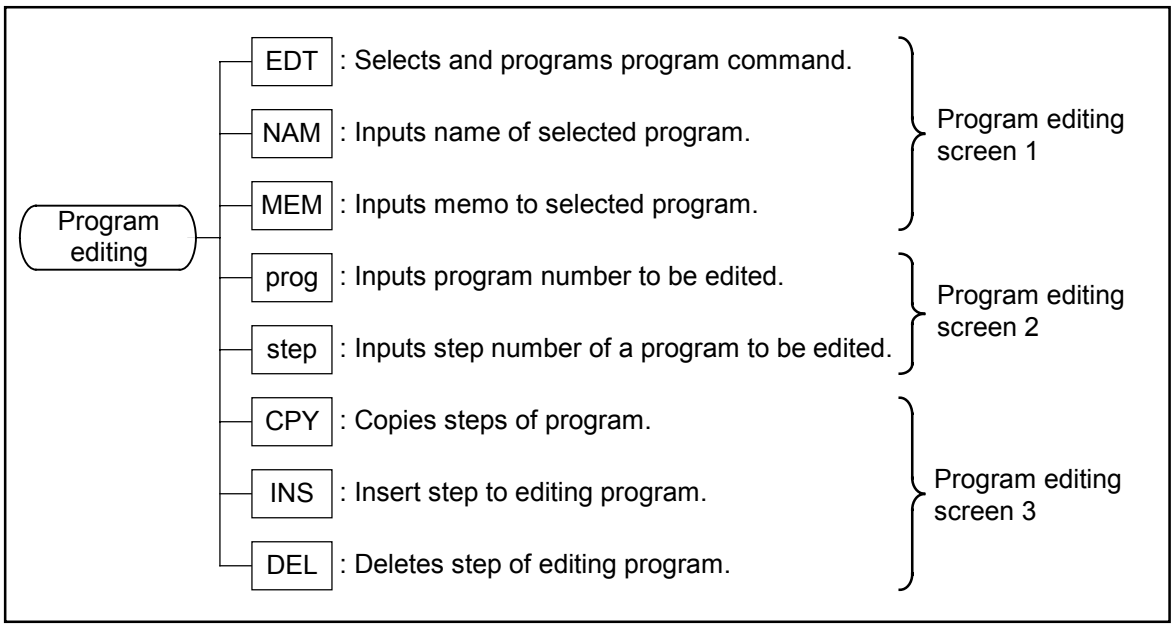


Figure 15-14: List of editing program menu



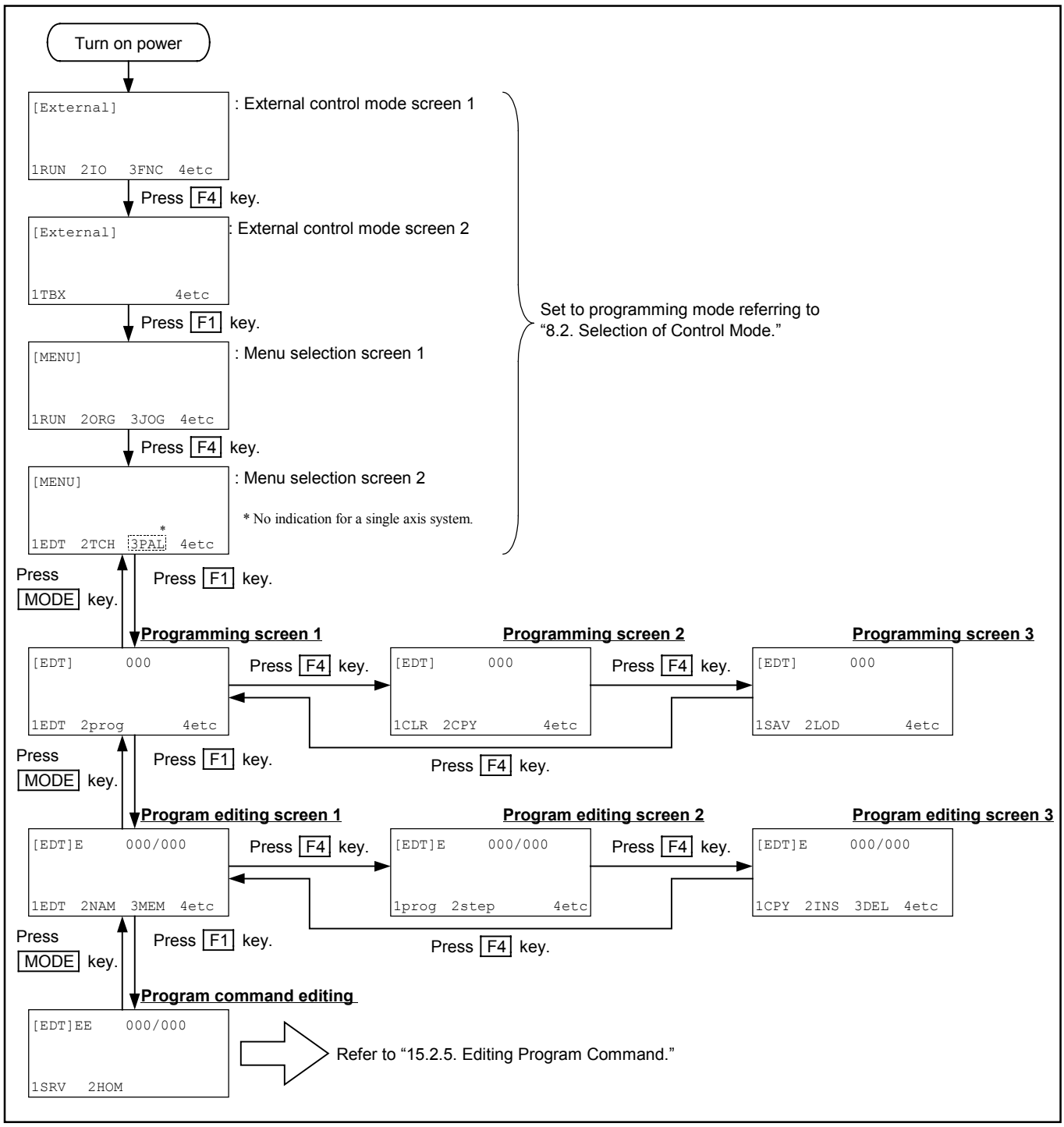
15.2.3. Programming Procedure

- Repetitive selections and sets of program commands in accordance with required procedures will lead you to make a desired program.

- In the programming screens 1 to 3, there are functions described below.
    1) Selection of program number
    2) Deletion of program
    3) Copy of program
    4) Write program of RAM to flash memory.
    5) Load program from flash memory to RAM

- In the program editing screens 1 to 3, there are functions described below.
    1) Input name of program and memo.
    2) Change program number and steps to be edited.
    3) Copy, delete and insert of program steps.

Figure 15-15: Programming procedure (Summary)



15.2.3.1. Programming Screen

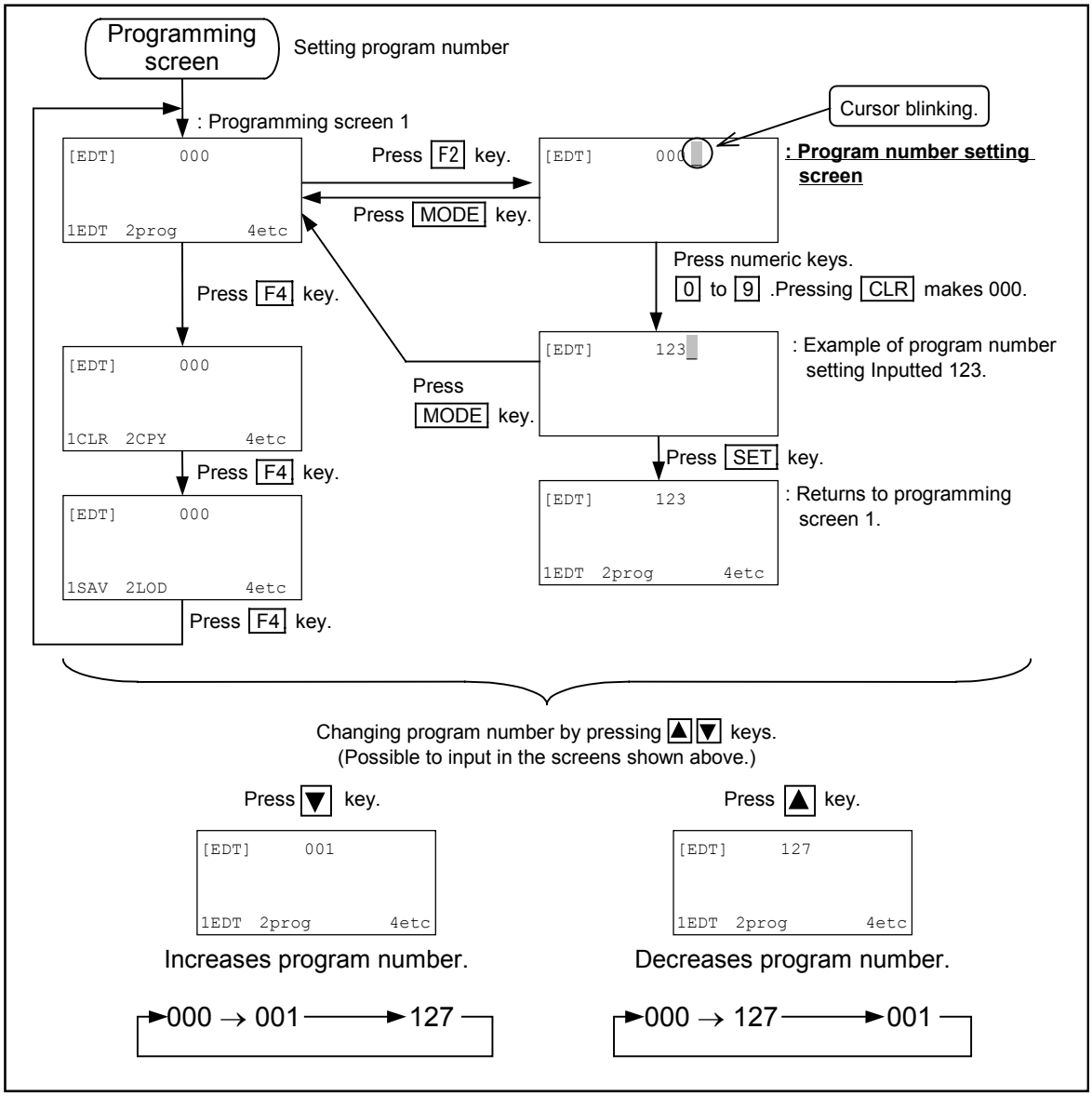
- In the programming screen 1 to 3, the functions described below can be carried out.
  1) Set number of program.
  2) Delete program data
  3) Copy programs
  4) Save program to flash memory.
  5) Read out programs from flash memory.

15.2.3.2. Setting Procedure for Program Number

Programming Screen — prog

- There are two ways to set a program number to be programmed.
  - ◊ Set the number in the program number setting screen that is selected using F2 key (prog) in the programming screen.
  - ◊ Use ▲ and ▼ keys.

Figure 15-16: Programming screen (Procedure for setting program number)

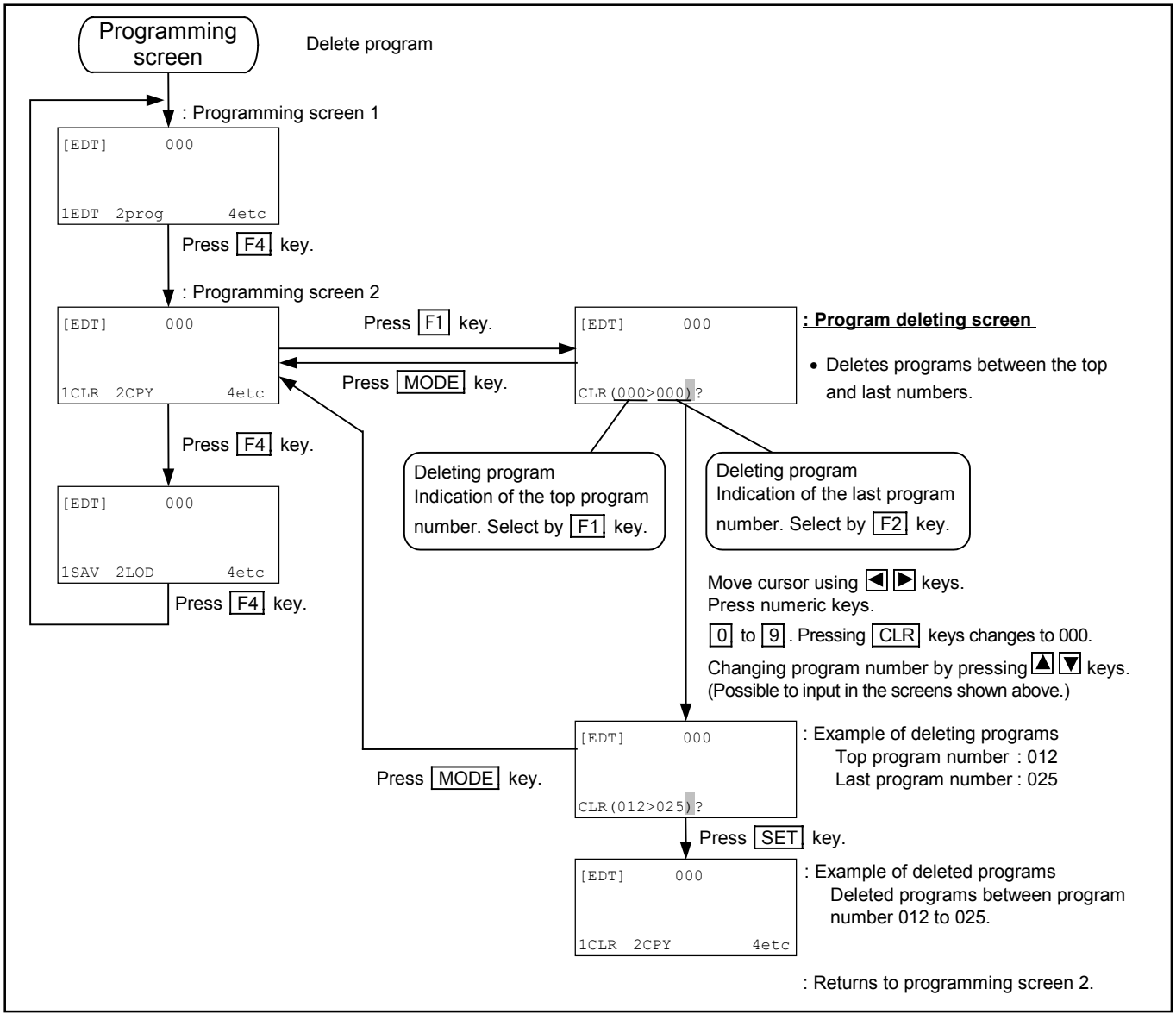


15.2.3.3. Procedure for Deleting Program

Programming Screen — CLR

- It can delete multiple programs together.
- Pressing MODE key during deleting process will cancel it. (Refer to Figure 15-17.)

Figure 15-17: Programming screen (Procedure for deleting program)

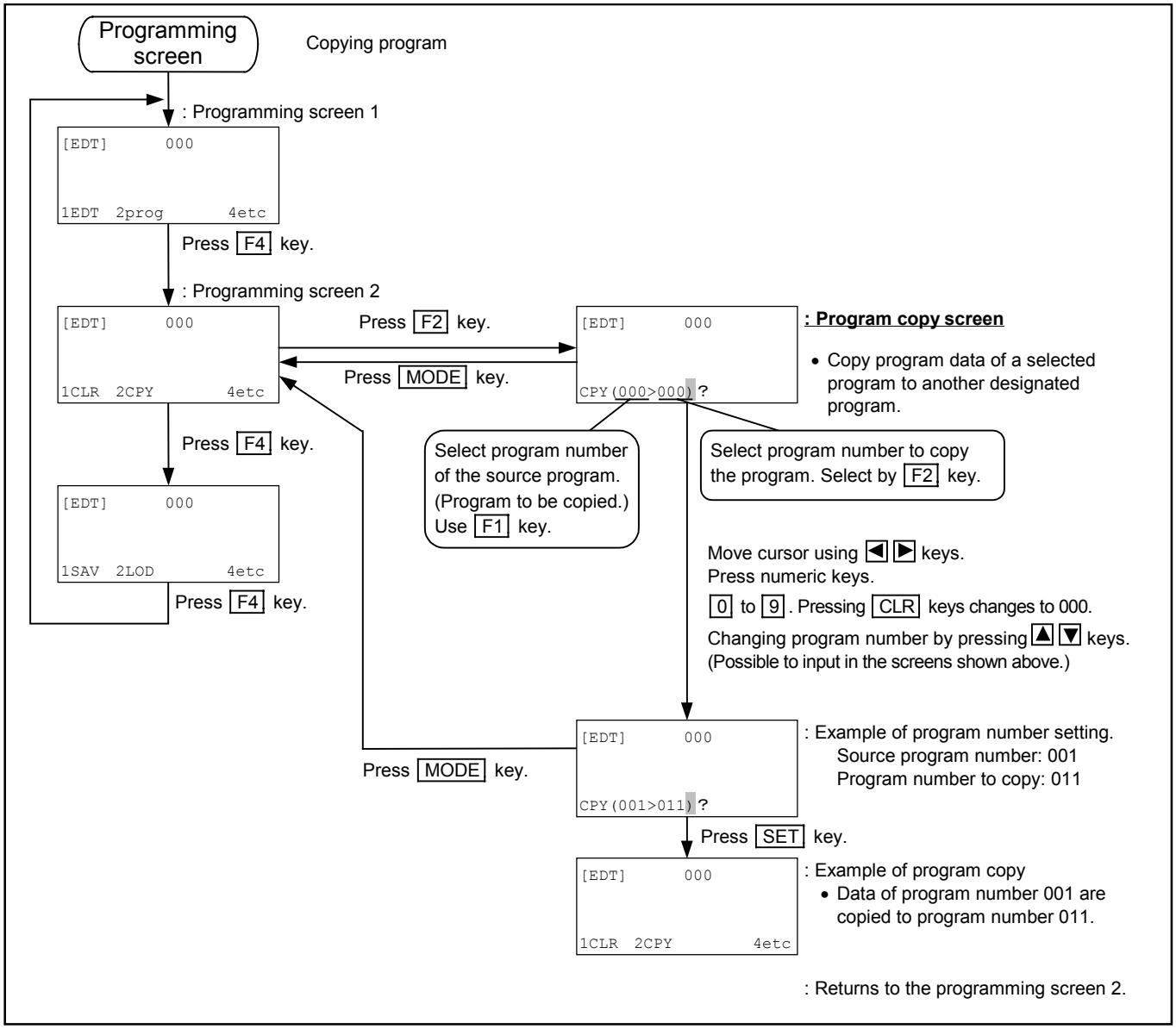


15.2.3.4. Procedure for Copying Program

Programming Screen — CPY

- Program data can be copied to other program (different program number).
- Pressing MODE key terminates to copy data. (Refer to Figure 15-18.)

Figure 15-18: Programming screen (Procedure for copying program)



15.2.3.5. Procedure for Saving Program

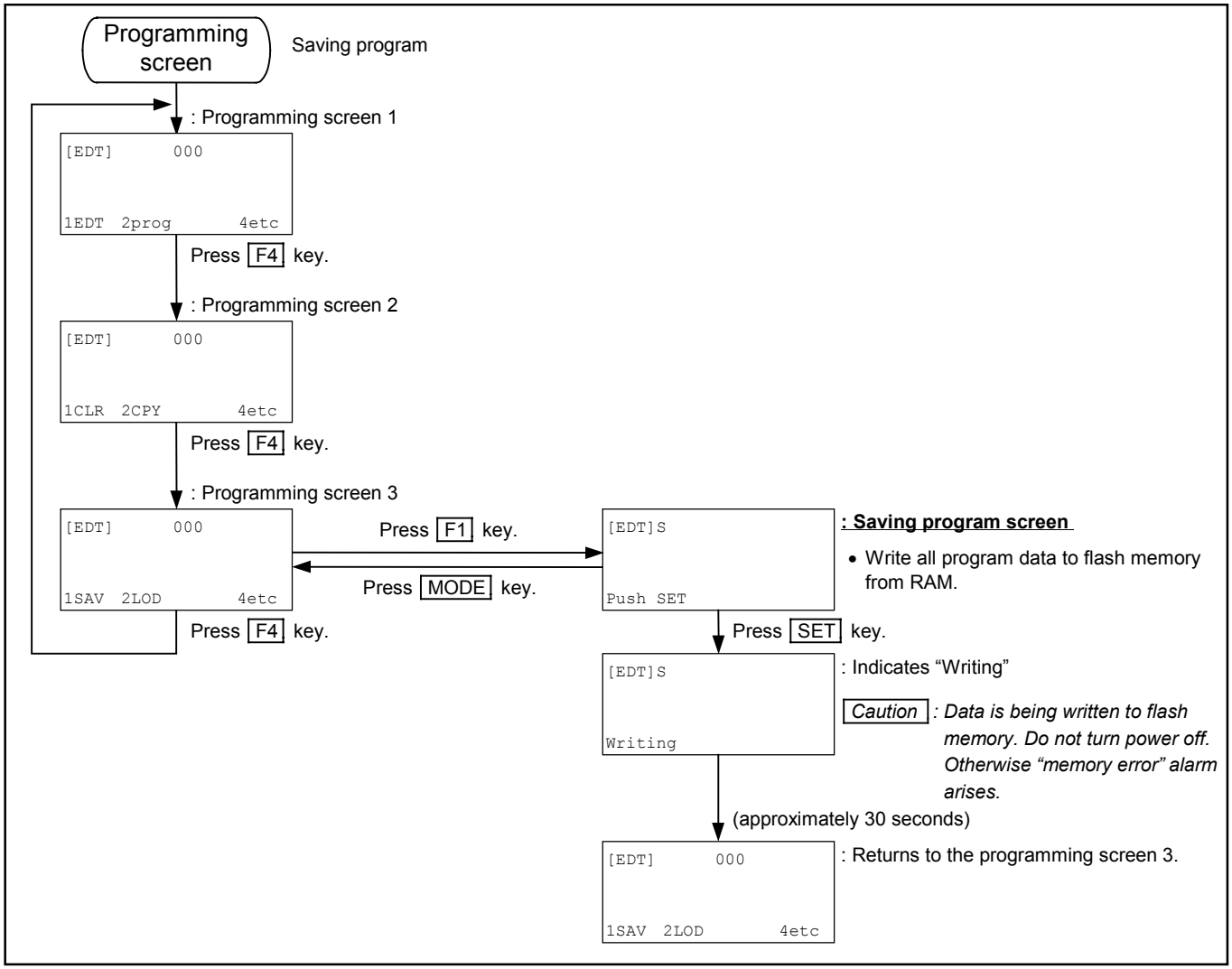
Programming Screen — SAV

- All program date shall be stored to the flash memory.

Caution : *Be sure to store new program or revised program immediately. New or revised program is lost if the power is turned off before store it.*

Caution : *"Writing" message appears on the screen while storing the program. Never turn off the power while storing. Otherwise Memory error alarm will be given.*

Figure 15-19: Programming screen (Procedure for saving program)

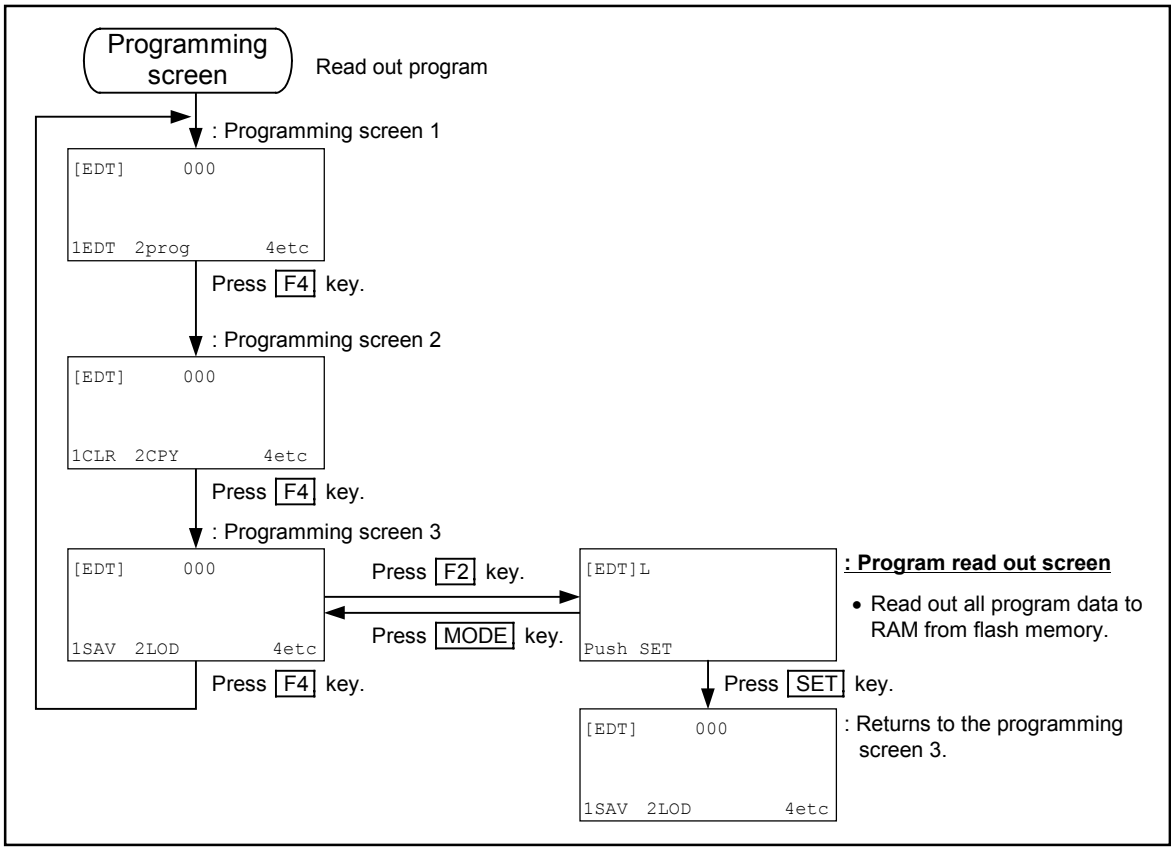


15.2.3.6. Procedure for Reading out Program

Programming screen — LOD

- Load program data to RAM from flash memory.

Figure 15-20: Programming screen (Procedure for reading out program)



15.2.4. Program Editing Screen

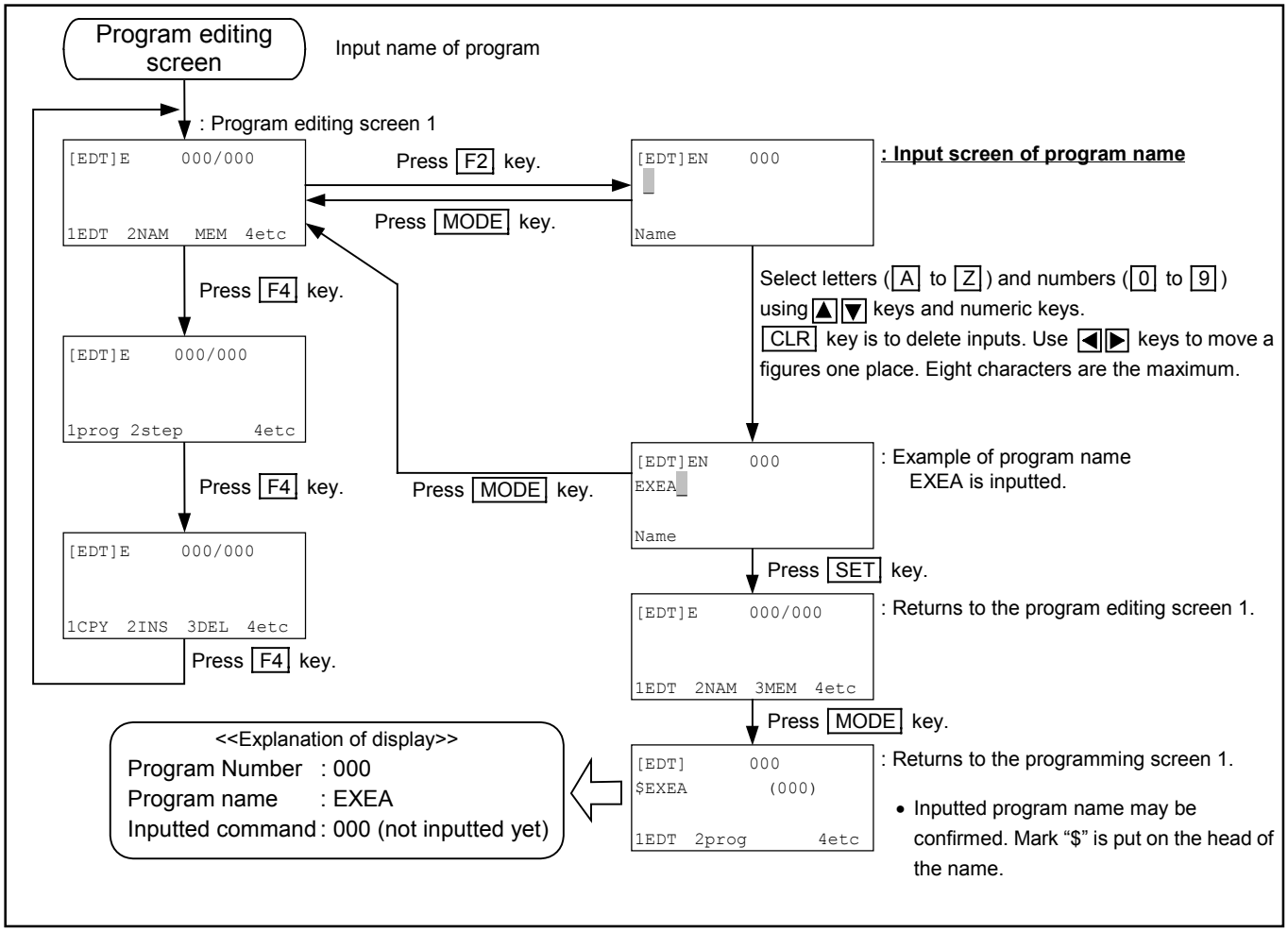
- In the program editing screen, entering name and memos of designated program number are possible. Copying, deleting and inserting of program command are possible as well.
- Program number and step number may be changed in this editing screen.

15.2.4.1. Procedure for Entering Name of Program

Program editing screen — NAM

- Significant digit of program name is 8 figures. This is used for calling a subroutine, changing the operation program and a multitask operation.
- Press [MODE] key to cancel the inputs.

Figure 15-21: Program editing screen (Procedure for entering name of program)

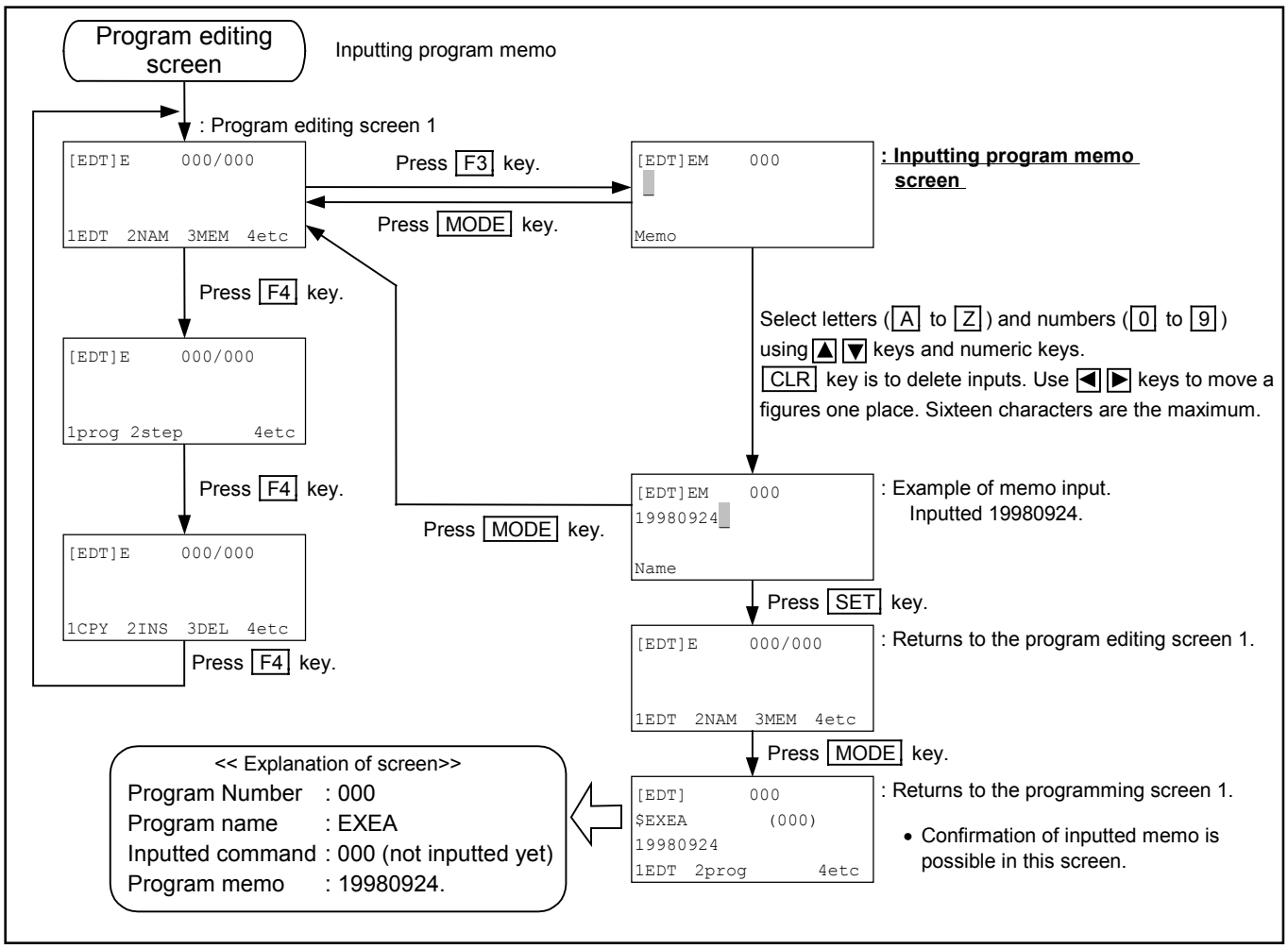


15.2.4.2. Inputting Memo to Program

Program editing screen — MEM

- Significant figure of the memo is 16 figures. Input of the memo does not affect a program. It may be used to note the date etc.
- Press MODE key to cancel an input.

Figure 15-22: Program editing screen (Procedure for inputting memo to program)

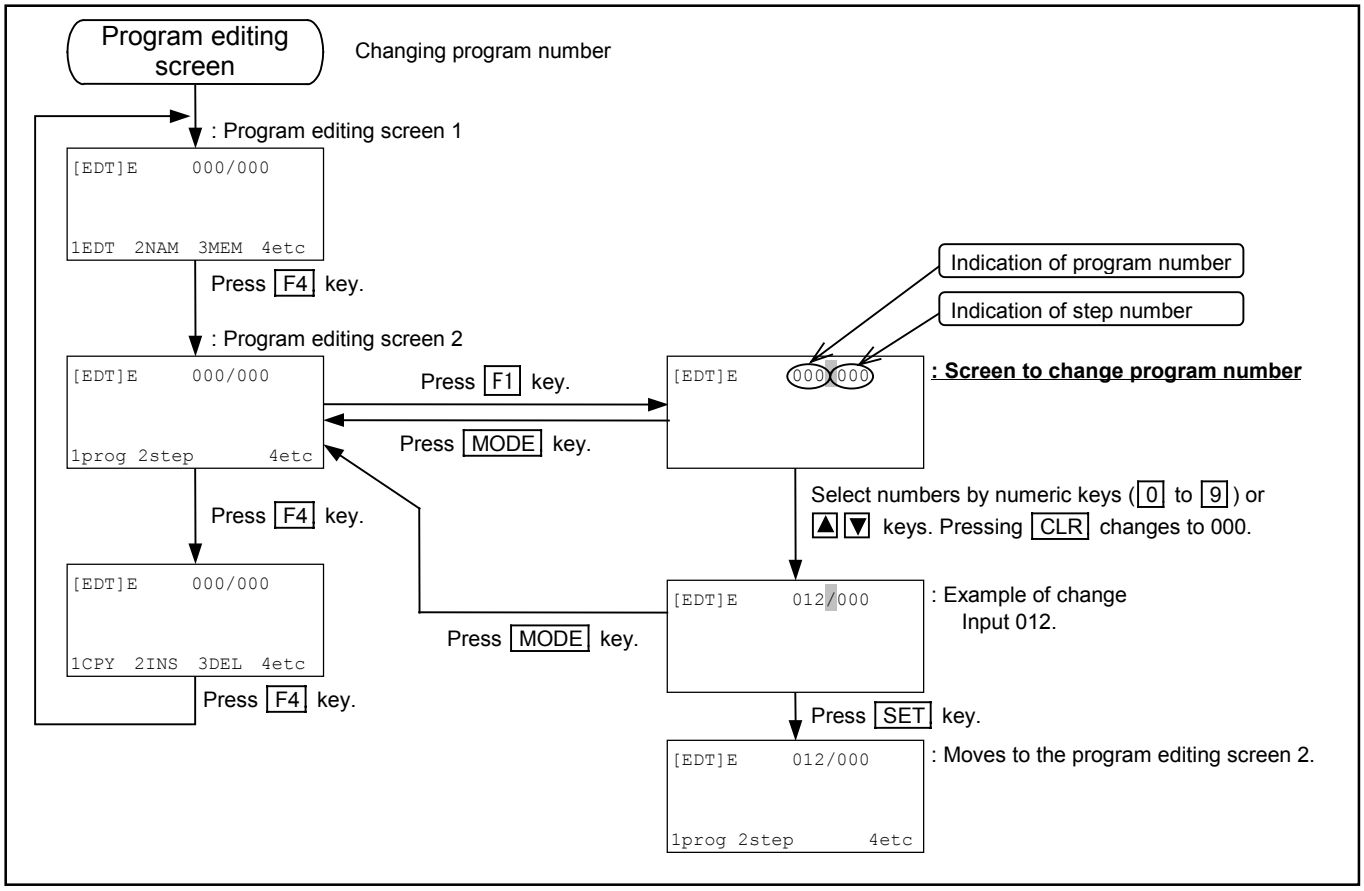


15.2.4.3. Procedure for Changing Program Number

Program editing screen — prog

- Number of program to be edited may be changed.
- Press MODE to cancel input.

Figure 15-23: Program editing screen (Procedure for changing program number)

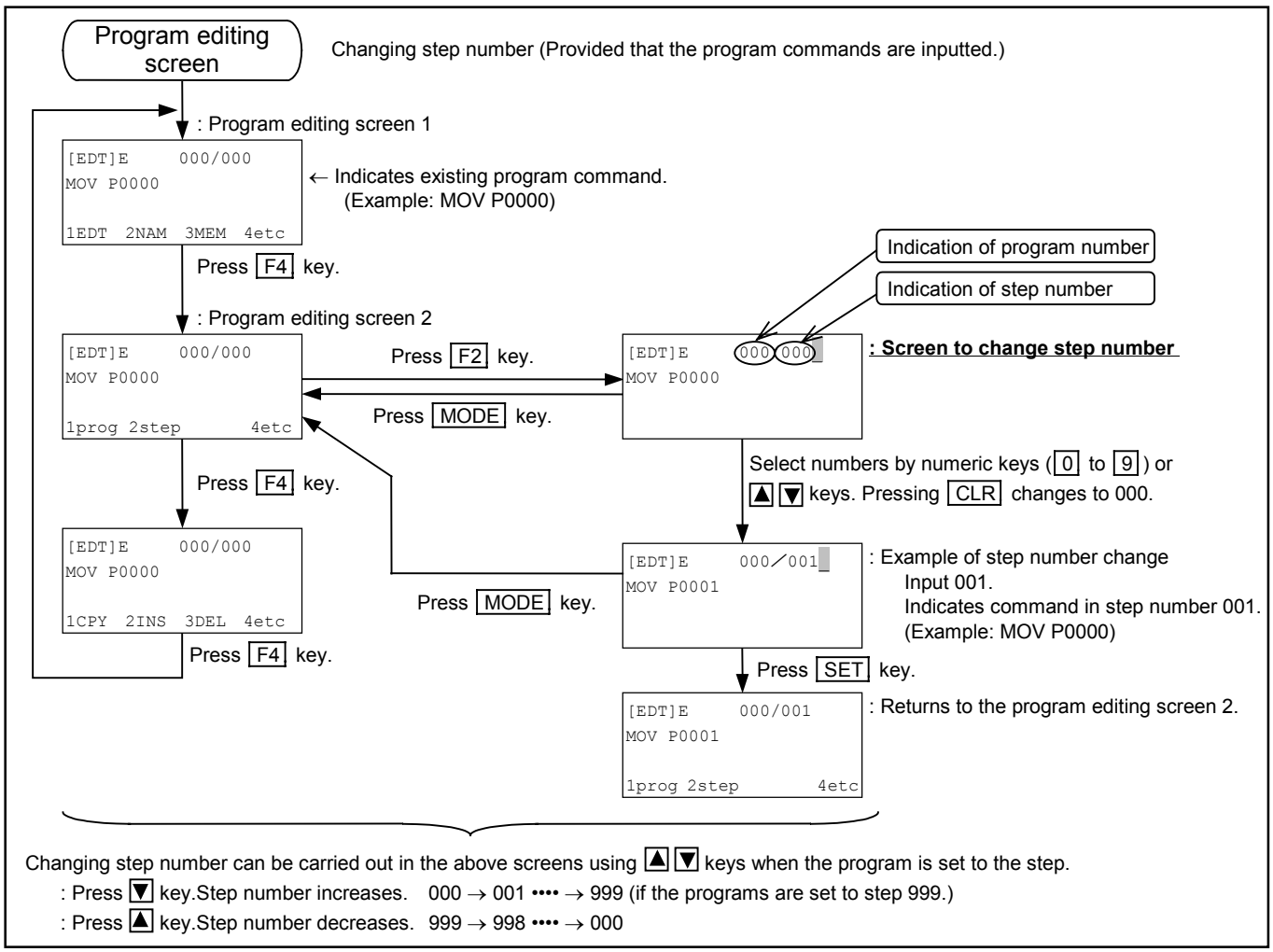


15.2.4.4. Procedure for Changing Step Number

Program editing screen — step

- You can change a step number of program to be edited. Changeable number range is from 000 to a number of step to which a program is set.
- Press MODE to cancel input.

Figure 15-24: Program editing screen (Procedure for changing step number)

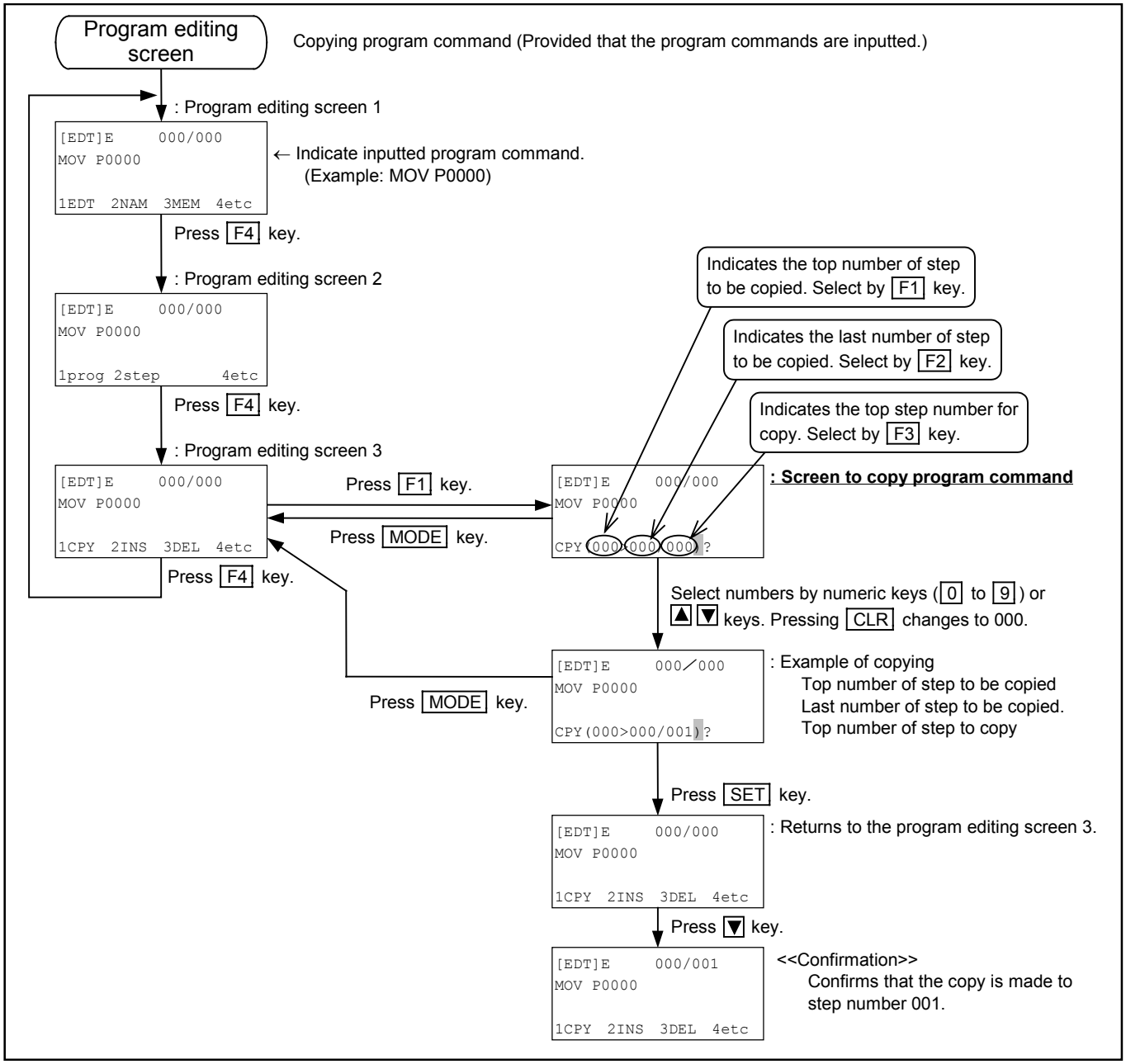


15.2.4.5. Procedure for Copying Program Command

Program editing screen — CPY

- You can copy edited program partially to the another part of the same program.
- Press MODE to cancel inputs.

Figure 15-25: Program editing screen (Procedure for copying program command)

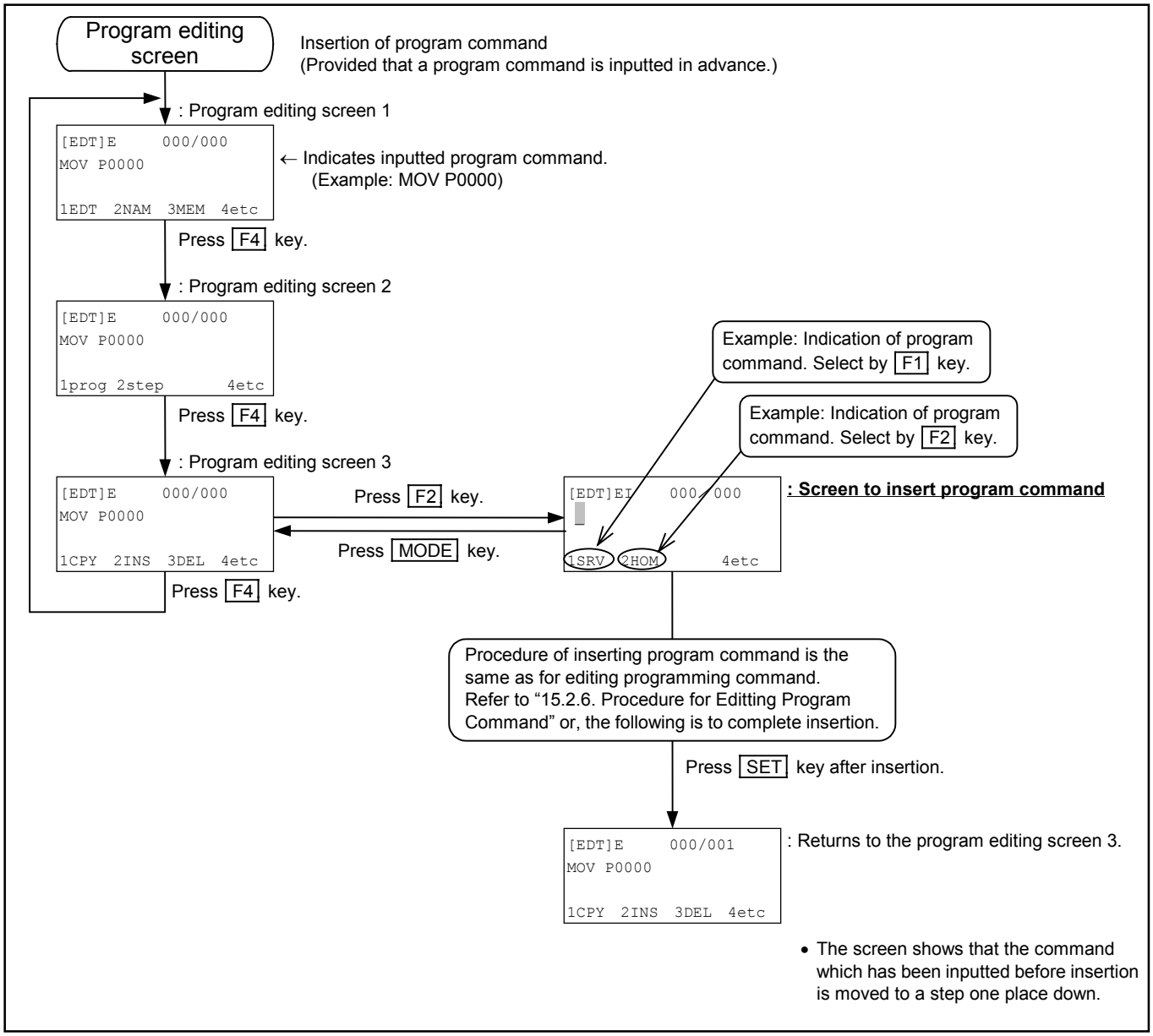


15.2.4.6. Procedure for Inserting Program Command

Program editing screen — INS

- This is to insert a command to an edited program.
- Press MODE key for cancellation of input or termination of insertion.

Figure 15-26: Program editing screen (Procedure for inserting program command)

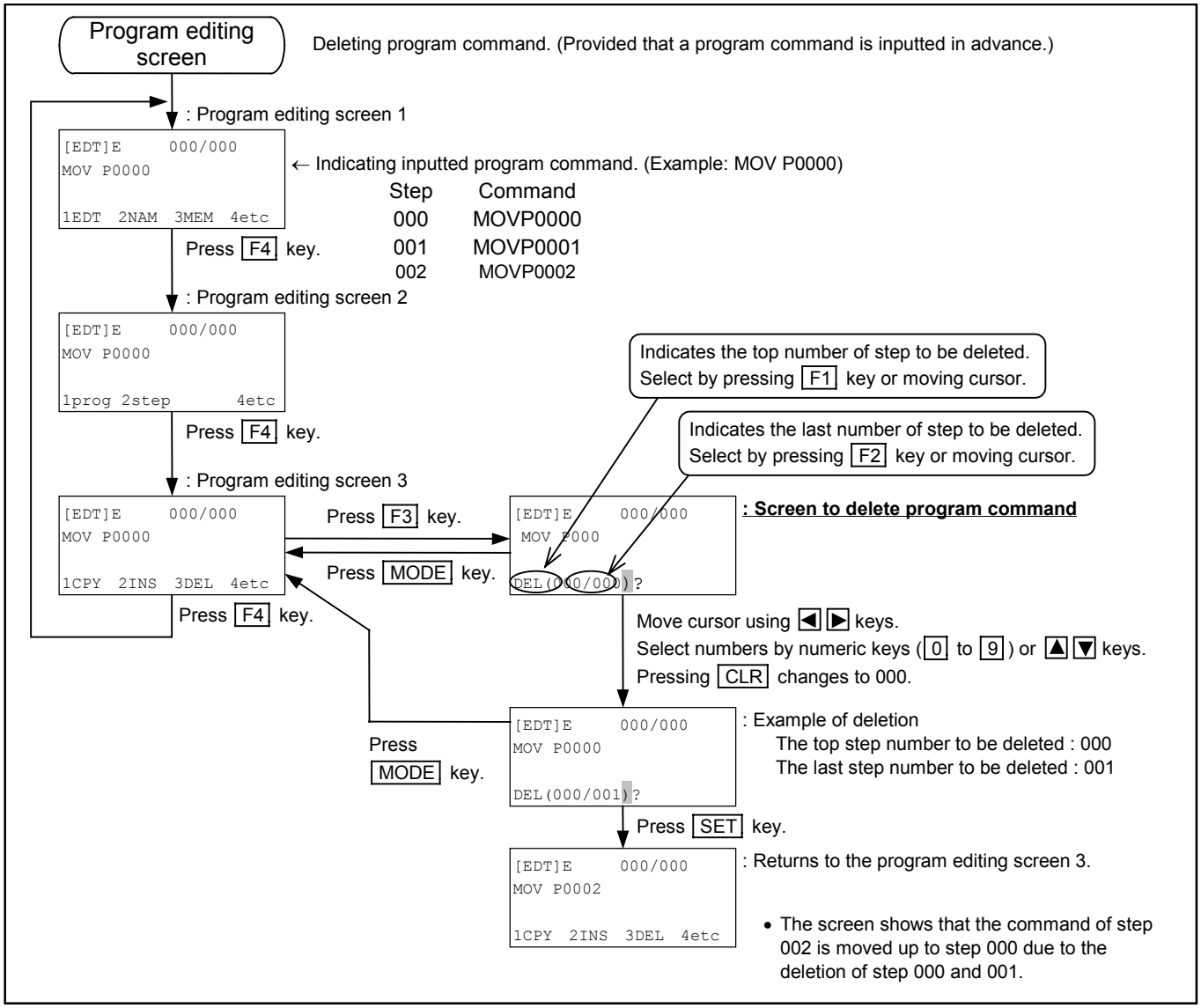


15.2.4.7. Procedure for Deleting Program Command

Program editing screen — DEL

- This is to delete an edited program command partially.
- Press MODE key to cancel input.

Figure 15-27: Program editing screen (Procedure for deleting program command)



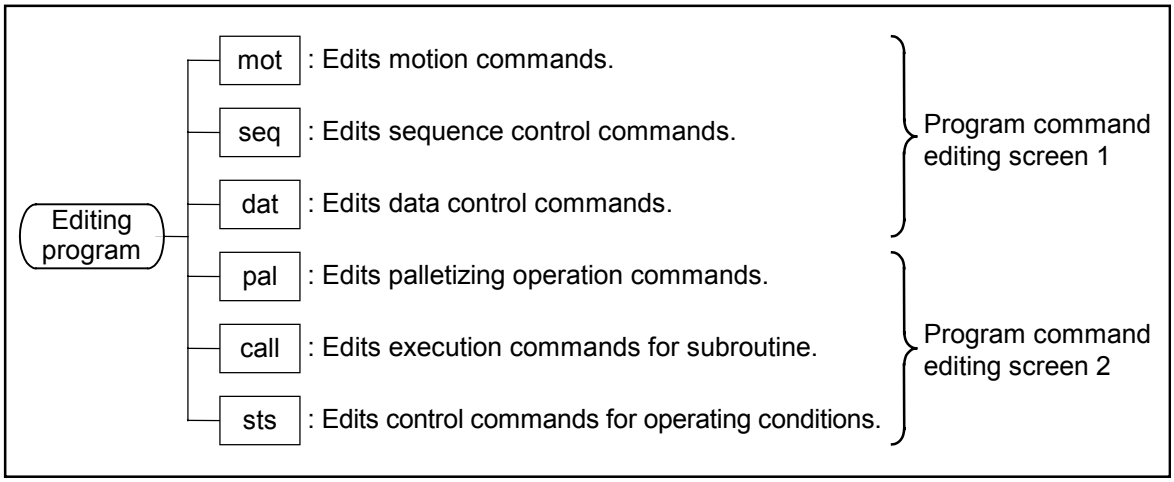
15.2.5. Editing Program Command

- This section describes editing program command.
- Press MODE key to cancel input.

15.2.5.1. Editing / Setting of Program Command

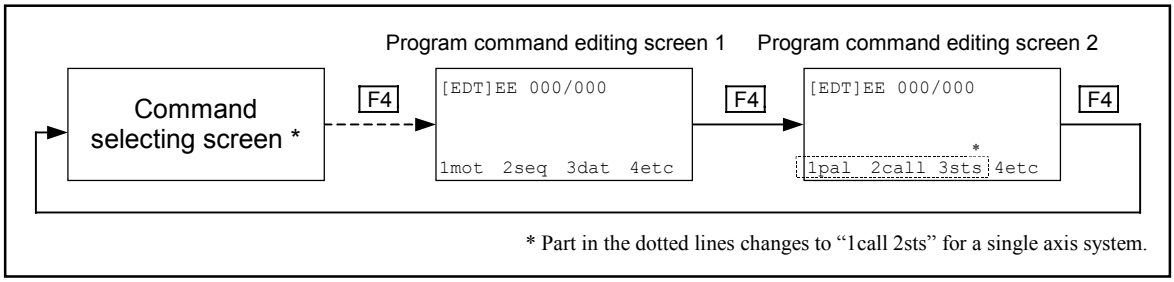
- Program commands consist of six groups (five groups for a single axis system) of command and selected necessary data from sub-menus accompanied with respective commands.

Figure 15-28: Editing menu list of program command



- F4 key scrolls the program command editing screen. In each program command editing screen, pressing F1, F2 and F3 keys selects the program command which is designated to each key.

Figure 15-29: Flow of editing screen for program command

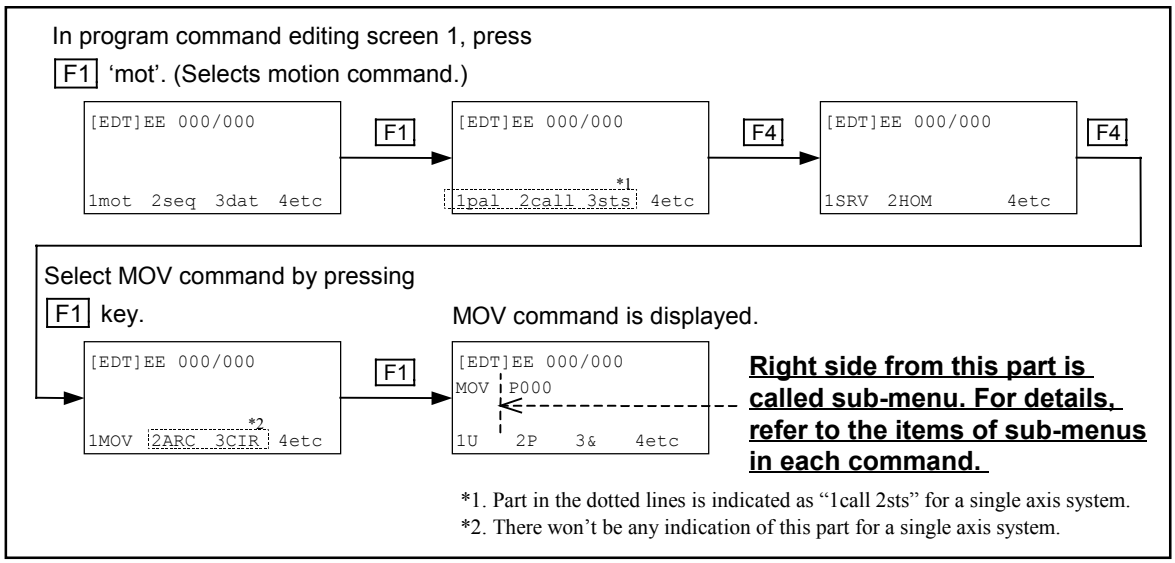


* In the Command selecting screen, the screen which was selected the last time appears.

Motion command editing screen----------Figure 15-30
Sequence control command editing screen ----------Figure 15-34
Data control command editing screen ----------Figure 15-37
Palletizing motion command editing screen----------Figure 15-40
(Note) Not available for a single axis system.
Subroutine execution command editing screen ----------Figure 15-43
Operating condition control command editing screen------Figure 15-46

(Motion command editing screen is selected for the screen just after the power is turned on.)

Figure 15-30: About sub-menu accompanied with respective commands



Note: Even though the sub-menus are common in each editing screen, the sub-menus you can select in each command screen differs. Refer to the syntax of respective descriptions in "15.2.7. Description of Program Command" for a sub-menu that is possible to select. Select a sub-menu that is indicated in a syntax as the screen shows other sub-menus that are not possible to select simultaneously. Otherwise an alarm will arise.

15.2.5.2. List of Program Command

- The program command listed in Table 15-2 are provided for EXEA controller.

Table 15-2: Program command list

Command group		Code	Function
mot	Motion command	SRV	Servo on/off
		HOM	Home return
		MOV	Linear interpolation **
		ARC *	Circular arc interpolation
		CIR *	Circular interpolation
		MSTP	Motion stop
		MEND	Wait for end of motion
		MSTS	Check for motion condition
		MOVM	Continuous linear interpolation
		CPS *	Start continue path ** (multi-point)
		CPE *	Complete continue path
seq	Sequence control command	'	Set comment line
		END	End of program
		TAG	Set label
		CMP	Compare data (with jump)
		JMP	Unconditional jump
		JEQ	Conditional jump (=)
		JGE	Conditional jump (≥)
		JLE	Conditional jump (≤)
		JNE	Conditional jump (≠)
		JGT	Conditional jump (>)
		JLT	Conditional jump (<)
		TIM	Set timer
		WAIT	Hold sequence
		REP	Repetition set
		NXT	Repetition end
dat	Data control command	LD	Set data
		CAL	Calculate data
		TCH	Set current data
		OUT	I/O output
		INP	I/O input
		LCAL	Data logical operation
pal	Palletizing motion command	PALI *	Initialize palletizing
		PALL *	Call subroutine for pallet change
		PALE *	Call subroutine for pallet change
		PALM *	Move pallet position
		PALN *	Change pallet position number
		QSTS *	Conform palletizing condition
call	Subroutine command	CALL	Subroutine call, start
		RET	Subroutine call, end
		RSTA	Initialize resuming operation setting
		CHG	Switch operation program
		CHLD	Start sub-sequence execution
		ENDC	End sub-sequence execution
sts	Edit operational condition command	UNT	Set moving unit (Do not use in a single axis system.)
		SPD	Set motion speed and acceleration
		TYP	Set motion format
		NOF	Set shift value of point register number
		PBS	Set position of working reference point
		ESCZ *	Set off-limits area of Z axis
		LDS	Read out, system setting state

* Not available for a single axis system.
** Linear interpolation is converted to "linear motion" in a single axis system.

15.2.5.3. Editing Motion Command

Program command editing screen — mot

- Call motion command, then make and edit a program.

Figure 15-31: Editing screen list of motion command

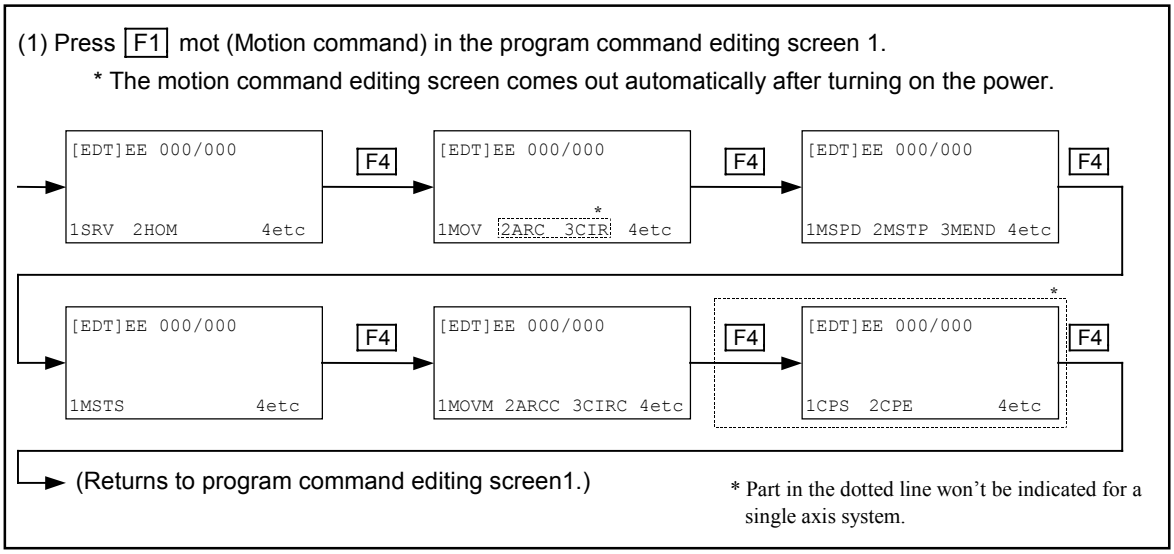


Figure 15-32: Editing menu list of motion control

Editing motion command. (Select by F1 mot.)

SRV	Servo on/off
HOM	Home return
MOV	Linear interpolation
ARC *	Circular arc interpolation (Defined by 3 points.)
CIR *	Circular interpolation (Defined 3 points.)
MSPD	(Do not set.)
MSTP	Stop motion
MEND	Wait for end of motion
MSTS	Check for motion status.
MOVM	Continuous linear interpolation
ARCC *	(Do not set.)
CIRC *	(Do not set.)
CPS *	Set continuous path motion (start)
CPE *	Set continuous path motion (end)

Select sub-menu of command setting. See Figure 15-33.

* Not available for a single axis system.

Figure 15-33: Sub-menu list of motion command editing

F4 etc

U	U@D	Unit number, specify direct/indirect
P	P@D	Point number, specify direct/indirect
&		Specify motion format
Xp	Xp@D	Point of X axis, specify direct/indirect
Xs	Xs@D	Condition of X axis, specify direct/indirect
X		Specify X axis
Yp *	Yp@D	Point of Y axis, specify direct/indirect
Ys *	Ys@D	Condition of Y axis, specify direct/indirect
Y *		Specify Y axis
Zp *	Zp@D	Point of Z axis, specify direct/indirect
Zs *	Zs@D	Condition of Z axis, specify direct/indirect
Z *		Specify Z axis
Rp *	Rp@D	Point of R axis, specify direct/indirect
Rs *	Rs@D	Condition of R axis, specify direct/indirect
R *		Specify R axis
S	S@D	Motion speed, specify direct/indirect
A	A@D	Motion acceleration, specify direct/indirect
B	B@D	Motion deceleration, specify direct/indirect
SR	SR@D	Motion speed, specify direct/indirect (in %)
AR	AR@D	Motion acceleration, specify direct/indirect (in %)
BR	BR@D	Motion deceleration, specify direct/indirect (in %)
D		Specify number of data register.
#		Numeric number setting (whole number)
=		Sign (equal sign)
OFF		Off
ON		On
ALL		Setting all axis / all unit
CUR		Set current position to Home position.
RSTA		Initialize re-start setting after power is turn on again.
ins		Insert sub-menu.
del		Delete one letter
chg		Select direct / indirect setting

* Not available for a single axis system.

15.2.5.4. Editing Sequence Control Command

Program command editing screen — seq

- Call all sequence control commands, then make and edit program.

Figure 15-34: Editing screen list of sequence control command

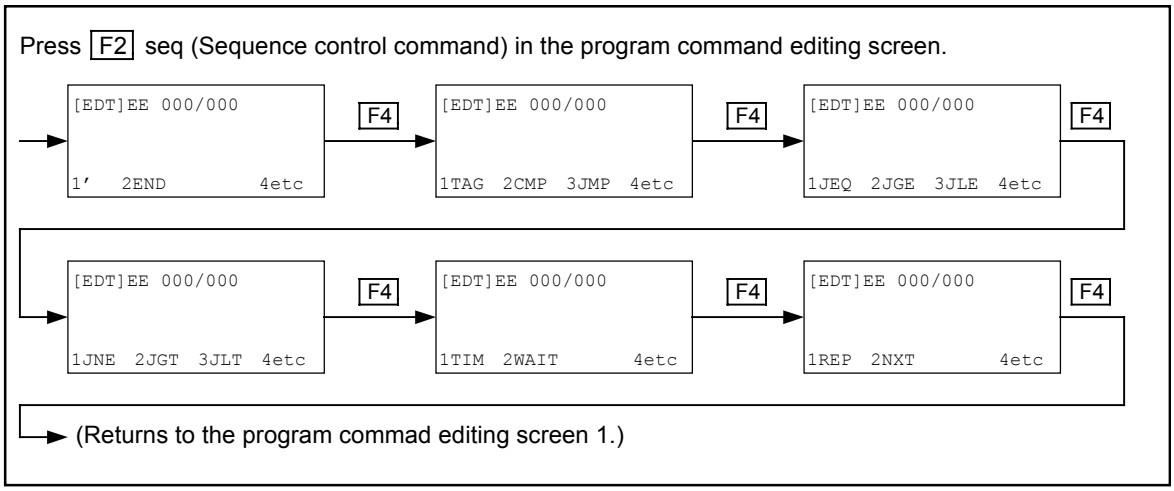


Figure 15-35: Editing menu list: Sequence control command

(2) Edit sequence control. (Select by F2 seq.)

'	Set comment line.
END	End of programmed operation
TAG	Set label
CMP	Compare data (with jump)
JMP	Jump
JEQ	Conditional jump (=)
JGE	Conditional jump (≥)
JLE	Conditional jump (≤)
JNE	Conditional jump (≠)
JGT	Conditional jump (>)
JLT	Conditional jump (<)
TIM	Timer setting
WAIT	Hold sequence
REP	Repetition (start)
NXT	Repetition (end)

Select sub-menu for command setting. See Figure 15-36.

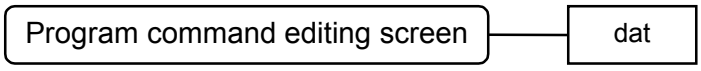
Figure 15-36: Sub-menu list: Editing sequence control command

F4 etc

_		Specify tag number
D		Specify data register number
#		Setting numeric value (whole number)
#.#		Setting numeric value, (decimal)
P	P@D	Point number, specify direct / indirect
PX	PX@D	X axis point number, specify direct / indirect
PY *	PY@D	Y axis point number, specify direct / indirect
PZ *	PZ@D	Z axis point number, specify direct / indirect
PR *	PR@D	R axis pint number, specify direct / indirect
IP	IP@D	Input port, specify direct / indirect
OP	OP@D	Output port, specify direct / indirect
MP	MP@D	Imaginary output, specify direct / indirect
IB	IB@D	Select input port by bit, specify direct / indirect
OB	OB@D	Select output port by bit, specify direct / indirect
MB	MB@D	Select imaginary input/output by bit, direct/indirect
; 8		I / O bit pattern, port
; 1		I / O bit pattern, bit
CSTP		Programmed operation, end
OFF		Off
ON		On
JEQ		Jump condition (equal)
JGE		Jump condition (over or equal)
JLE		Jump condition (less or equal)
JNE		Jump condition (not met)
JGT		Jump condition (over)
JLT		Jump condition (less)
EQ		Waiting condition (equal)
GE		Waiting condition (over or equal)
LE		Waiting condition (less or equal)
NE		Waiting condition (not met)
GT		Waiting condition (over)
LT		Waiting condition (less)
ins		Insert sub-menu.
del		Delete sub-menu.
chg		Select direct / indirect

* Not available for a single axis system.

15.2.5.5. Editing Data Control Command



- Call data control commands, then make and edit data.

Figure 15-37: Editing screen list: Data control

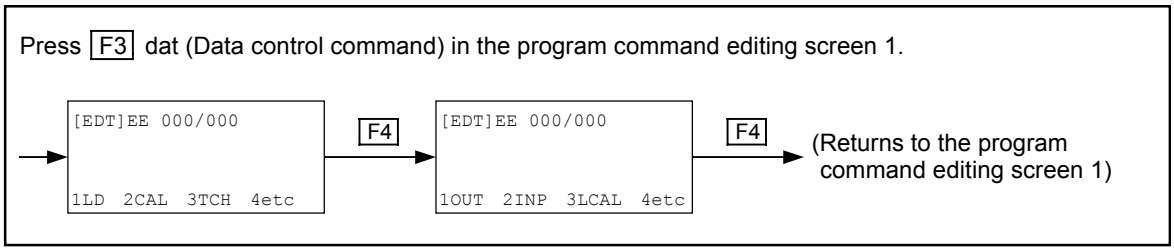


Figure 15-38: Editing menu list: Data control command

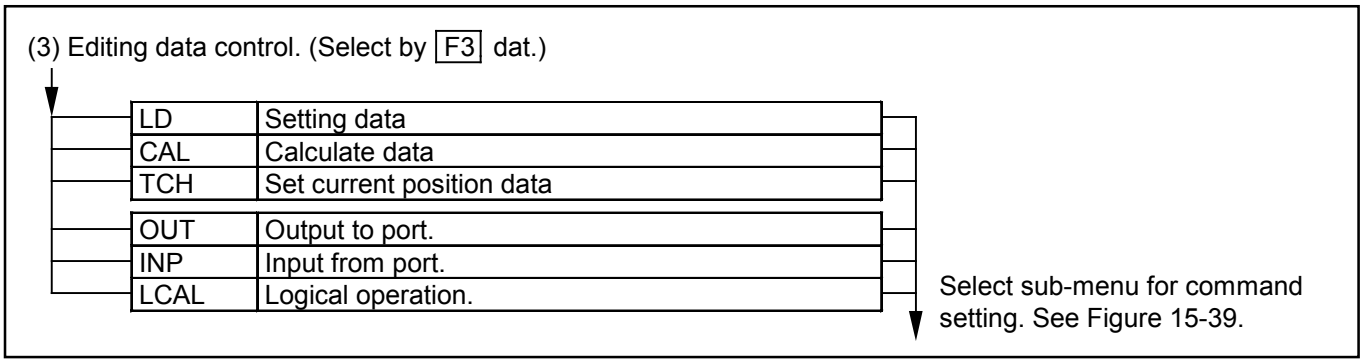


Figure 15-39: Sub-menu list: Editing data control command

F4 etc

D		Specify data register number
#		Setting numeric value (whole number)
#.#		Setting numeric value, (decimal)
U	U@D	Unit number, specify direct / indirect
P	P@D	Point number, specify direct / indirect
PX	PX@D	X axis point number, specify direct / indirect
PY *	PY@D	Y axis point number, specify direct / indirect
PZ *	PZ@D	Z axis point number, specify direct / indirect
PR *	PR@D	R axis point number, specify direct / indirect
Xp	Xp@D	X axis point, specify direct / indirect
Xs	Xs@D	Switch of X axis, specify direct / indirect
Yp *	Yp@D	Y axis point, specify direct / indirect
Ys *	Ys@D	Switch of Y axis, specify direct / indirect
Zp *	Zp@D	Z axis point, specify direct / indirect
Zs *	Zs@D	Switch of Z axis, specify direct / indirect
Rp *	Rp@D	R axis point, specify direct / indirect
Rs *	Rs@D	Switch of R axis, specify direct / indirect
IP	IP@D	Selection of Input port, specify direct / indirect
OP	OP@D	Selection of Output port, specify direct / indirect
MP	MP@D	Imaginary output, specify direct / indirect
IB	IB@D	Select input port by bit, specify direct / indirect
OB	OB@D	Select output port by bit, specify direct / indirect
MB	MB@D	Select imaginary input/output by bit, direct/indirect
; 8		Specify bit pattern of I / O port
; 1		Select bit pattern of I / O bit
=		Sign (equal)
+		Sign (addition)
-		Sign (subtraction)
*		Sign (multiplication)
/		Sign (division)
%		Sign (percent)
OFF		Off
ON		On
REV		Reverse output condition
OR		Logical sum
AND		Logical multiplication
XOR		Exclusive OR
RSTA		Return to former state / condition
ins		Insert sub-menu.
del		Delete sub-menu.
chg		Select direct / indirect

* Not available for a single axis system.

15.2.5.6. Editing Palletizing Motion Command (Multi-axis Combination only.)

Program command editing screen — pal

- Call palletizing motion command, then make and edit the program.

Figure 15-40: Editing screen list: Palletizing operation command

Press F1 pal (Palletizing motion command) in the program command editing screen 2.

[EDT]EE 000/000
1PALI 2PALL 3PALE 4etc
→ F4 →
[EDT]EE 000/000
1PALM 2PALN 3QSTS 4etc
→ F4 → (Returns the program command editing screen 1.)

Figure 15-41: Editing menu list: Palletizing operation command

(4) Edit palletizing operation command. (Select by F1 pal.)

PALI	Initialize palletizing operation.
PALL	Call subroutine, changing pallet
PALE	Call subroutine, changing pallet
PALM	Move pallet position
PALN	Change palletizing pattern number
QSTS	Confirm condition of the palletizing

Select sub-menu for command setting. See Figure 15-42.

Figure 15-42: Editing sub-menu list: Palletizing operation command

F4 etc

QN	QN@D	Palletizing operation number, specify direct / indirect
QP	QP@D	Palletizing pattern number, specify direct / indirect
U	U@D	Unit number, specify direct / indirect
P	P@D	Point number, specify direct / indirect
&		Specify motion format.
D		Specify data register number.
#		Numeric value (whole number)
_		Specify tag number.
$		Specify name of program or pallet.
S	S@D	Motion speed, specify direct / indirect
A	A@D	Motion acceleration, specify direct / indirect
B	B@D	Motion deceleration, specify direct / indirect
SR	SR@D	Motion speed (%), specify direct / indirect
AR	AR@D	Motion acceleration (%), specify direct / indirect
BR	BR@D	Motion deceleration (%), specify direct / indirect
=		Sign (equal)
QPM		Total number of points for pallet
QPC		Next point number of palletizing operation
ins		Insert sub-menu.
del		Delete one letter.
chg		Selection of direct /indirect.

15.2.5.7. Editing Subroutine Execution Command

Program command editing screen — call

- Call all command related to execution of subroutine, then make and edit program,

Figure 15-43: Editing screen list: Subroutine execution command

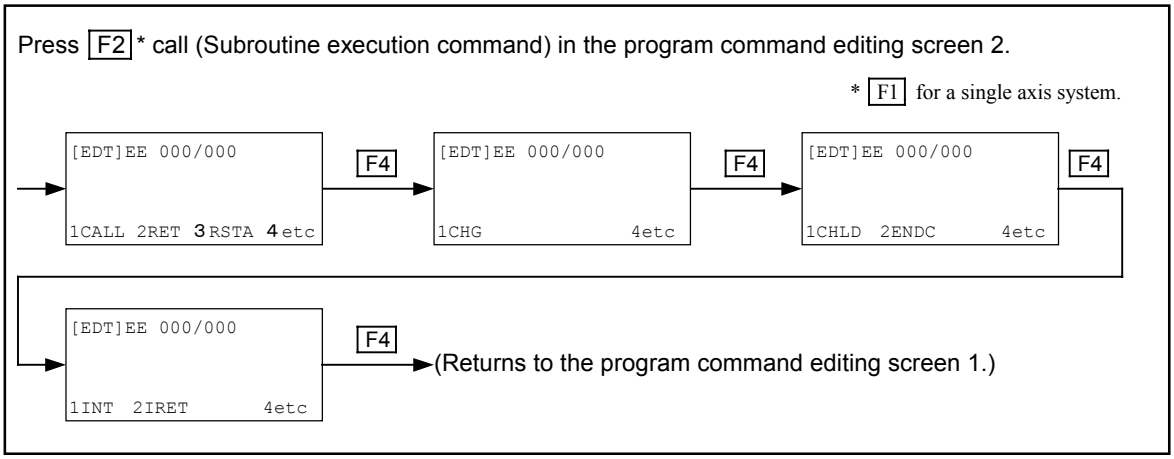


Figure 15-44: Editing menu list: Subroutine execution command

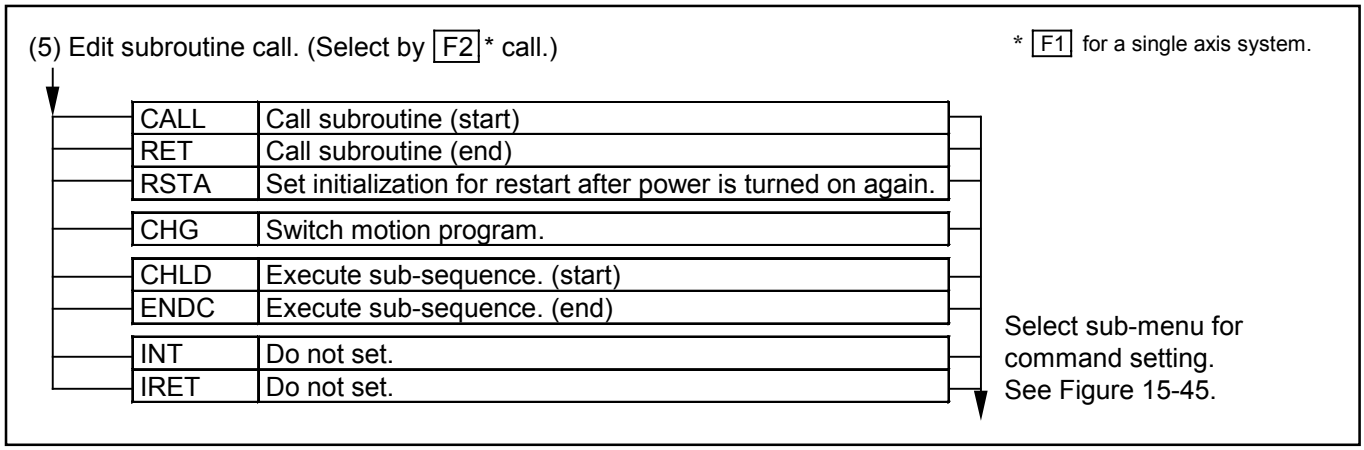


Figure 15-45: Sub-menu list: Editing subroutine execution command

F4 etc

D		Specify data register number
#		Setting numeric value (whole number)
#.#		Setting numeric value, (decimal)
P	P@D	Point number, specify direct / indirect
PX	PX@D	X axis point number, specify direct / indirect
PY *	PY@D	Y axis point number, specify direct / indirect
PZ *	PZ@D	Z axis point number, specify direct / indirect
PR *	PR@D	R axis pint number, specify direct / indirect
_		Specify tag number.
$		Specify name of program or pallet.
IP	IP@D	Input port, specify direct / indirect
OP	OP@D	Output port, specify direct / indirect
MP	MP@D	Imaginary output, specify direct / indirect
IB	IB@D	Select input port by bit, specify direct / indirect
OB	OB@D	Select output port by bit, specify direct / indirect
MB	MB@D	Select imaginary input/output by bit (every bit) direct/indirect
; 8		I / O bit pattern, port
; 1		I / O bit pattern, bit
OFF		Off
ON		On
CSTP		Cycle stop
DSTP		Decelerating stop
STP		Immediate stop
DIS		Prohibit interruption
ENA		Permit interruption
EQ		Interruption condition, (equal)
GE		Interruption condition, (more or equal)
LE		Interruption condition, (less or equal)
NE		Interruption condition, (non conformity)
GT		Interruption condition, (more)
LT		Interruption condition, (less)
ins		Insert sub-menu.
del		Delete sub-menu.
chg		Select indirect or indirect.

* Not available for a single axis system.

15.2.5.8. Command for Operating Condition Control

Program command editing screen — sts

- Sets or reads out operating conditions.

Figure 15-46: Editing screen list: Operating condition control command

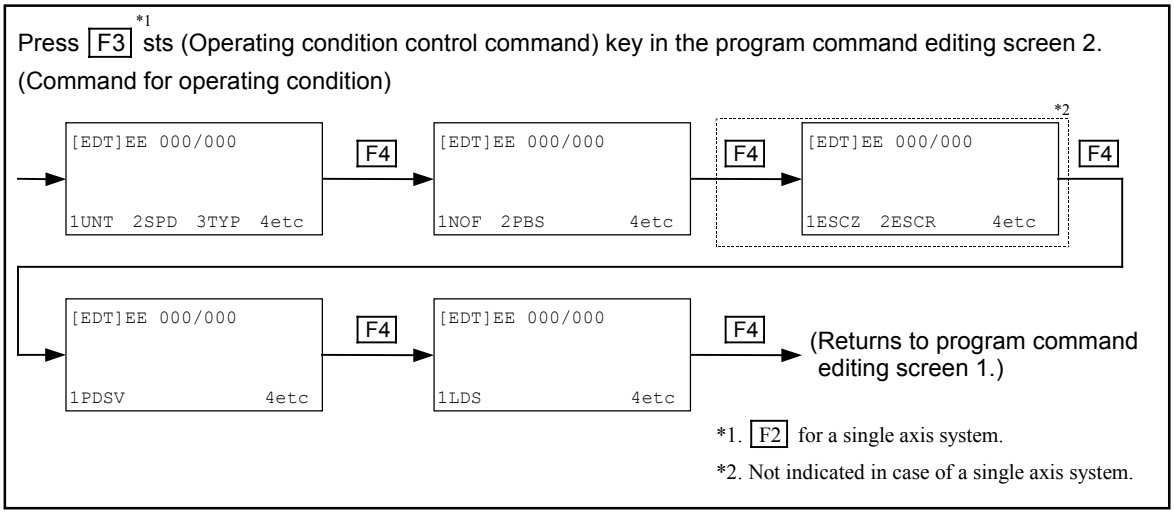


Figure 15-47: Command menu list: Operating condition control command

(6) Edit operating condition. (Select by F3 *1 sts key.)

UNT	Set operating unit.
SPD	Set motion speed and acceleration / deceleration.
TYP	Set motion format.
NOF	Set off-set value of point register number.
PBS	Position of working reference point
ESCZ *2	Off-limits area, Z axis motion
ESCR *2	Off-limits area, R axis motion (Do not set.)
PDSV	Save operating condition. (Reserved)
LDS	Read out motion setting data.

Select sub-menu for command of operating condition.
See Figure 15-48.

*1. F2 for a single axis system.

*2. Not indicated in case of a single axis system.

Figure 15-48: Sub-menu list: Command for operating condition control

F4 etc

D		Specify data register number.
#		Setting numeric value (whole number)
#.#		Setting numeric value (decimal)
U	U@D	Unit number, specify direct / indirect.
P	P@D	Point number, specify direct / indirect.
&		Specify motion format.
PX	PX@D	X axis point number, specify direct / indirect.
PY *	PY@D	Y axis point number, specify direct / indirect.
PZ *	PZ@D	Z axis point number, specify direct / indirect.
PR *	PR@D	R axis point number, specify direct / indirect.
Xp	Xp@D	X axis point, specify direct / indirect.
Yp *	Yp@D	Y axis point, specify direct / indirect.
Zp *	Zp@D	Z axis point, specify direct / indirect.
Rp *	Rp@D	R axis point, specify direct / indirect.
S	S@D	Motion speed, specify direct / indirect.
A	A@D	Motion acceleration, specify direct / indirect.
B	B@D	Motion deceleration, specify direct / indirect.
SR	SR@D	Motion speed (%), specify direct / indirect.
AR	AR@D	Motion acceleration (%), specify direct / indirect.
BR	BR@D	Motion deceleration (%), specify direct / indirect.
=		Sign (equal)
EMST		Save operating condition. (Emergency stop) (Reserved)
ALM		Save operating condition. (stop for alarm) (Reserved)
STP		Save operating condition. (immediate stop) (Reserved)
CSTP		Save operating condition. (cycle stop) (Reserved)
UNT		Setting value, motion unit
NOF		Off-set value, point register number
SPD		Set value, motion speed
ACC		Set value, motion acceleration
DAC		Set value, motion deceleration
SPDR		Set value (%), motion speed
ACCR		Set value (%), motion acceleration
DACR		Set value (%), motion deceleration
PBS		Set value, coodinate offset.
ESCZ *		Off-limits area, Z axis motion
ESCR *		Off-limits area, R axis motion (Do not set.)
UPR *		Set upper boundary.
LWR *		Set lower boundary.
POS *		Set value, turnout position
ins		Insert sub-menu.
del		Delete sub-menu.
chg		Select direct / indirect

* Not available for a single axis system.

15.2.5.9. Sub-menu

- Following figures describe the sub-menus of each command.

Figure 15-49: Sub-menu No.1

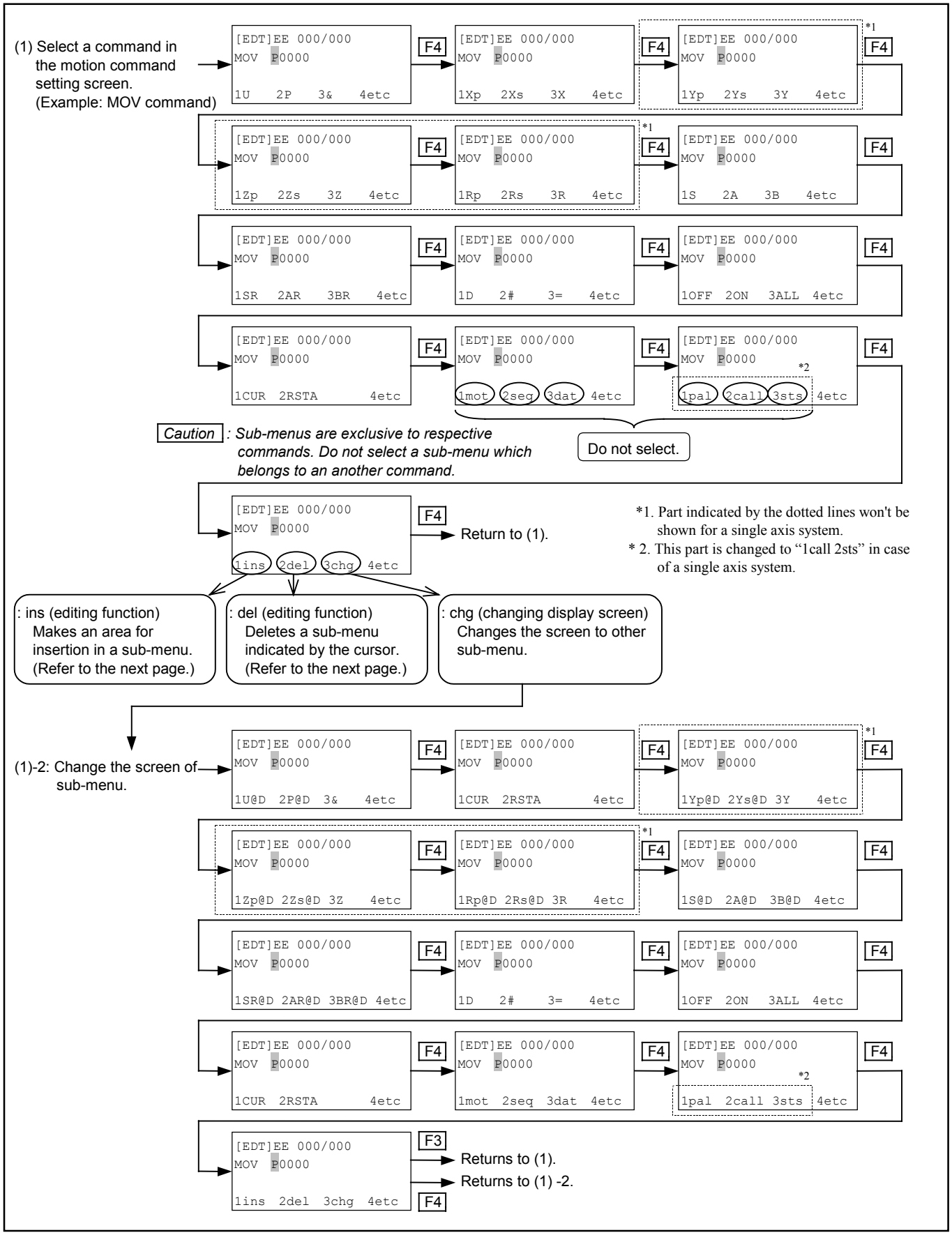


Figure 15-50: Sub-menu (No.2)

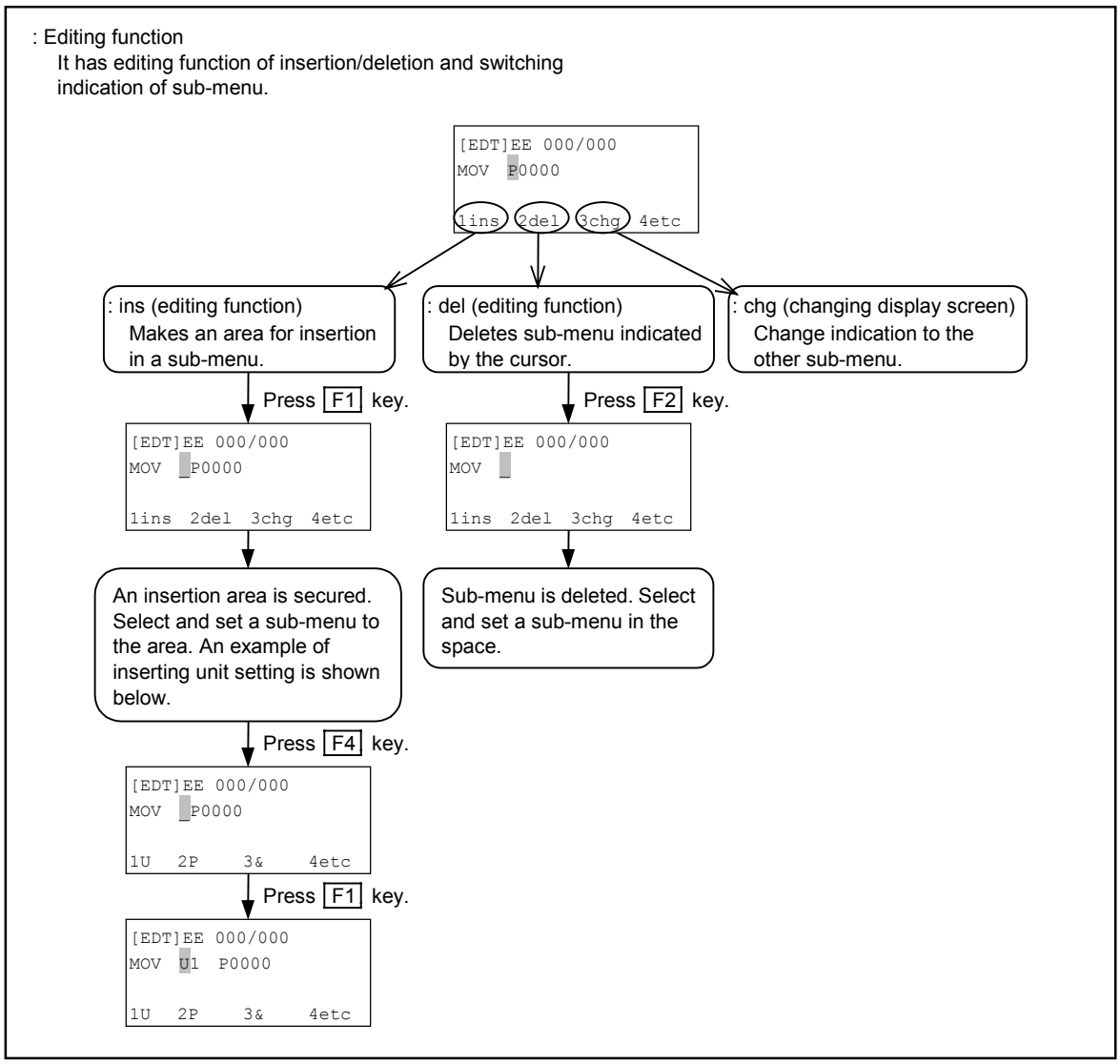


Figure 15-51: Sub-menu (No.3)

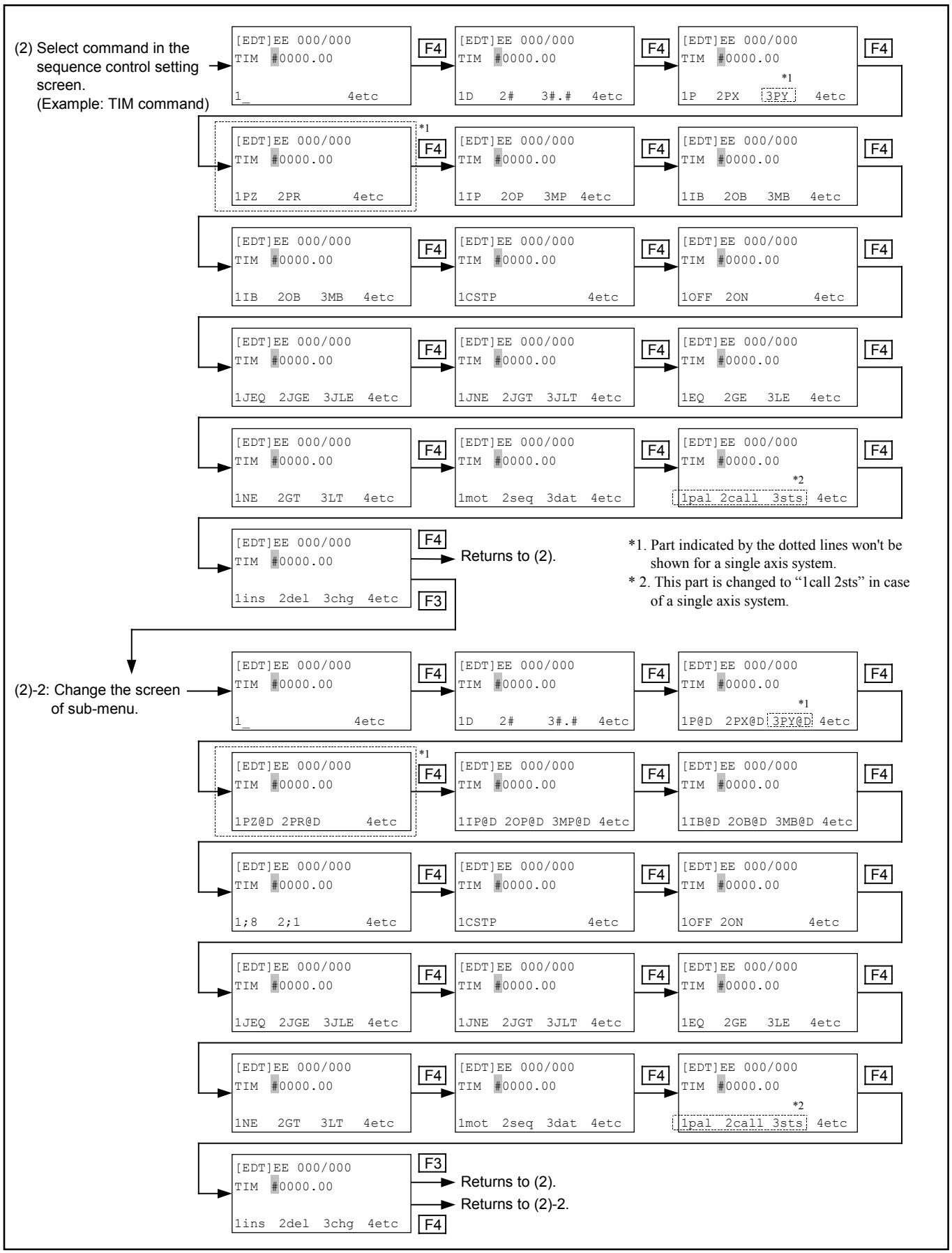


Figure 15-52: Sub-menu (No. 4)

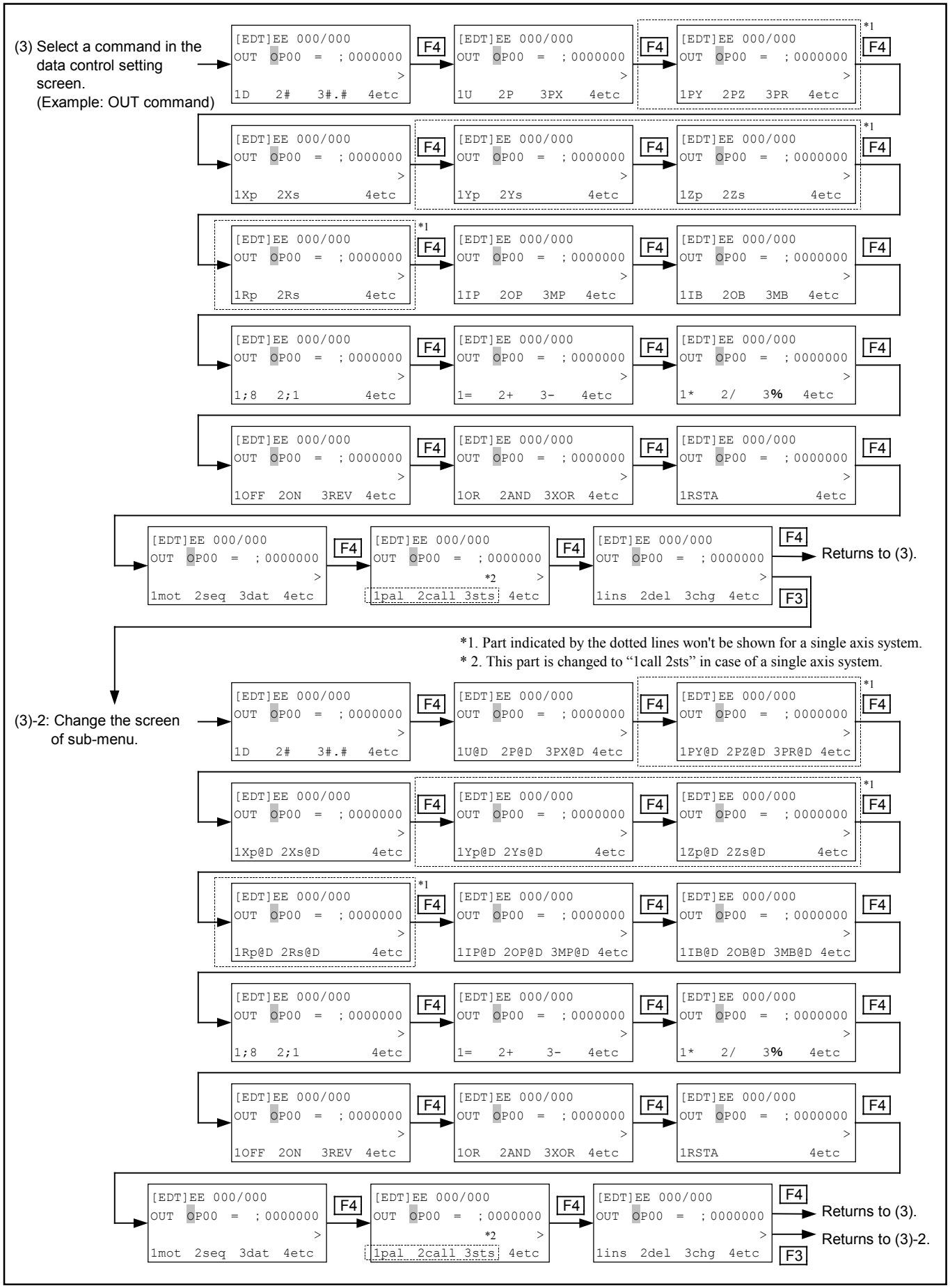


Figure 15-53: Sub-menu (No.5) (Mulit-axis combination only)

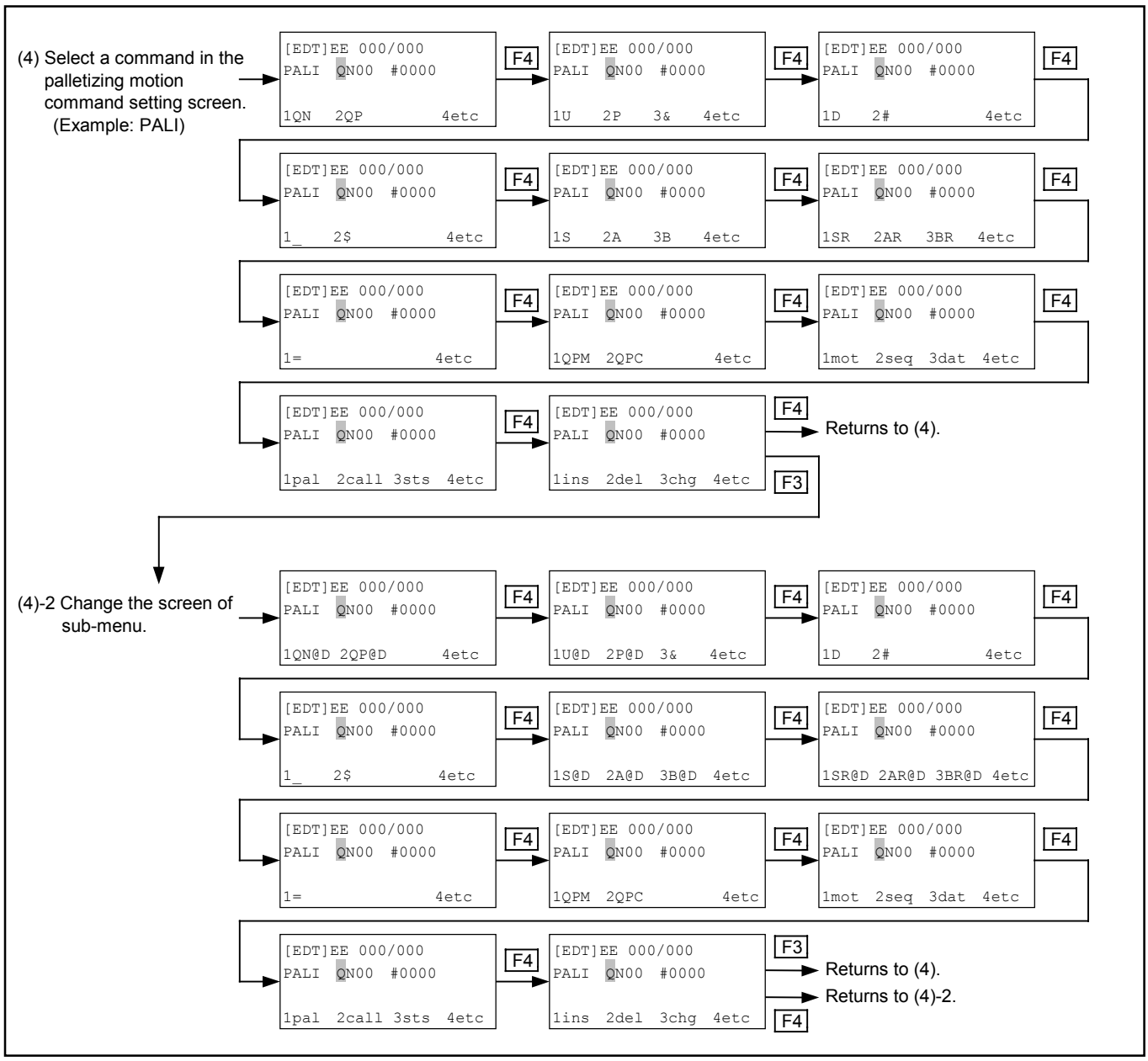


Figure 15-54: Sub-menu (No.6)

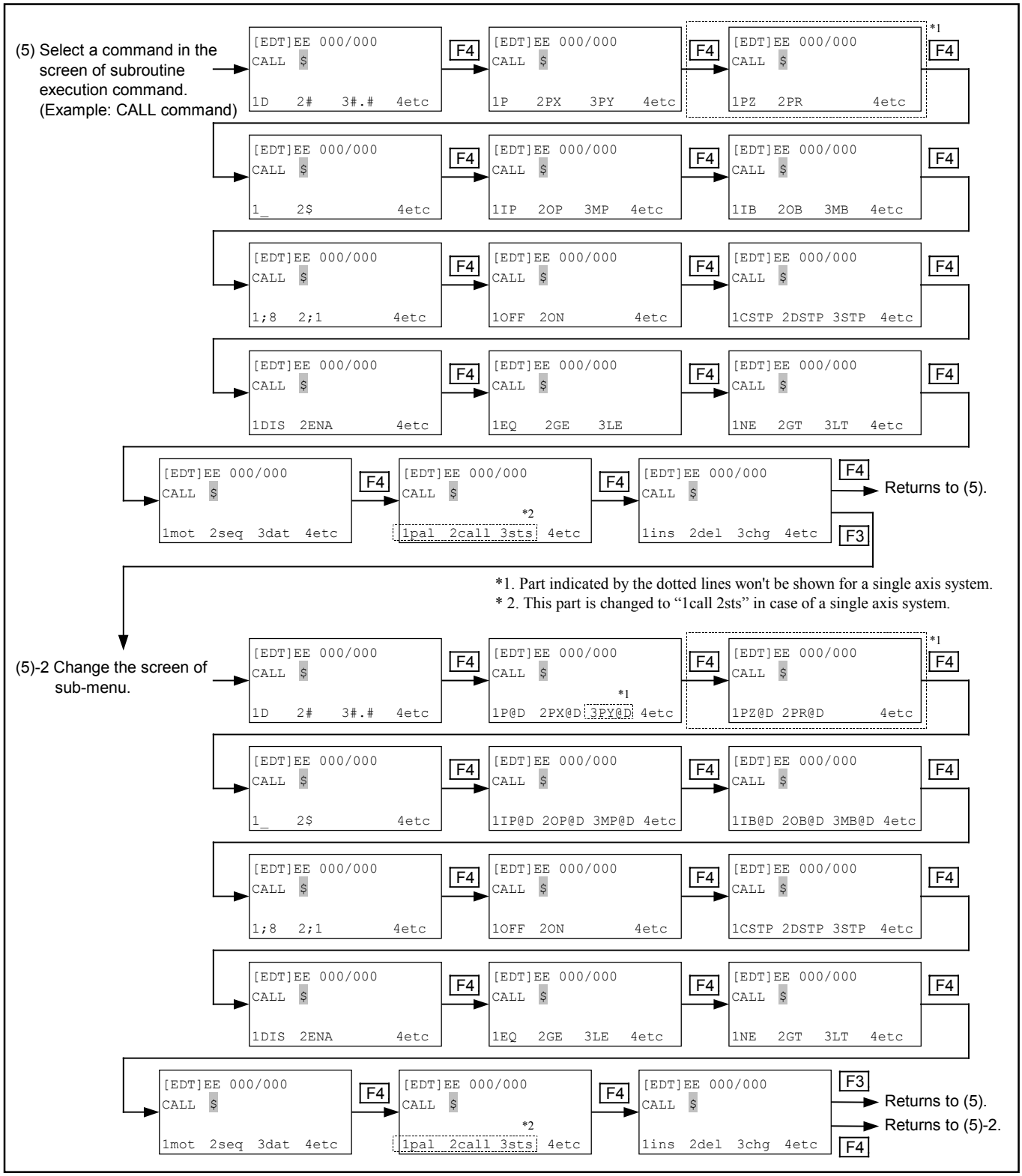
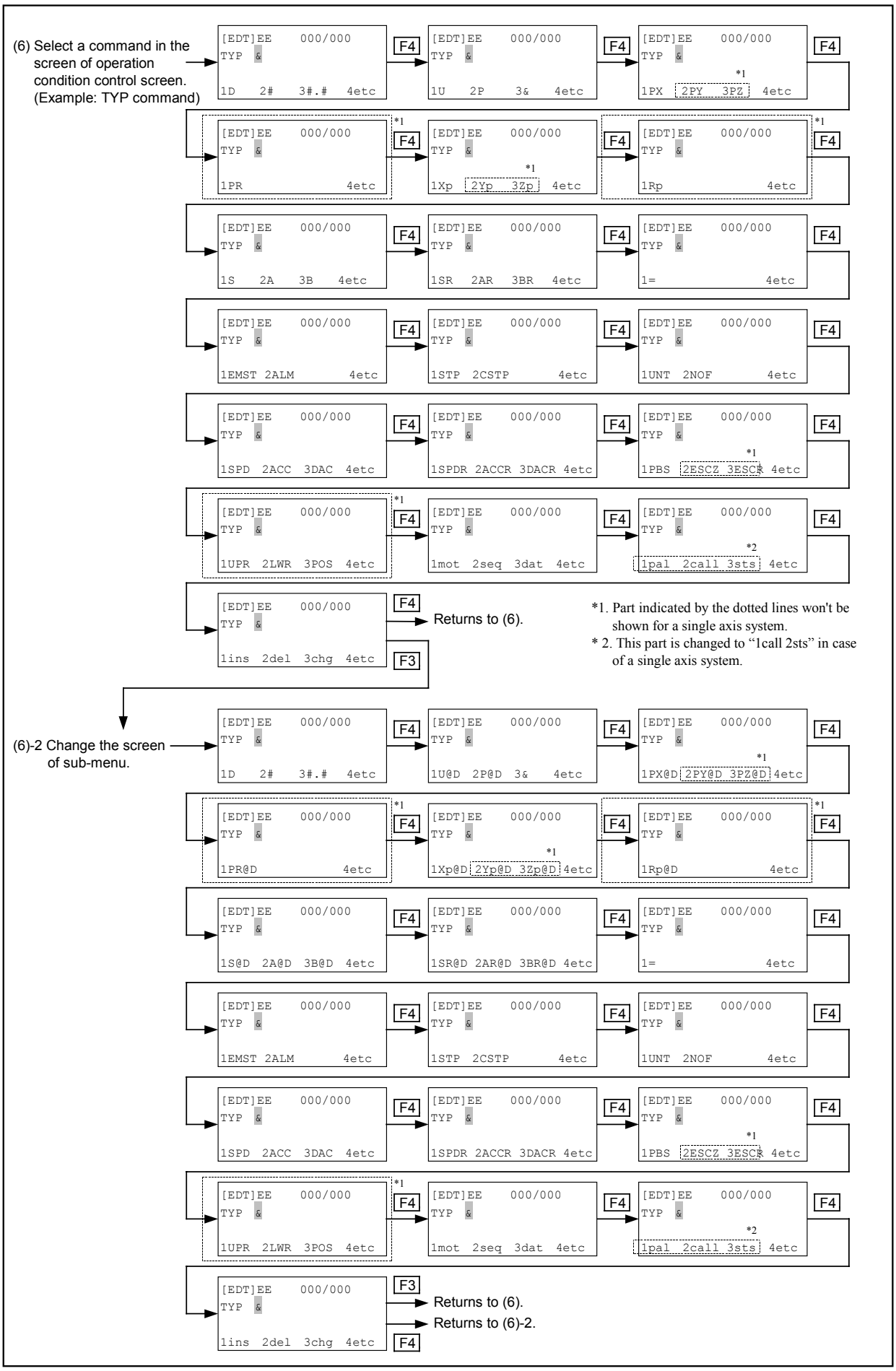


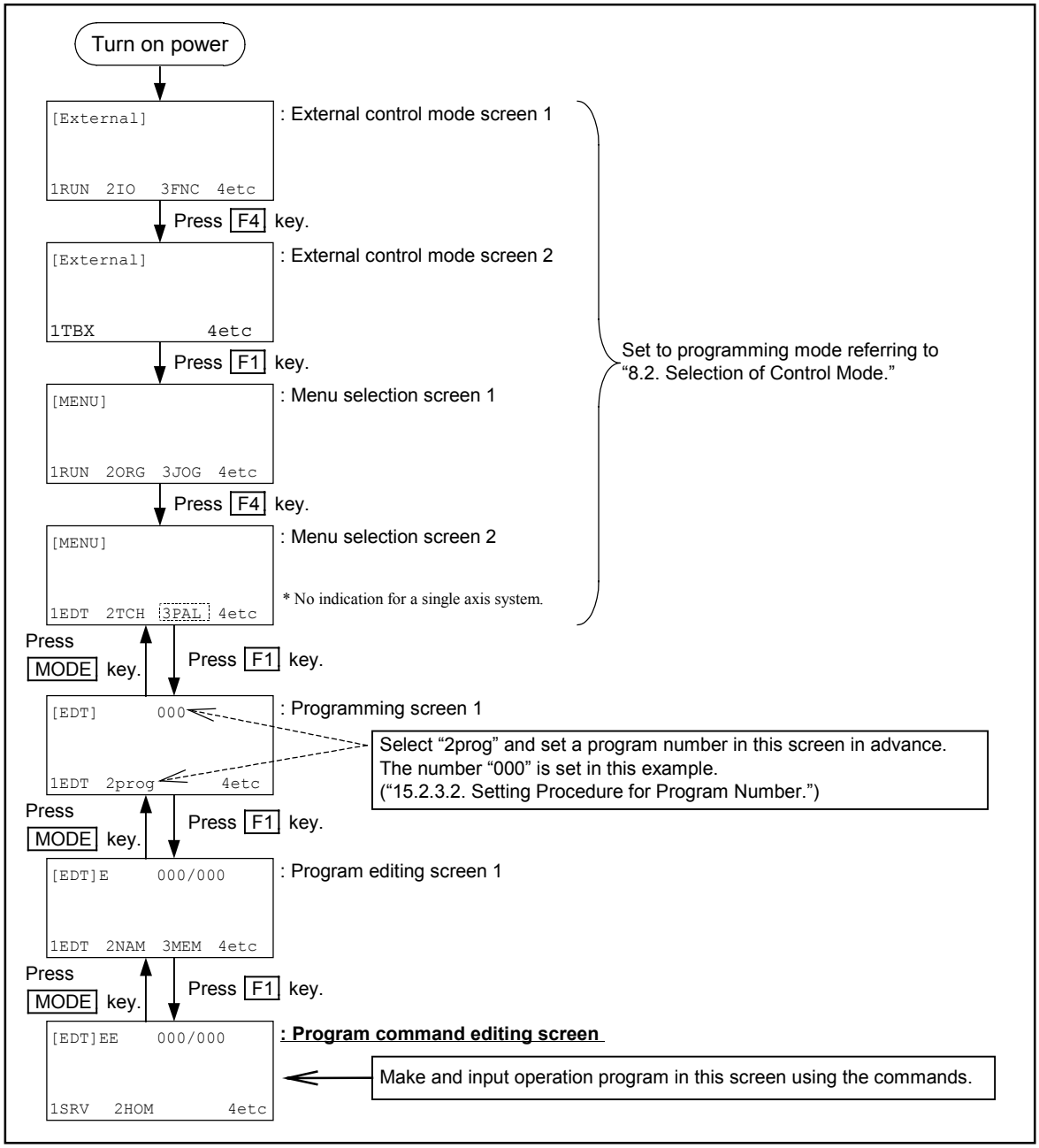
Figure 15-55: Sub-menu (No.7)



15.2.6. Procedure for Editing Program Command

- Inputting and editing typical command are described below. Refer to "15.2.7. Description of Program Command" for other command.
- Input a command, then press SET key to advance to next step automatically. Proceed to input commands for programming.
- See "15.3. Examples of Program" for the examples of program.

Figure 15-56: Editing program command



15.2.6.1. Editing Program Command: MOV

Exercise example: MOV P1234 S300 A5 B2 &AFW

MOV : Command for linear motion.
P1234 : Move to the point on the coordinates 1234.
S300 : Moving speed 300 mm/s
A5 : Acceleration 5m/s^2
B2 : Deceleration 2m/s^2
&AFW : A --- Positioning in absolute coordinates.
F --- FIN (complete positioning) output is on.
W--- Normal process. (No multitask.)

Figure 15-57: Editing MOV command: No.1

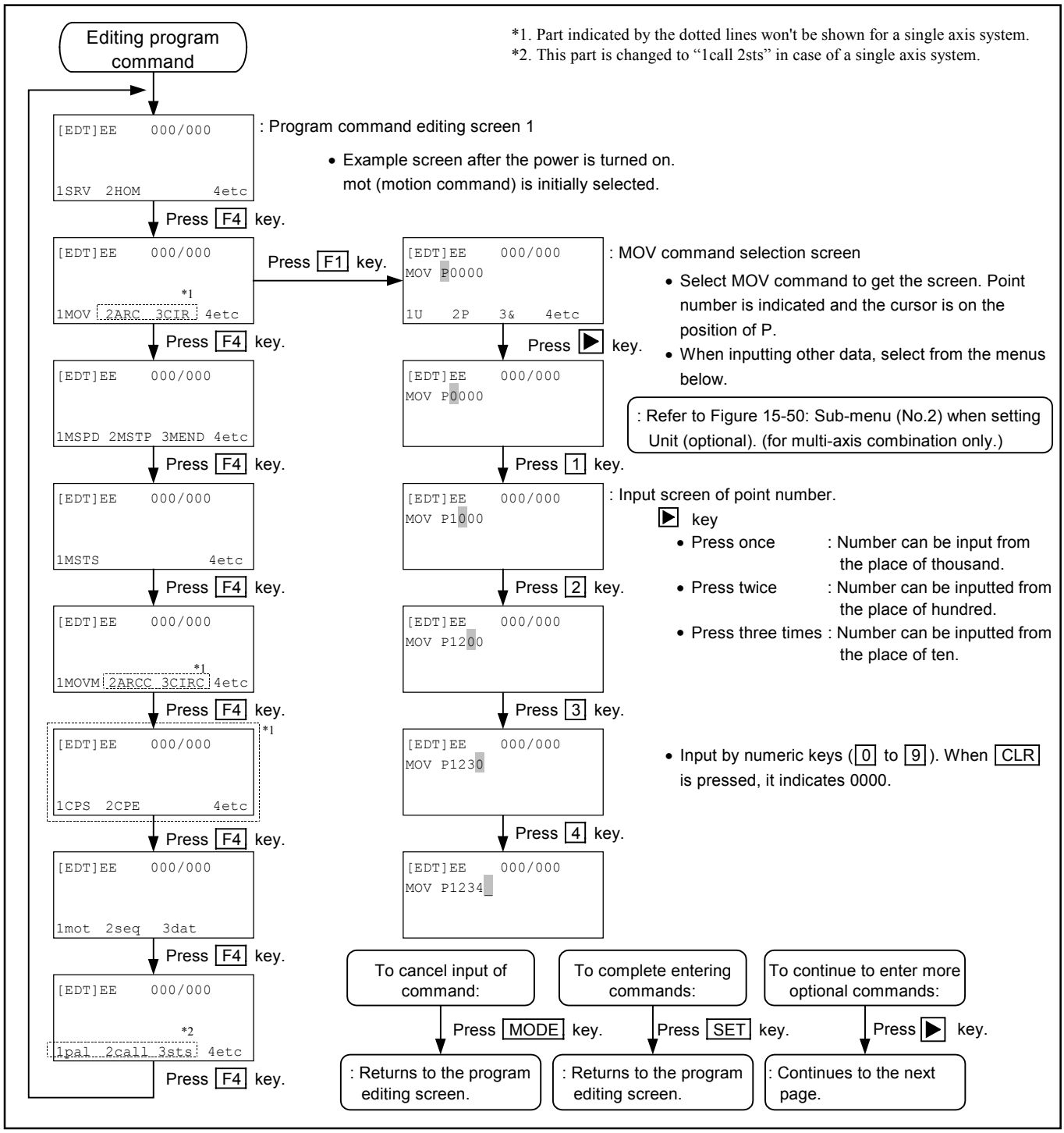
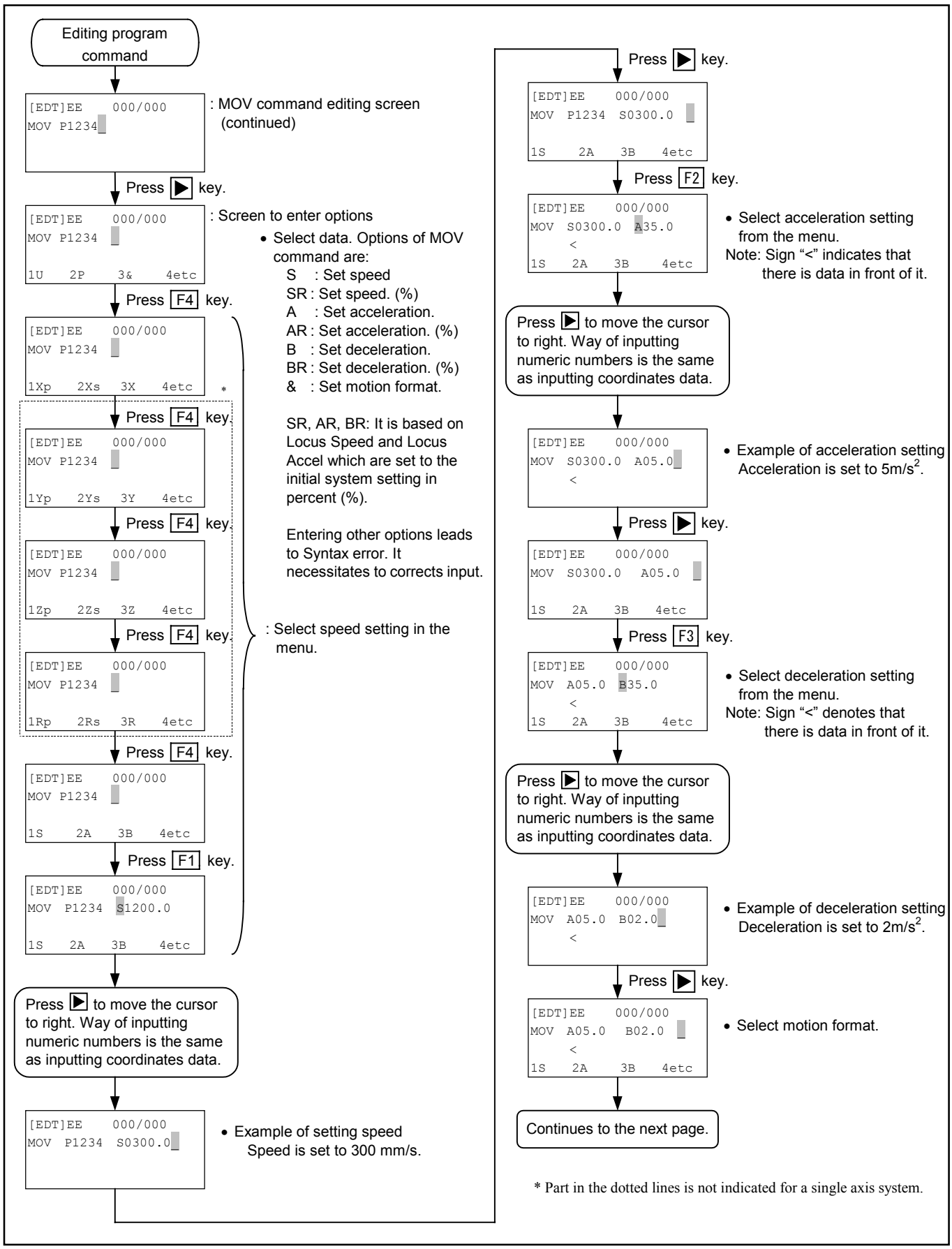
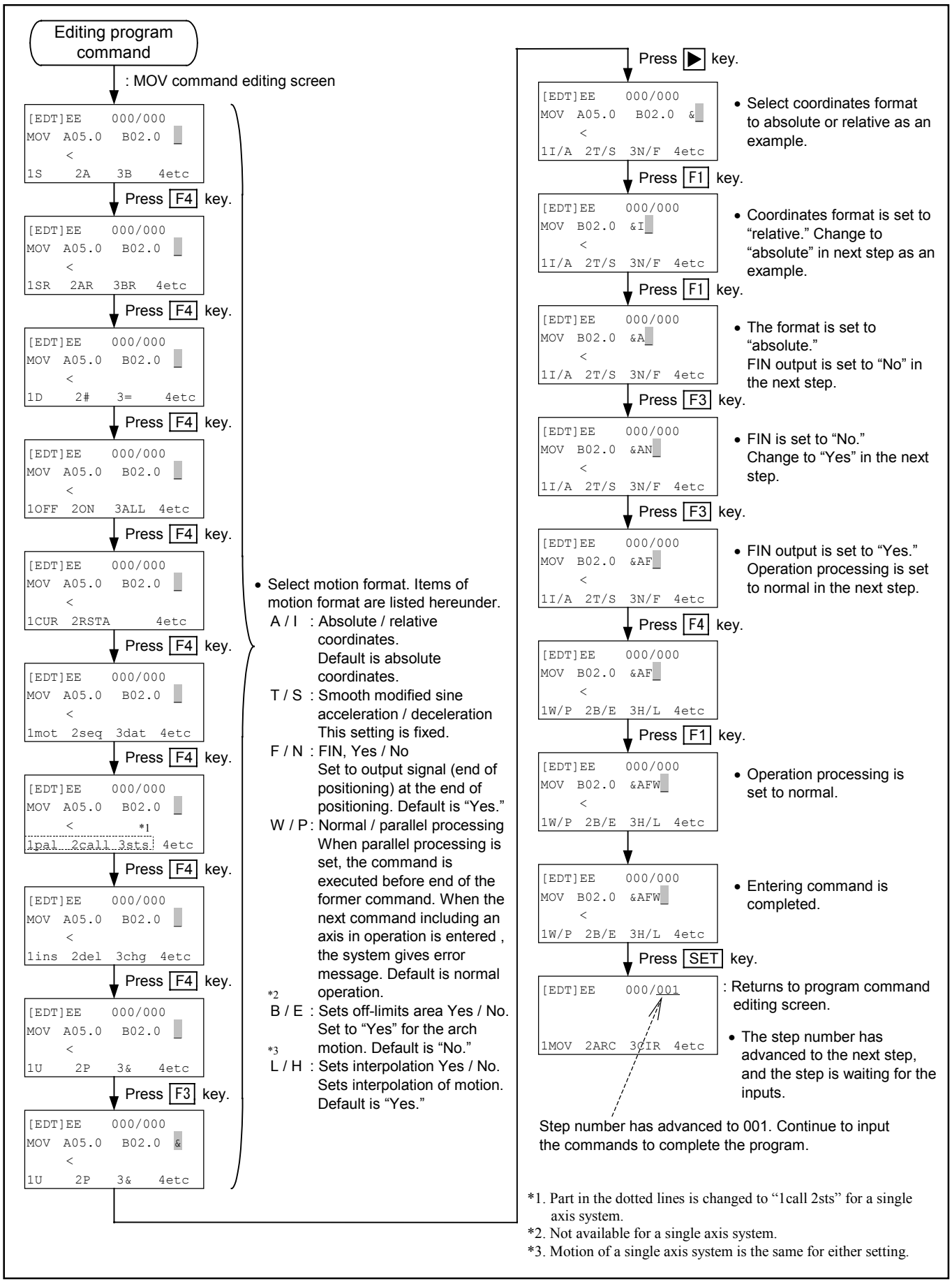


Figure 15-58:Editing MOV command: No.2



* Part in the dotted lines is not indicated for a single axis system.

Figure 15-59: Editing MOV command: No.3



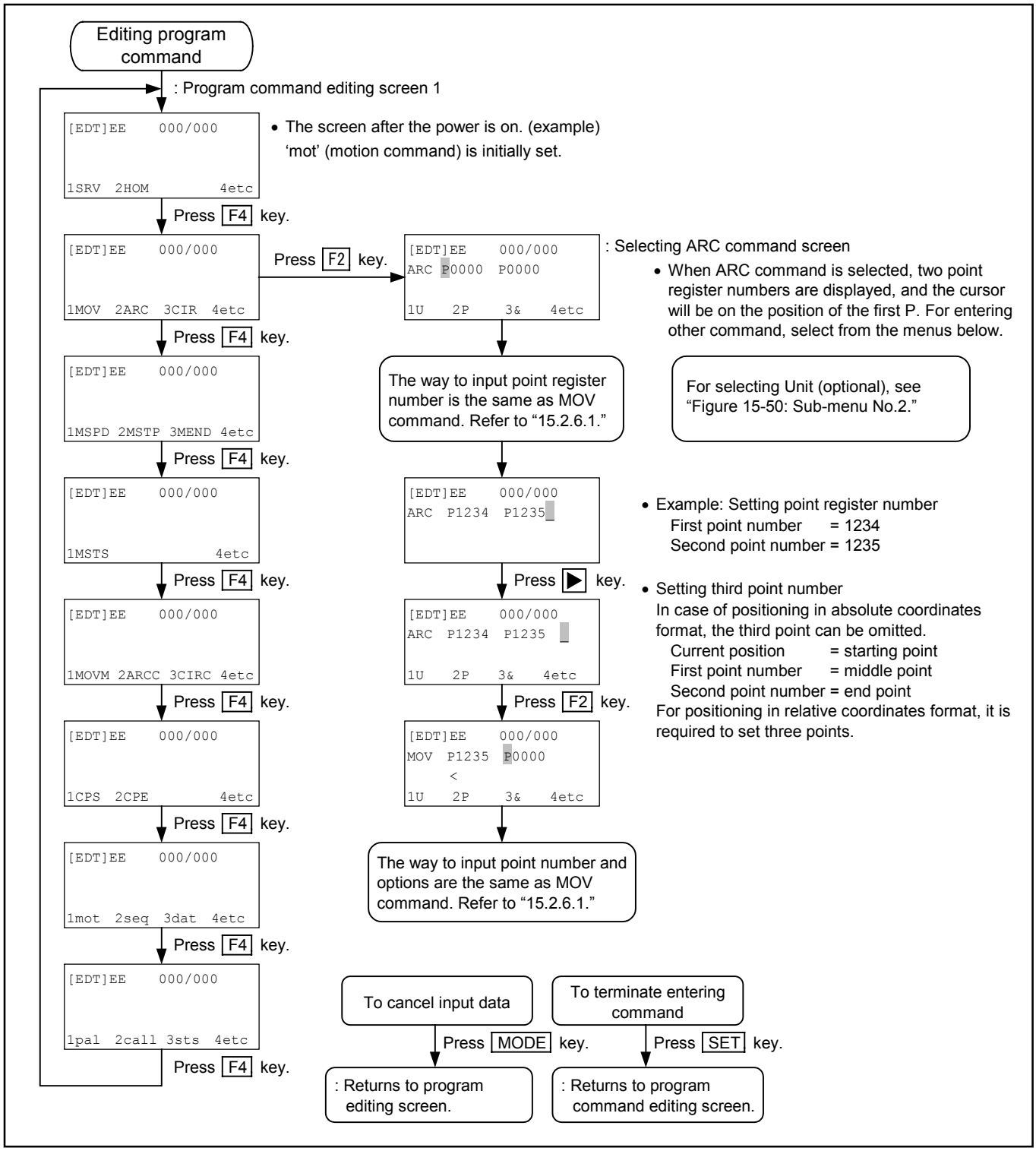
*1. Part in the dotted lines is changed to "1call 2sts" for a single axis system.
*2. Not available for a single axis system.
*3. Motion of a single axis system is the same for either setting.

15.2.6.2. Editing Program Command: ARC (for multi-axis combination only.)

Exercise example: ARC P1234 P1235

ARC : Circular arc interpolation command.
P1234, P1235 : Execute circular arc interpolation passing the points of point register number P1234 and P1235.
In case of absolute coordinates, set two points for the interpolation as the current position becomes the starting point. For positioning in relative coordinates, it is required to set 3 points for the interpolation.

Figure 15-60: Editing ARC command

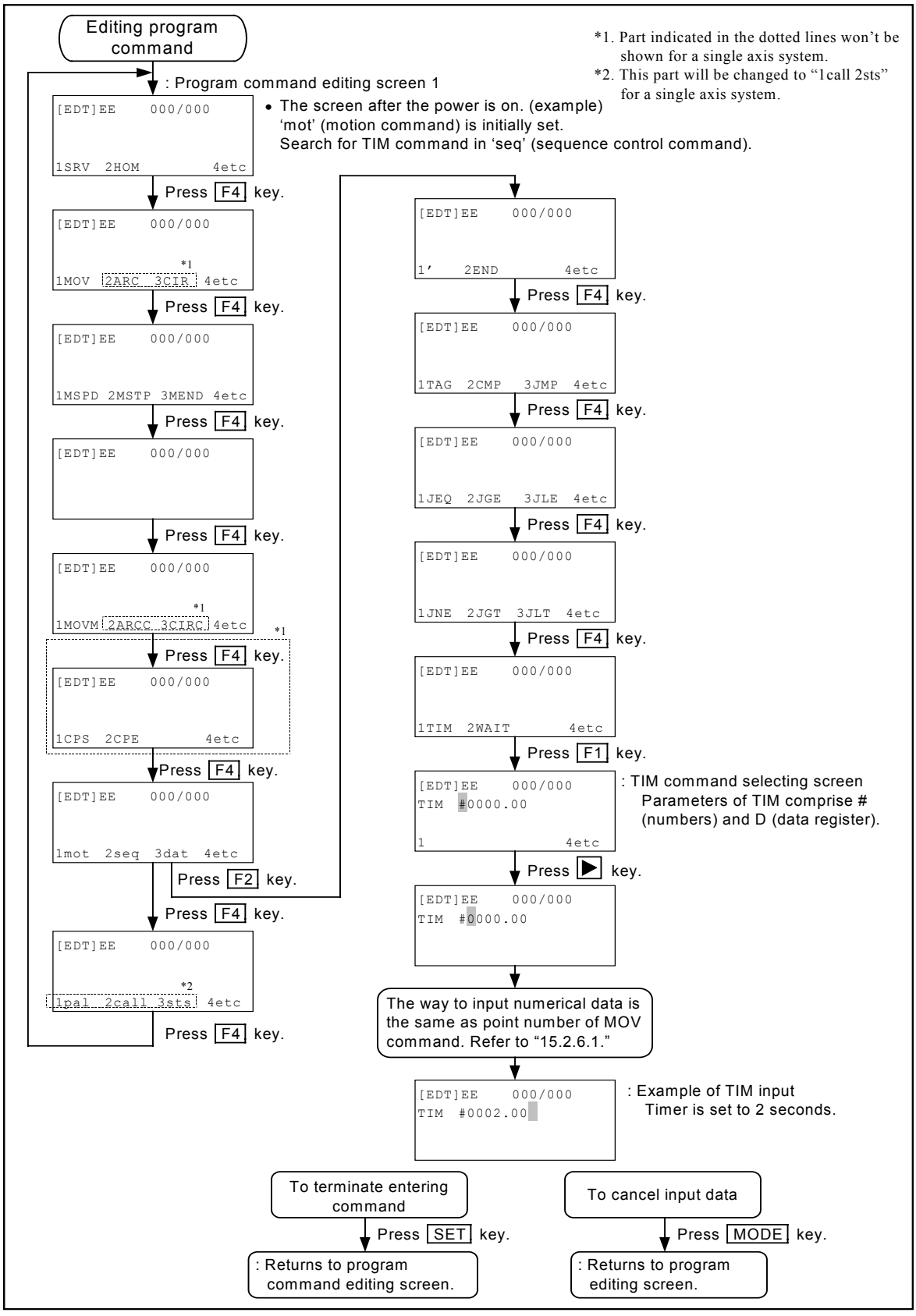


15.2.6.3. Editing Program Command: TIM

Exercise example: TIM #0002.00

TIM : Command for timer.
#0002.00 : Timer is set to 2 seconds.

Figure 15-61: Editing TIM command

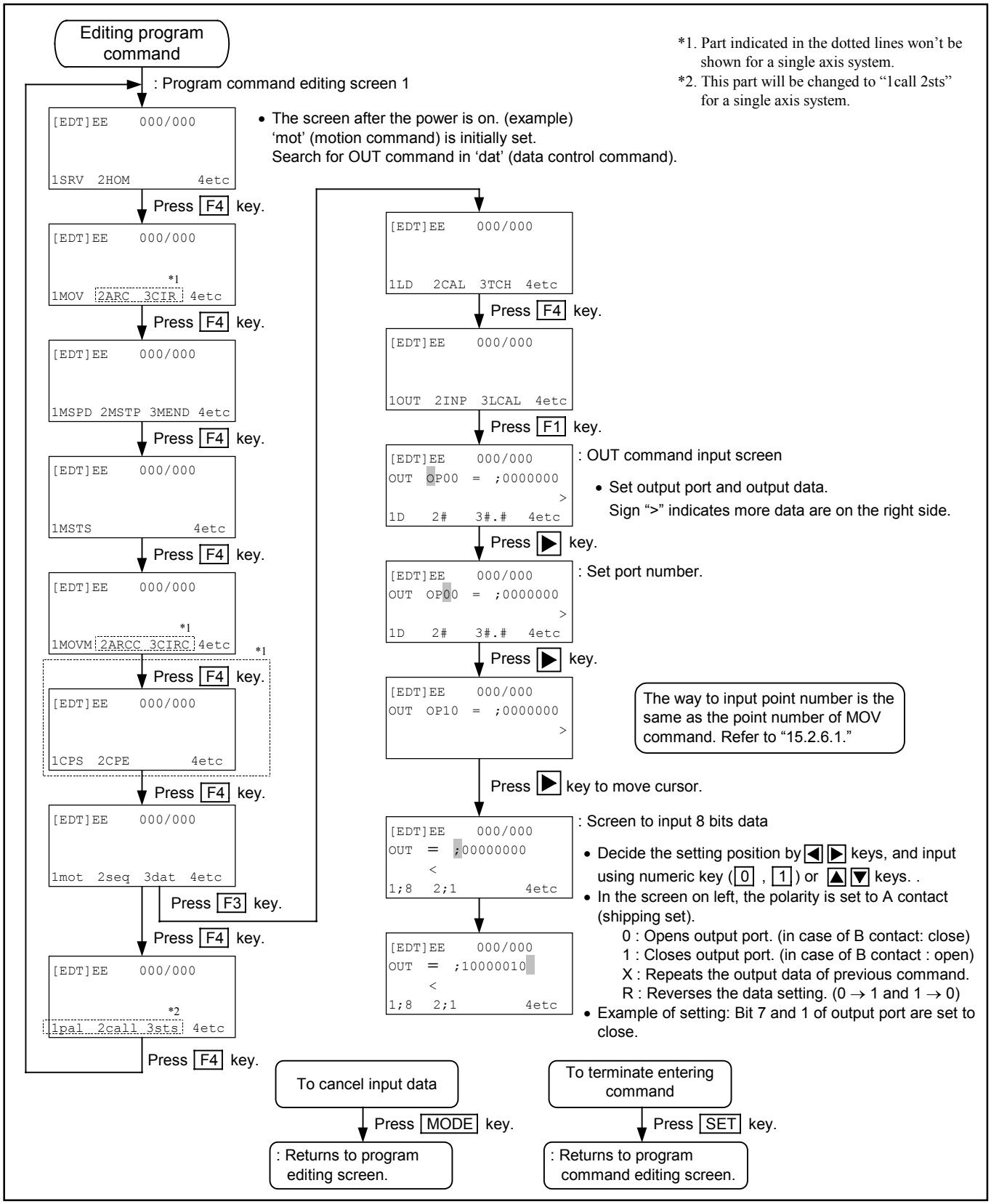


15.2.6.4. Editing Program Command: OUT

Exercise example: OUT OP00 ; 10000010

OUT : Command to set output port.
OP10 : Set output port to OP10 (OB100 ~ OB107) of general IO port.
; 10000010 : Close output bit 1 (OB101) and bit 7 (OB107).

Figure 15-62: Editing OUT command



15.2.6.5. Editing Program Command: PALI (for multi-axis combination only.)

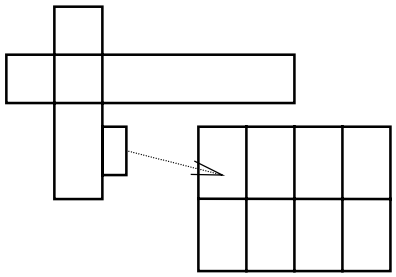
Exercise example: PALI QN12 #0003

PALI : Command to initialize palletizing operation
QN12 : Specify palletizing number to palletizing operation number 12.
#0003 : Set #0003 as the palletizing number.

- For execution of palletizing operation, it is necessary to make the palletizing data in palletizing mode, then set the palletizing number (or name) to it and store them in advance.
  (Refer to "13.1. Palletizing" for making palletizing data.)
  PALI command correlates the palletizing number, which are set in advance, with the palletizing operation number QN (write a palletizing data number on QN), then initializes the palletizing operation to make it operable.

- By correlating the same palletizing number with different numbers of palletizing operatin QN, they can be controlled and operated separately. (This is effective to make the same palletizing operation simultaneously for two sets of X - Y axis combinations in different place.)

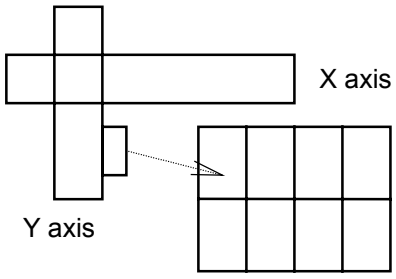
Example 1) Execute palletizing number #0003 by palletizing opeation number 12.



Editing program command
PALI QN12 #0003

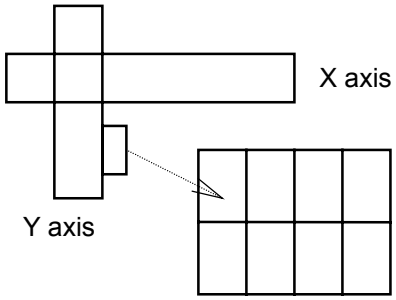
← Palletizing operation program #0003

Example 2) Execute palletizing number #0003 in palletizing operations 12 and 5.



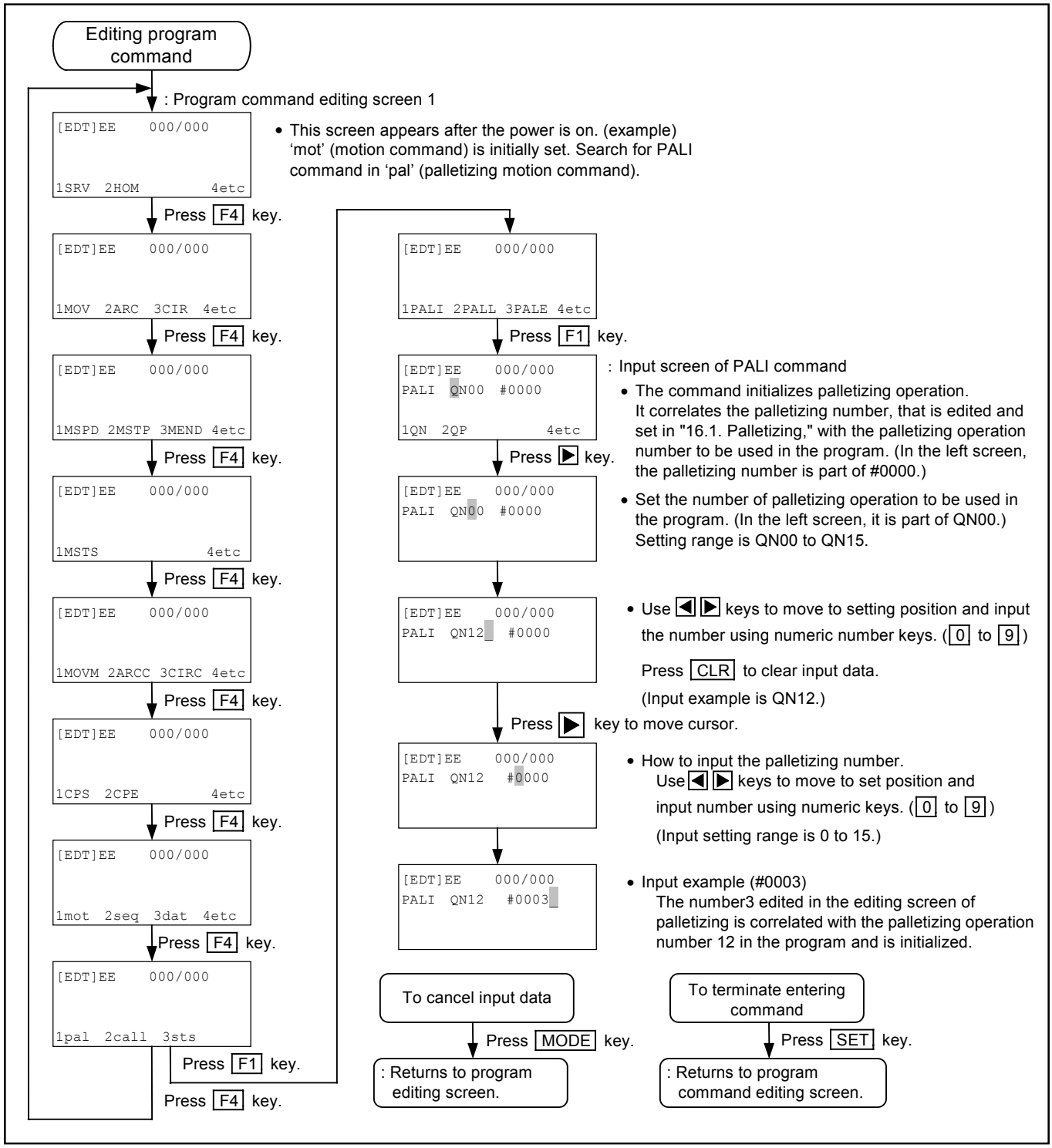
Editing palletizing operation program
PALI QN12 #0003
PALI QN5 #0003

← Palletizing operation program #0003
(Execute on palletizing operation number QN12.)



← Palletizing operation program #0003
(Execute on palletizing opeartion number QN5.)

Figure 15-63: Editing PALI command

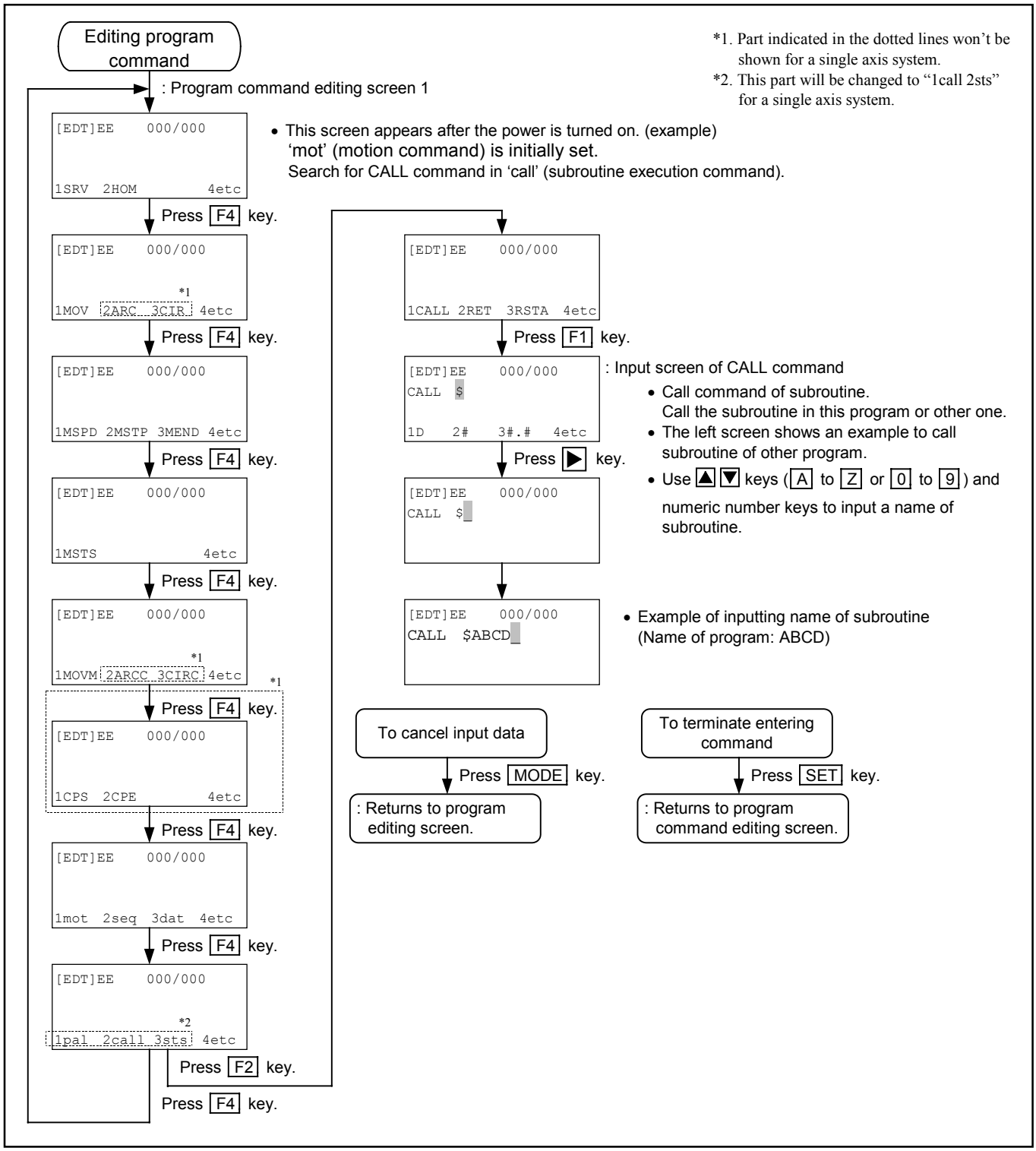


15.2.6.6. Editing Program Command: CALL

Exercise example: CALL $ABCD

CALL : Command to call subroutine.
$ABCD : Call the program named $ABCD.

Figure 15-64: Editing CALL command



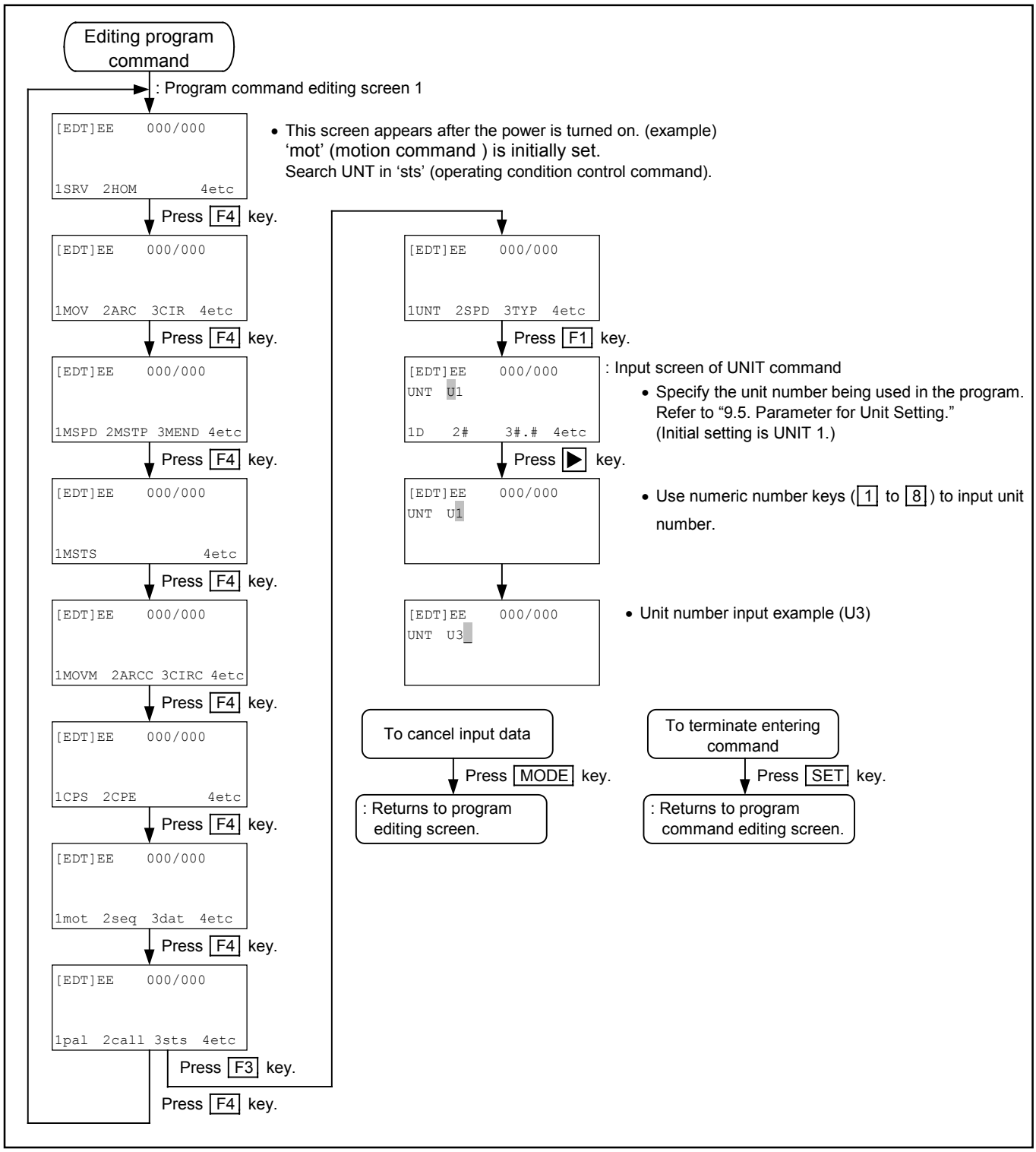
15.2.6.7. Editing Program Command: UNT (for multi-axis combination only.)

(Refer to "9.5. Parameter for Unit Setting.")

Exercise example: UNT U1

UNT : Command to set unit.
U1 : Specify unit number 1.

Figure 15-65: Editing UNIT command

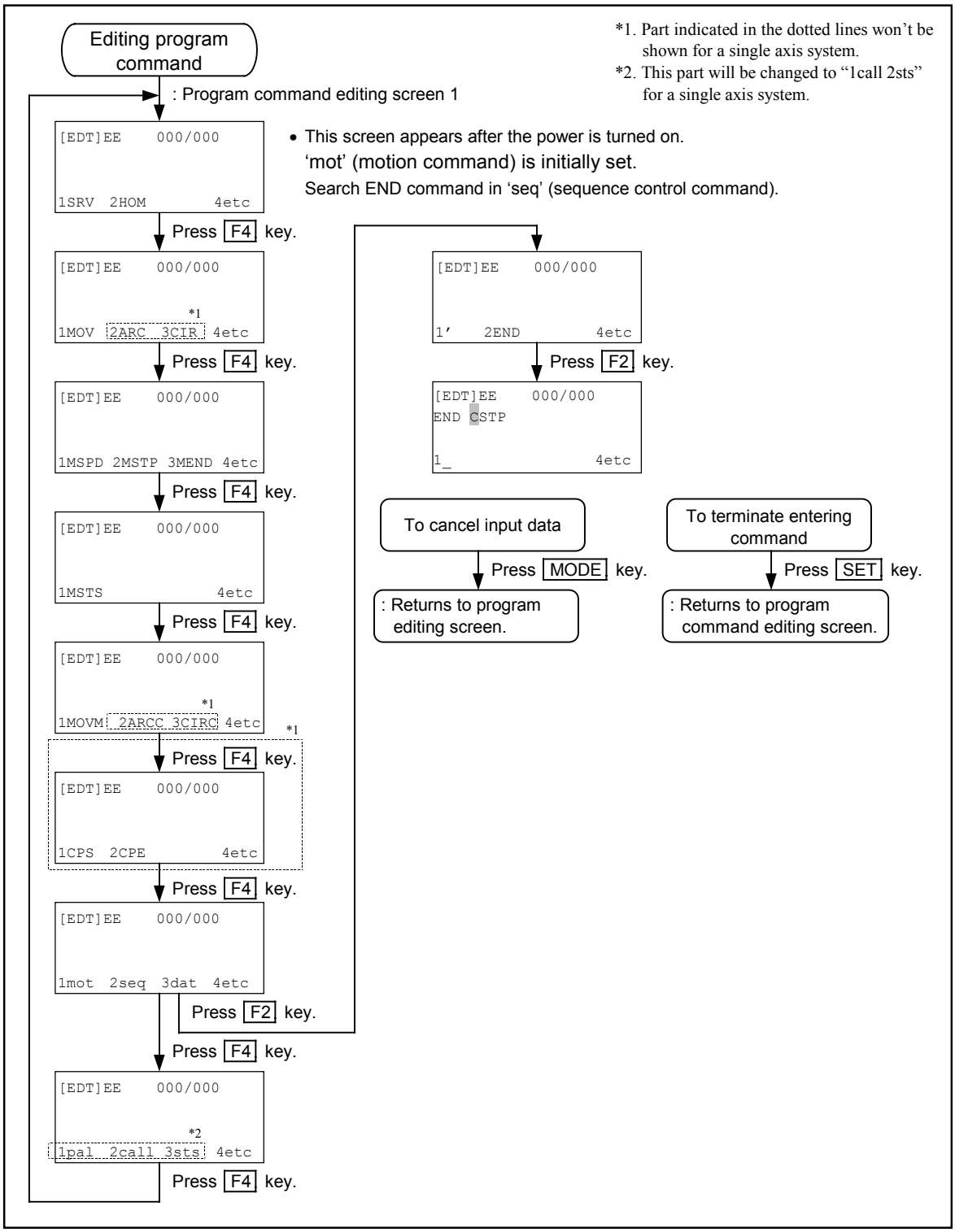


15.2.6.8. Editing Program Command: END

Exercise example: END CSTP

END CSTP: Complete the programmed operation.

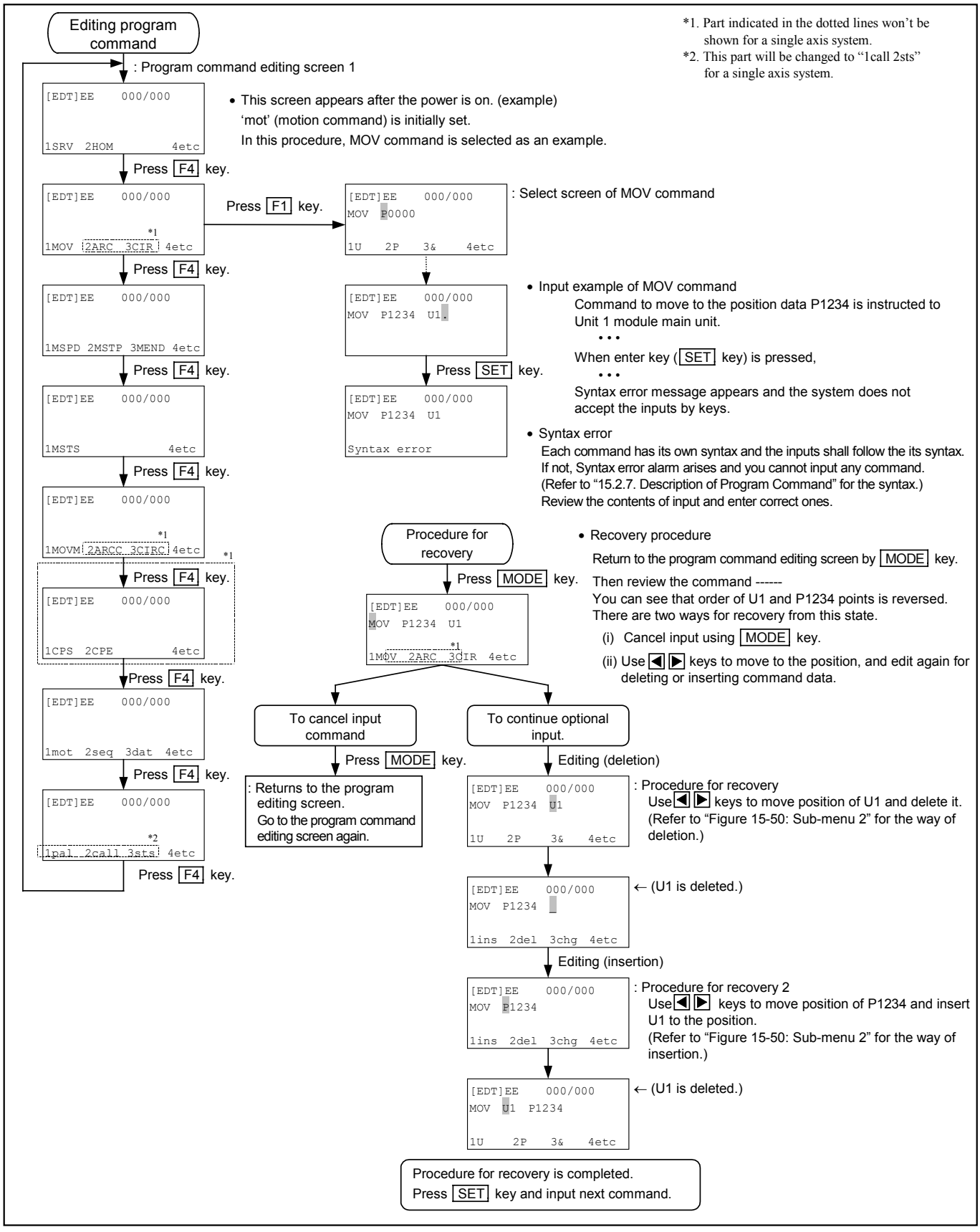
Figure 15-66: Editing END command



15.2.6.9. Editing Program Command: Recovery from Syntax Error

- Input of command will be disabled when a syntax error occurs due to incorrect instruction. The following procedures show the way of how to recover from the syntax error. Be sure to input the correct instructions.

Figure 15-67: Procedure for recovery from Syntax error



15.2.7. Description of Program Command

15.2.7.1. Usable Data for Program

1 D×××--- Data register

- 200 data registers, from register number D000 to D199, are available. Effective data range of contents in data register is between - 9999.99 to +9999.99.

  * Data register number D200 to D999 are reserved.

2 P×××× --- Point register (Coordinate data)

- Point registers between P000 and P3999 are available. Number of coordinate data in accordance with controllable axes of EXEA controller may be set to a point register. For a single axis controller, only one coordinate data of X axis can be registered, while for a multi-axis controller, four coordinate data of X to R axes can be registered. Effective range of data is -9999.99 to 9999.99.

  * Point register numbers P4000 to P9999 are reserved.

- Normally a point register number is specified by numeric numbers. However, it is possible to specify a point register using a data register indirectly. An example of expression is shown below.

  Example: P@D000 --- Contents of D000 indicates the point register number.

- For specifying an axis in point register, put the name of axis (X, Y, Z and R) following letter P. Examples of indication are shown below.

  Example: PX0000 (Specify number directly.), PX@D000 (Specify number indirectly.)

3 X×××, Y×××, Z×××, R××× --- Coordinate data, 'axis switch' and specifying axis

- When specifying the coordinate data, write numeric numbers next to the name of an axis as shown below to set 'axis switch' and 'axis indication'.

  Example: (for multi-axis combination, only X axis can be set for a single axis system.)

  (1) Coordinate data (select Xp～Rp)

  X0001.00 Y0003.00--- Specify coordinate data 1.00 to X axis and 3.00 to Y axis.

  (2) Switching axis (select Xs～Rs)

  X1 Y0 ---------- Set "axis switch" for command such as TCH.

  (3) Specify axis (select X～R)

  X Y ---------- Specify axes for commands such as MSTP and MEND.

- Coordinate data is specified normally by numeric numbers. However, it may be specified by data register indirectly. Example of specifying coordinate data using a data register is shown below.

  Example: X@D000---Contents of D000 indicates status of coordinates or value of X coordinate.

4 S×××, A×××, B××× --- Specifying data of motion speed and acceleration / deceleration

- Speed and acceleration / deceleration data used in the motion command (MOV etc.) are specified as shown below.

  Example:
  S1200------ 1200 (mm/s) is specified to motion speed.
  A10.3 ------ 10.3 (m/s^2) is specified to motion acceleration.
  B10.3 ------ 10.3 (m/s^2) is specified to motion deceleration.

- When the data is set to a motion command, it is only effective in the specific motion command in the same step.

- Speed and acceleration / deceleration data are specified normally by numeric numbers. However, it may be specified by data register indirectly. Examples of indicating data are shown below.

  Example:
  S@D000 ---- Contents of D000 indicate motion speed.
  A@D001 --- Contents of D000 indicate motion acceleration.
  B@D001---- Contents of D000 indicate motion acceleration.

5 SR×××, AR×××, BR××× --- Specifying data of motion speed and acceleration / deceleration in percent

- Motion speed and acceleration / deceleration may be specified by the percentage (%) of the specified speed (resultant speed) and acceleration / deceleration, which are initially set as the parameters. Setting range is 1 to 200.

  Example:
  SR30.5 ------ Motion speed is set to 30.5 % of the specified motion speed
  AR80.5------ Motion acceleration is set to 80.5 % of the specified motion acceleration.
  BR80.5------ Motion deceleration is set to 80.5 % of the specified motion deceleration.

- Normally speed and acceleration/deceleration are specified by the numeric values. However, it is possible to specify those factors indirectly using the data registers. Examples are shown below.

  Example:
  SR@D000-- Contents of D000 indicate ratio to the specified motion speed in %.
  AR@D001 - Contents of D001 indicate ratio to the specified motion acceleration in %.
  BR@D001-- Contents of D001 indicate ratio to the specified motion deceleration in %.

6 Ux --- Unit number

- When the robot system has more than one unit, it is necessary to specify a unit number in the program to the units which are objectives of motion command. Setting range of unit numbers is between U1 and U8. Example of indication is shown below.

  Example:
  U1-------- Specify Unit1 for the subject of motion command.

- Unit number is set normally by numeric keys. However, it may be set by the data resister indirectly. Example is shown below.

  Example:
  U@D000 --- Contents of data register D000 specify unit number.

- Only unit number 1 (U1) can be set to a single axis system.

7 IP×××, Op×××, IB×××OB 02 and Ob××× --- Input / Output port

- It is possible to use Input / Output port number 00 to 01 and 10 to 13. Port number is controlled by 8 bit and set as shown below.

  Example:
  00-------Control port 0
  01-------Control port 1
  10-------General port 0
  11-------General port 1
  12-------General port 2
  13-------Control port 3

  * Port number 02 to 07 and 14 to 77 are reserved.

- Input / Output port is described as shown below when the port is used as one unit.

  Example:
  IP10------ Input, general port 0
  OB13 ---- Output, control port 1

- Describe the I/O port as shown below when only one bit in the port is used. Port number is 3 digits. Two digits number in left indicate port number and one digit in right indicates bit number.

  Example:
  IB101---- General Input port 0 : bit 1.
  OP01 ---- Output control port 1: bit 3

- Way of using Input / Output port is limited by the type of the port. The user may use the general I / O port at its discretion, while the control I / O port is only available for reading out the status.
  Control Input port---------Only possible to read out the status of Input.
  General Input port---------Only possible to read out the status of Input.
  General Output port ------Both reading out and handling are possible.
  Control Output port-------It is possible to reading out the status of Output, while it is not possible to handle the Output port.

- The port number can be set by numeric keys normally. However, it can be set by Data register indirectly. Indication of the setting are shown below.

  Example:
  OP@D000-------Contents of point register D000 indicate the Output port number.
  OB@D000 ------Contents of point register D000 indicate the Output port number or bit number.

8 Pattern of Input / Output

- When setting Output port by OUT command or reading out the status of Input port by CMP command, the task can be done either 8 bit data or 1 bit data. In such a case, Input / Output pattern bit can be specified one by one. Each bit is specified by codes shown below.
  ";" (semicolon) is on the head of the character string of pattern.
  0 ・・・ Output: Off ----------Input: Off
  1 ・・・ Output: On----------Input: On
  X ・・・ Output: No change -----Input: Ignore
  R ・・・ Output: Reverse---------Input: – –

  Example:
  ;0101XR00------8 bit pattern
  ;0 ----------1 bit pattern

9 Character string for naming

- It is possible to specify name of tags and programs by a string of 1 to 8 characters. Capital letters of alphabet and numeric numbers are only usable for naming. For a name of tag, "_ "(under bar) shall be the head of character string while "$" shall be the head of character string for name of a program. ("_ " and "$" are not counted as a character.)

  Example:
  _ABCDEFG ------ Name of a tag "ABCDEFG"
  $ABCDEFG ------ Program "ABCDEFG"

10 Character string for memo

- String of approximately 40 characters can be inserted as the comment to a program. Capital and small letters of alphabet, numeric numbers and the space are usable. An exclusive line for comment shall be set. Character string of comment follows the space after " ' ".

  Example:
  ' ABC DEF GHI -------Comment: "ABC DEF GHI"

11 Numeric data

- When handling numeric data in programming, "#" shall be put on the head of numeric data. Numeric data may be an integer or a decimal fraction.

  Example:
  #1--------- Numeric number.
  # 100.2 -- Coordinates, indication of percentage, acceleration/deceleration and speed, etc.

12 &××× --- Motion format of unit

- When operating a unit by motion command (such as MOV), it is possible to specify the motion format. Motion format data begins with & and the codes of motion format follow it. The motion format is listed below. The paired items cannot be set simultaneously.

- There are formats that cannot be specified depending on command.
  (Refer to "15.2.7.3. Program Command.")

- In addition to it, the rules listed below shall be applied to specify the motion format.

  (1) At least, one motion code shall be specified.

  (2) Do not repeat motion code.

  (3) The motion format shall be appeared in the following order.
  (A / I → T / S → F / N → W / P → B / E* → L / H)
  A / I ---- Absolute coordinates / Relative coordinates
  T / S---- Reserved: Either setting is for smooth modified sine acceleration / deceleration.
  F / N --- Set FIN / No FIN (FIN: Output for end of positioning)
  W / P--- Normal processing command / Parallel processing command
  *B / E --- Sets turnout. B: No E: Yes (Sets arch motion.)
  L / H --- Sets linear interpolation: L: Yes H: No
  [Effective only for linear interpolation (MOV and MOVM). If linear interpolation is not set, each axis will move independently under Locus speed and Locus accel. For this time, motion of each axis is set to its maximum speed and acceleration/deceleration.]

  * Not available for a single axis system.

- For instance, motion formats are initially set as follows.
  &ASFWBL (&ASFWL for a single axis system.)

- Example shown below are for improper order setting.
  &SAFL (incorrect) → &ASFL (correct)

13 QN××, QP×××× --- Control palletizing data (for multi-axis combination only.)

- In palletizing operation command, the control data different from the point register (P××××) is used for positioning. Setting ranges of two control data are shown below.
  Palletizing operation number (QN): QN00 to QN15 (total 16 operation numbers)
  Palletizing position number (QP) : QP0000 to QP9999 (total 1000 palletizing position numbers)

  * QN 16 to QN99 are reserved.

  The maximum number of palletizing position number QP×××× varies according to the setting format of a pallet.

- Palletizing operation and palletizing position number are normally specified in numeric numbers. However, it is possible to specify them by data register indirectly. The followings show the examples.

  Example:
  QN@D000 ------Contents of D000 show the palletizing motion number.
  QP@D000-------Contents of D000 show the pallet number.

14 MP××, MB××× (Imaginary Input / Output port)

- This is to hold Input / Output pattern of actual input and output (IP, OP) and to judge status of a port.

- Port numbers 0 to 1 are able to use for imaginary I/O port. The imaginary I/O port is kept in the memory same as the data register. It does not affect the outside conditions of EXEA controller.

  * Port number 2 to 7 are reserved.

- It can substitute the actual port (IP, OP) in a program which uses I/O port.

- When it is used as a unit of port, describe the I/O port as shown below.

  Example:
  MP1----------Imaginary I/O port 1

- When use one bit only in the port, describe as shown below. The port number is two digit. The left side indicates the port number and the right indicates the bit number.

  Example:
  MB11----------Bit 1 of imaginary I/O port 1

- Normally the port number is indicated by numeric numbers, however, it is indicated indirectly using data registers. Following shows the examples.

  Example:
  MP@D000 ------Contents of D000 indicate the number of I/O port.
  MB@D000------Contents of D000 indicates the numbers of port and bit.

15.2.7.2. Symbol List to Describe Syntax of Program Command

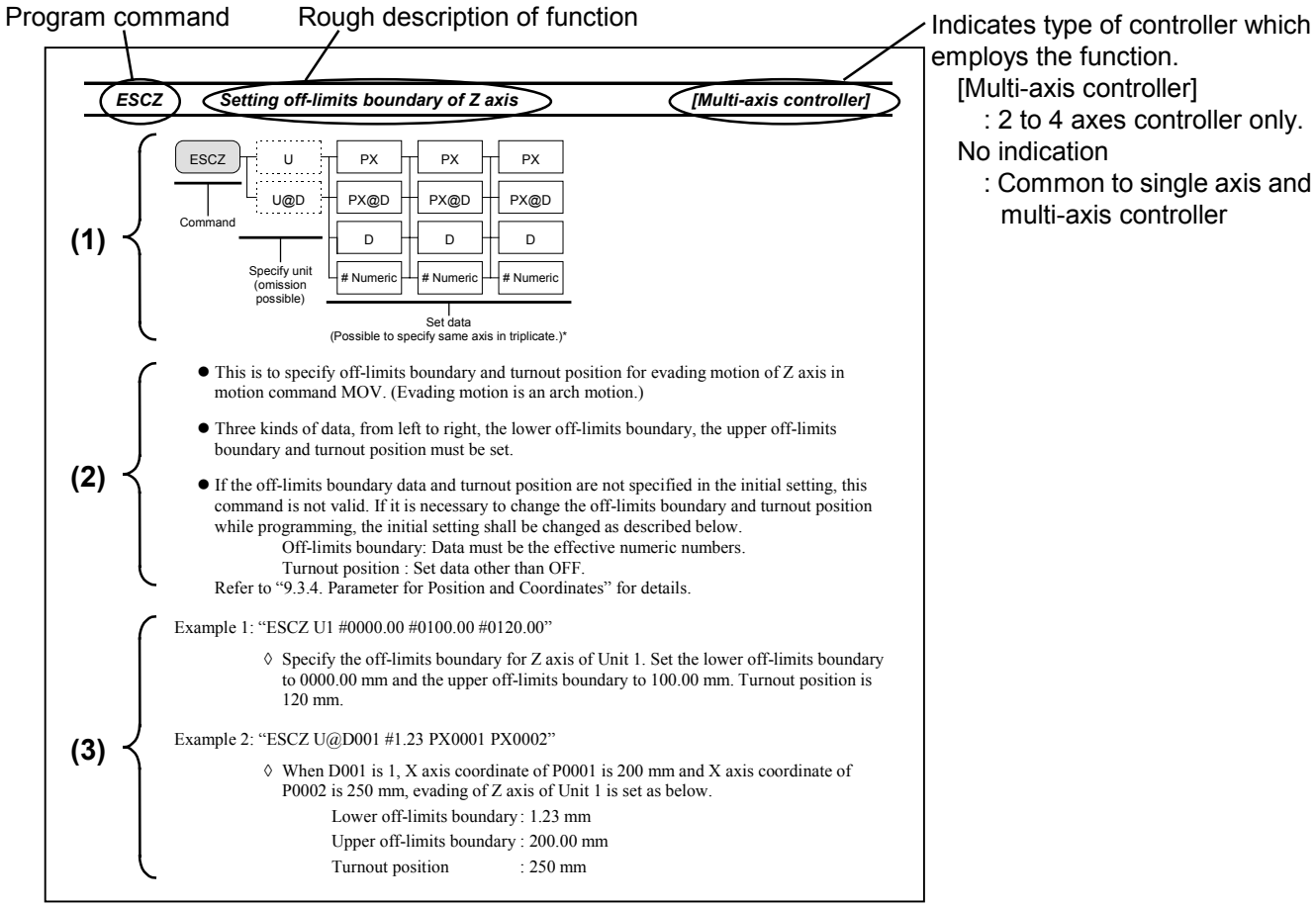
- The symbols listed hereunder are used to describe the syntax of program command. ("15.2.7.3. Program Command.")

	UNT	••• Command		MP	••• Imaginary Input/Output port (direct setting)
	#Numeric	••• Numeric number (direct setting)		IP@D	••• Input port (indirect setting)
	D	••• Data register		OP@D	••• Output port (indirect setting)
	P	••• Point register (direct setting)		IB	••• Input port / every bit (direct setting)
	P@D	••• Point register (indirect setting)		OB	••• Output port / every bit (indirect setting)
*1	PX	••• Point register : Specified axis unit data (direct setting)		MB	••• Imaginary Input/Output • every bit (direct setting)
*1	PX@D	••• Point register : Specified axis unit data (indirect setting)		IB@D	••• Input port / every bit (indirect setting)
*1	X	••• Coordinate data / switch axis / specify axis (direct setting)		OB@D	••• Output port / every bit (indirect setting)
*1	X@D	••• Coordinate data / switch axis / specify axis (indirect setting)		MB@D	••• Imaginary Input/Output • every bit (indirect setting)
	S	••• Motion speed (direct setting)		U	••• Motion unit (direct setting)
	S@D	••• Motion speed (indirect setting)		U@D	••• Motion unit (indirect setting)
	SR	••• Motion speed : Percentage (direct setting)		Pattern	••• Input / Output pattern
	SR@D	••• Motion speed: Percentage (indirect setting)		T character string	••• Character string of Tag name (less than 8 characters)
	A	••• Motion acceleration (direct setting)		P character string	••• Character string of program name (less than 8 characters)
	A@D	••• Motion acceleration (indirect setting)		C character string	••• Character string of comment (less than 40 characters)
	AR	••• Motion acceleration : Percentage (direct setting)		& motion format	••• Motion format of unit
	AR@D	••• Motion acceleration : Percentage (indirect setting)	*2	QN	••• Palletizing operation number (direct setting)
	B	••• Motion deceleration (direct setting)	*2	QN@D	••• Palletizing operation number (indirect setting)
	B@D	••• Motion deceleration (indirect setting)	*2	QP	••• Palletizing position number (direct setting)
	BR	••• Motion deceleration : Percentage (direct setting)	*2	QP@D	••• Palletizing position number (indirect setting)
	BR@D	••• Motion deceleration : Percentage (indirect setting)	*2	PAL name	••• Character string of pallet name
	IP	••• Input port (direct setting)		***	••• Keyword
	OP	••• Output port (direct setting)		***	••• Omission possible

*1. X axis only for a single axis system. Indicate either one of X, Y, Z and R axis for a multi-axis combination.
*2. Do not use in case of a single axis system.

15.2.7.3. Program Command

- The program commands are described hereunder in alphabetical order.
- Syntax, description of function and examples are shown in the manner as indicated below.



(1) Syntax

Describes how to make program command.
Solid lines in the syntax indicate necessary items while dotted lines indicate the items that shall be added in case of need.

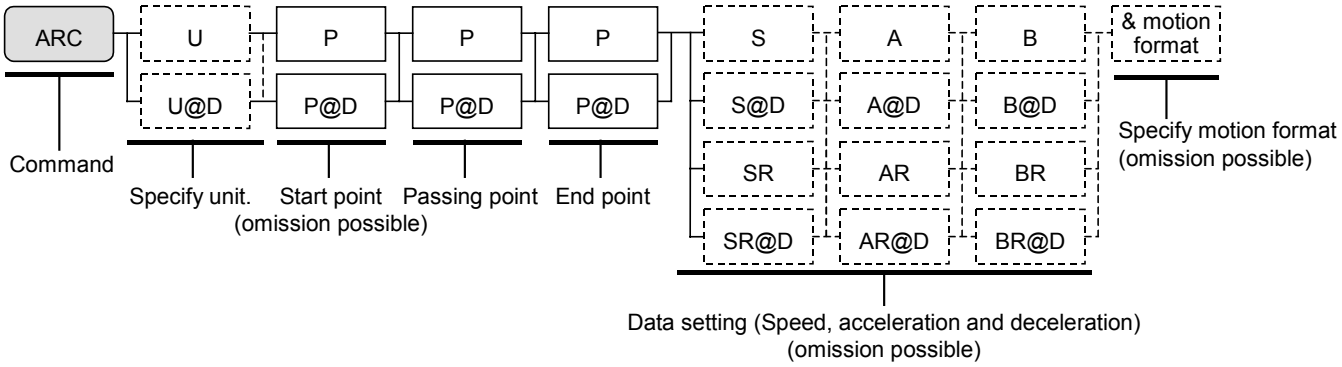
(2) Description

This part describes fundamental function and the points to note.

(3) Examples of program command

Program commands that can be practically used and description of them.

ARC Circular arc motion [Multi-axis controller]



- This is circular arc motion command for a specified unit.
- This command is to execute circular arc interpolation defined by the point registers of start point, passing point 1 and end point.

  Note: An intended circular interpolation cannot be executed or its accuracy may be degraded if the trajectory defined by the coordinates of the three points does not meet the requirements for circular interpolation.

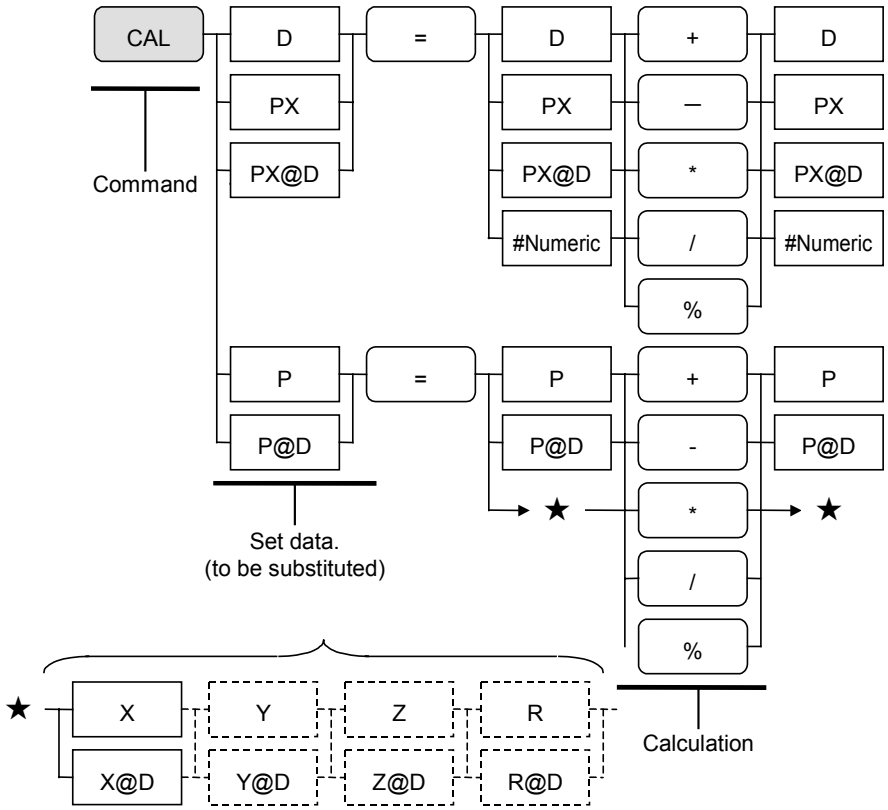
- Use data for motion speed, acceleration and motion format specified by SPD and TYP commands.
- Speed, acceleration and motion format are possible to change in this command. The way of data setting is to add parameters after setting of point register. These changes are only effective for a program step.
- When specifying motion format, it is possible to set only the motion codes that are necessary to change. (Refer to "15.2.7.1. 12 .")
- When "W" (normal processing command) is set as a motion format, the program does not go to the next step until the current step completes. When "P" (parallel processing command) is set, the program operates the subsequent steps in parallel while executing current step. However, if a motion command is programmed in subsequent steps to an axis unit currently in motion, an alarm arises. In such a case, set MEND to wait for completion of motion, or set MSTS command to confirm the motion not to duplicate operating axis unit.
- Motion format of A/I, F/N and W/P are only possible to set. Do not set other motion format as it may lead to an alarm.
- Start point of circular arc motion can be omitted in absolute coordinates (Current position is a start point.), while it cannot be omitted in case of relative coordinates.
- When NOF command is set, be careful for changes of point register numbers.
- When PBS command is set, be careful for changes of positioning point.

Example: "ARC U1 P0001 P0002 P0003"

◊ Execute circular arc motion of Unit 1. The locus of motion shall be on the coordinates of start point register P001, passing point P0002 and end point P0003.

ARCC Reserved (Do not use.)

CAL Calculate data



- While executing programmed operation, this command is set to calculate data between specified points and substitute the results for the specified data register or point register.
- Calculation
  "+" ------- Execute addition.
  "-" -------- Execute subtraction.
  "*" ------- Execute multiplication.
  "/" -------- Execute division.
  "%" ------ Calculate percentage.
- When a result of calculation is over the range of ±9999.99, an alarm arises.

Example 1: "CAL D000 = D001+ D002"

◊ Substitute the result of addition of D001 contents and D002 contents for data register D000.

Example 2: "CAL PX0001 = #1 % #1"

◊ Substitute the result of percentage calculation of 1 and 1 for X axis coordinate of point register P0001.

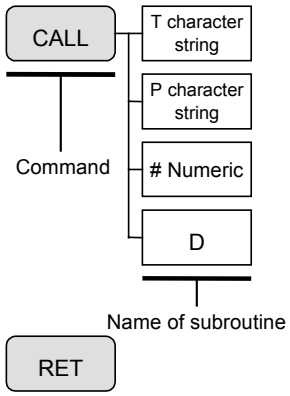
Example 3: "CAL P0001 = X0001.0 + P0002" (in case of single axis)

◊ Substitute the addition of 1 of the X coordinate and the coordinate data of the point register P0002 to the point register P0001.

Example 4: "CAL P0001 = X00001.00 Y0001.00 Z0001.0 R0001.00 + P0002"
(in case of a multi-axis combination)

◊ Add coordinate data of P0002 to X axis 1.00, Y axis 1.00, Z axis 1.00 and R axis 1.00, then substitute the result for point register P0001.

CALL ~ RET Call subroutine ~ End of subroutine

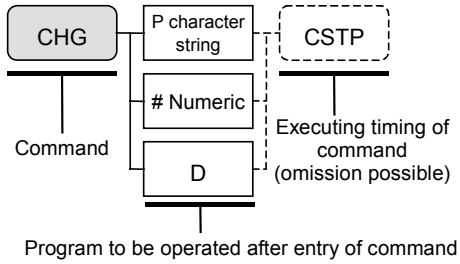


- Change sequence by making jump to a specified subroutine during programmed operation.
- It is possible to specify a local subroutine in the same program selected by TAG command or a subroutine in another program. The subroutine in another program can be set by the program name and number (data or data register).
- RET command terminates the subroutine processing and the program moves to next step of CALL command after end of the subroutine.
- Up to quadruple call between CALL and RET commands are possible.

Example: "CALL _ SUBPRG"

◊ Change sequence to a local subroutine that begins with a step tagged name of "_SUBPRG".

CHG Change operating program

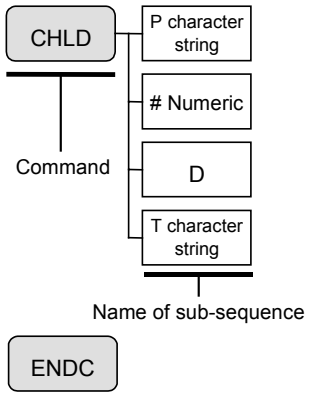


- This command is to stop the main program in the middle of execution and switches to a specified new program.
- Interrupts all motions, even there are moving unit or other sequence is in multitask operation, and switches the programmed operation to another program.
- When program is switched, all data or settings of data register (D00 to D199), repetition and call of subroutine will be initialized.
- Executing timing of the command
  No setting--------Stops all sequence immediately and changes program.
  "CSTP" ----------Execute cycle-stop of all sequence and changes program.

Example: "CHG D003 CSTP"

◊ Execute cycle stop to all sequence, then change to an operation in the program of which number is shown in the contents of D003.

CHLD ~ ENDC Multitask operation (Start and complete of sub-sequence)

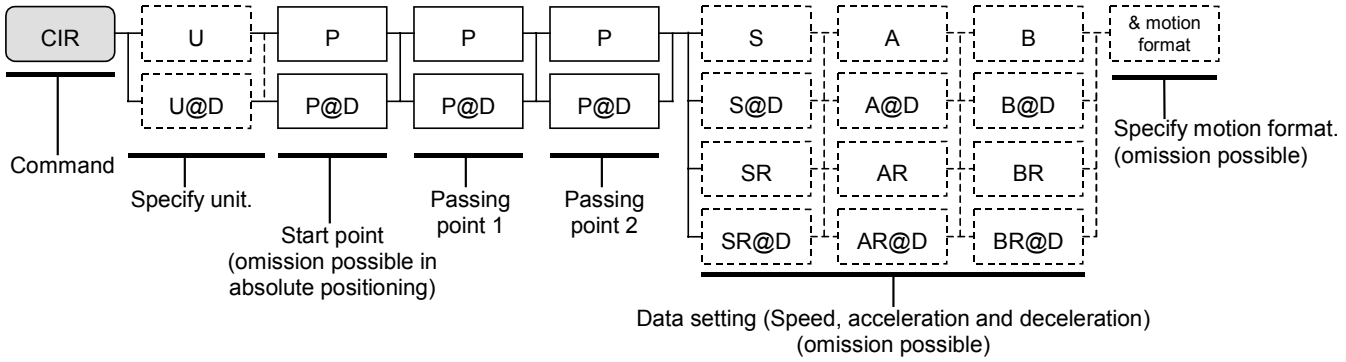


- It is possible to execute up to 15 sub-sequences for a multi-axis controller and up to 3 sub-sequences for a single axis controller (multitask) operation.
- A sub-sequence operates independently to the end of main sequence, or operates independently until ENDC command is executed to finish its own sequence.
- It is possible to execute a different program or the steps after TAG command. Specify the different program by its name or program number (numerical value, data register).
- Data of programmed operation such as data register or point register are common to the main sequence and other sub-sequences.

Example: "CHLD $SUBSEQ"

◊ Start a program named "SUBSEQ" as a sub-sequence.

CIR Circular motion [Multi-axis controller]



- This is to executed circular motion of a specified unit.
- This command is to execute circular interpolation defined by the point registers of start point, passing point 1 and passing point 2.

  Note: An intended circular interpolation cannot be executed or its accuracy may be degraded if the trajectory defined by the coordinates of the three points do not meet the requirements for circular interpolation.

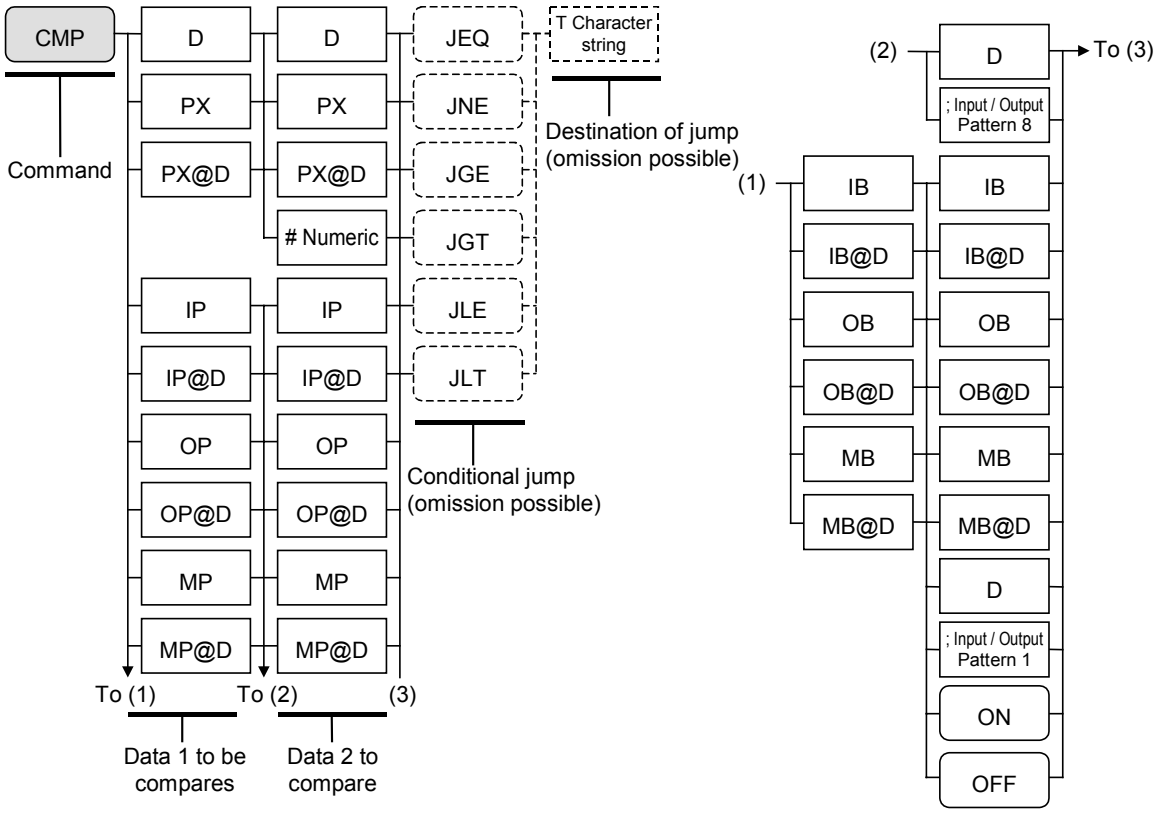
- The data for motion speed, acceleration and motion format are normally instructed by SPD and TYP commands.
- Speed, acceleration and motion format are possible to change in this command. The way of data setting is to input parameters after a point register. These changes are only effective in a program step.
- When specifying motion format, it is possible to set only the motion code which is necessary to change. (Refer to "15.2.7.1. 12 .")
- When "W" (normal processing command) is set as a motion format, the program does not go to the next step until the current step completes. When "P" (parallel processing command) is set, the program operates the subsequent steps in parallel while executing current step. However, if a motion command is programmed in subsequent steps to an axis unit currently in motion, an alarm arises. In such a case, set MEND to wait for completion of motion, or set MSTS command to confirm the motion not to duplicate operating axis unit.
- Motion format of A/I, F/N and W/P are only possible to set. Do not set other motion format as it may lead to an alarm.
- Start point of circular motion can be omitted in absolute coordinates. (Current position is a start point.) It cannot be omitted in case of the relative coordinates.
- When NOF command is set, be careful for changes of point register numbers.
- When PBS command is set, be careful for changes of positioning point.

Example: "CIR U1 P0001 P0002 P0003"

◊ Execute circular motion with the unit 1 defined by the point registers P0001 (start point), P0002 (passing point 1) and P0003 (passing point 2).

CIRC Reserved (Do not use.)

CMP Compare data (with jump)



- This command is to compare two data and keeps judgment flag. This flag is used for the condition of conditional jump instruction.

- Conditional jump and destination of jump can be set in the CMP command. Set the destination of jump by approximate 8 characters of number and alphabets.

- ON is 1 and OFF is 0 for comparing Data 2.
  "JEQ" --- Jump to specified Tag when (Data 1 = Data 2).
  "JNE" --- Jump to specified Tag when (Data 1 ≠ Data 2).
  "JGE" --- Jump to specified Tag when (Data 1 ≥ Data 2).
  "JGT" --- Jump to specified Tag when (Data 1 > Data 2).
  "JLE"---- Jump to specified Tag when (Data 1 ≤ Data 2).
  "JLT"---- Jump to specified Tag when (Data 1 < Data 2).

Example 1: "CMP IP10 ;00000001 JEQ _ABCDEF"

◊ When state of general input port IP10 is "00000001," jump to a step of which Tag name is "_ABCDEF."

Example 2: "CMP D001 PX0001 JLT _ABCD"

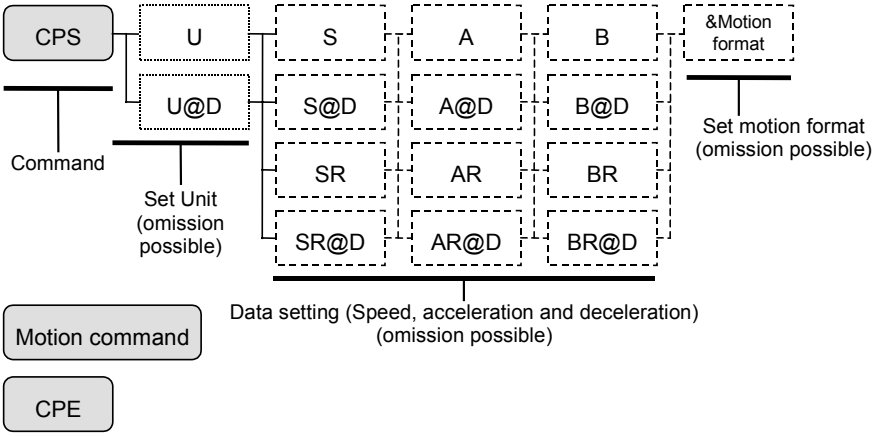
◊ Compare contents of data register D001 and X coordinate data of point register P0001, jump to the step of which Tag name is "_ABCD" when date register D001 is smaller than point register P0001.

Example 3: "CMP IB117 ON JEQ _1"

◊ When general input port IB117 is ON, jump to the step of which Tag name is "_1."

- Execute jump by J×× instruction after the step to which CMP command is set when conditions and destination of jump are omitted.

CPS ~ CPE Specify data for continuos path control [Multi-axis controller]



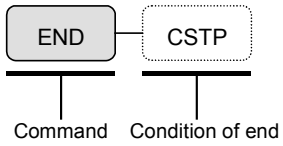
- These commands are used for continuos path operation. They start a continuos path operation of a specified unit using the steps, which are put between CPS and CPE, as the data.
- Normally use the data specified by SPD and TYP commands for motion speed, acceleration and motion format.
- Speed, acceleration and motion format are possible to change in this command. Add data setting for changes. These changes are only effective in the continuous path between CPS and CPE.
- When specifying motion format, it is possible to set only the motion codes which are necessary to be changed. (Refer to "15.2.7.1. 12 .")
- When "W" (normal processing command) is set as a motion format, the program does not go to the next step until the current step completes. When "P" (parallel processing command) is set, the program operates the subsequent steps in parallel while executing current step. However, if a motion command is programmed in subsequent steps to an axis unit currently in motion, an alarm arises. In such a case, set MEND to wait for completion of motion, or set MSTS command to confirm the motion not to duplicate operating axis unit.
- Motion format of A/I, F/N and W/P are only possible to set. Do not set other motion format as it may lead to an alarm.
- Set all motion command between CPS and SPE to absolute coordinate positioning ("A" or default) when the motion format is specified to "I" (relative coordinate positioning).
- When NOF command is set, be careful for changes of point register numbers.
- When PBS command is set, be careful for changes of positioning point.
- Following commands are only applicable between CPS and CPE. 100 motion commands are possible to set between the two commands.

  (1) MOV command (4) ARC command
  (2) MOVM command (5) OUT command
  (3) CIR command (6) SPD command

Example: "CPS U1 S250"

◊ Start continuous path operation of Unit1. Motion speed is 250 mm/sec.

END End of program



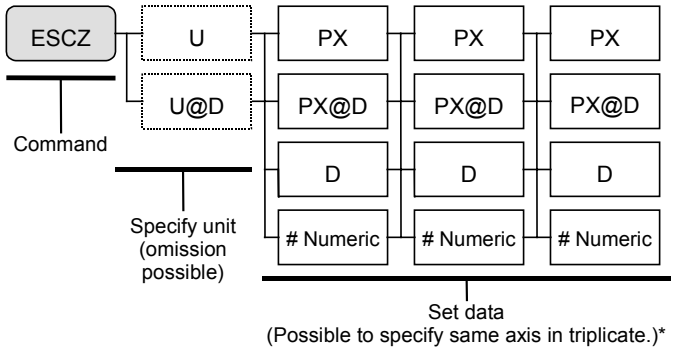
- This is the command to declare the end of program. Be sure to put the command on the end of the program. Otherwise, "program error alarm" areises.
- Execution of end command resets the contents of data registers and the flags that are resulted by CMP command.
- Condition of end
  None--------- Finishes all sequences immediately.
  "CSTP" ----- Makes all sequence cycle stop and finishes program.

Example "END CSTP"

◊ Makes all sequence cycle stop and finishes the programmed operation.

ESCR Reserved (Do not use.)

ESCZ Setting off-limits boundary of Z axis unit [Multi-axis controller]



- This is to specify off-limits boundary and turnout position for evading motion of Z axis unit in motion command MOV. (Evading motion is an arch motion.)
- Three kinds of data, from left to right, the lower off-limits boundary, the upper off-limits boundary and turnout position must be set.
- If the off-limits boundary data and turnout position are not specified in the initial setting, this command is not valid. If it is necessary to change the off-limits boundary and turnout position while programming, the initial setting shall be changed as described below.
  Off-limits boundary: Data must be the effective numeric numbers.
  Turnout position : Set data other than OFF.
  Refer to "9.3.4. Parameters for Position and Coordinates" for details.

Example 1: "ESCZ U1 #0000.00 #0100.00 #0120.00"

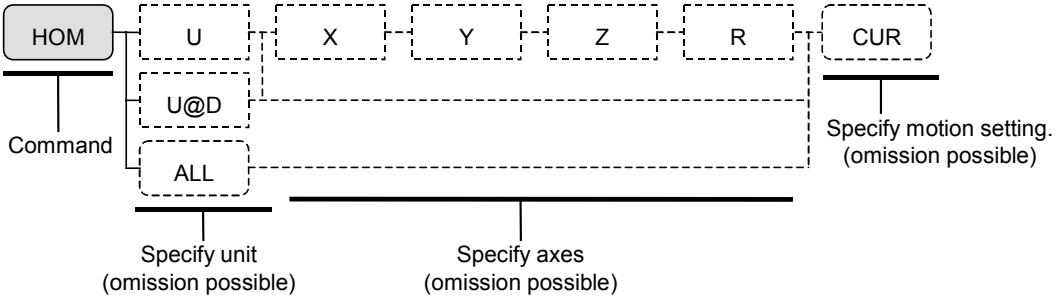
◊ Specify the off-limits boundary for Z axis unit of Unit 1. Set the lower off-limits boundary to 0000.00 mm and the upper off-limits boundary to 100.00 mm. Turnout position is 120 mm.

Example 2: "ESCZ U@D001 #1.23 PX0001 PX0002"

◊ When D001 is 1, X axis coordinate of P0001 is 200 mm and X axis coordinate of P0002 is 250 mm, evading of Z axis of Unit 1 is set as below.

Lower off-limits boundary : 1.23 mm
Upper off-limits boundary : 200.00 mm
Turnout position : 250 mm

HOM Start home return



- Use this command when the home return is necessary in the middle of programmed operation.
- Motion setting
  "ALL"--- All axis units execute the home return. (Default is the same.)
  "CUR" -- Current position is set to the home position without executing the home return.
- Refer to "9.3.2. Parameters for Home Return Operation" for Home return of each axis.

Example 1: "HOM"

◊ Start the home return.

Example 2: "HOM CUR"

◊ Set current position to the home position.

Example 3: "HOM U3" (in case of a multi-axis combination)

◊ Start Home return of all axes in Unit 3.

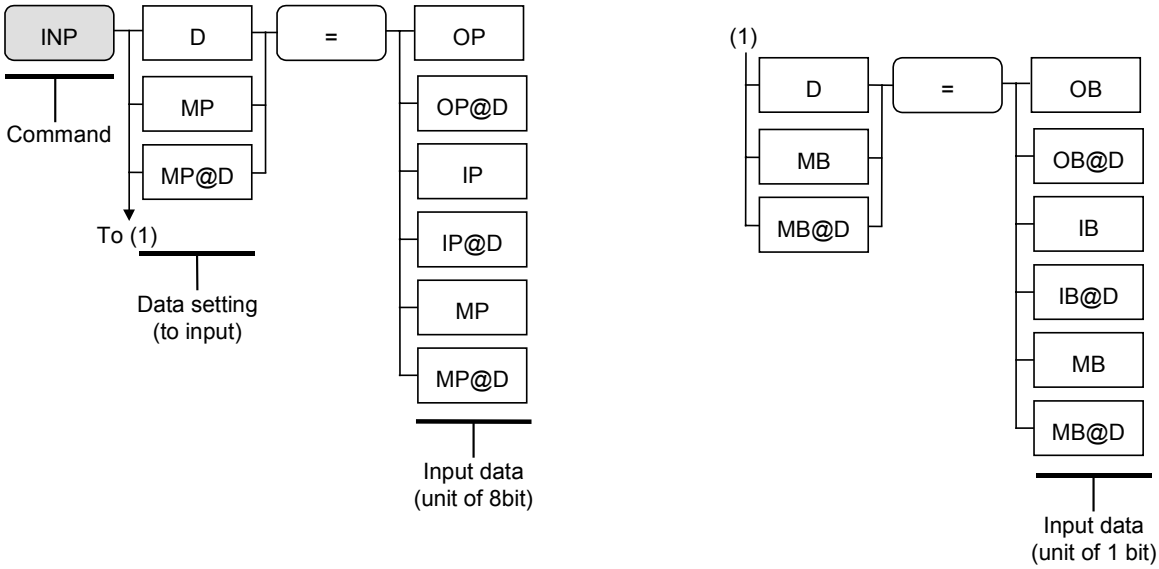
Example 4: "HOM U3 R CUR" (in case of a multi-axis combination)

◊ Only current position of R axis of Unit 3 is set to the home position.

Example 5 : "HOM ALL CUR" (in case of a multi-axis combination)

◊ Current positions of all axes of all units are set to the home position.

INP Input from port



- This is to input status of specified port to the data register or the imaginary I/O port.
- When input to a data register, the input data is recognized as a pattern of binary number without sign and is converted into integers of 0 to 255.
- When input to a imaginary I/O port, the input data keeps the same pattern as actual input / output.
- You can select either 8 bit unit or 1unit for port setting.
- You can set the general output and control I/O besides the general input.

● Correspondence of data to the signals of output port are shown in the table below.

Input port		EXT.I/O Connector pin	Name of connectors
8 bit unit	1 bit unit		
IP10	IB100	IN1	P1-EXT.I/O (standard)
	IB101	IN2	
	IB102	IN3	
	IB103	IN4	
	IB104	IN5	
	IB105	IN6	
	IB106	IN7	
	IB107	IN8	
IP11	IB110	IN9	
	IB111	IN10	
	IB112	IN11	
	IB113	IN12	
	IB114	IN13	
	IB115	IN14	
	IB116	IN15	
	IB117	IN16	
IP12	IB120	IN1	P2-EXT.I/O (optional)
	•	•	
	•	•	
	•	•	
	IB127	IN8	
IP13	IB130	IN9	
	•	•	
	•	•	
	•	•	
	IB137	IN16	
IP14	IB140	IN1	P3-EXT.I/O (optional: for multi-axis controller only)
	•	•	
	•	•	
	•	•	
	IB147	IN8	
IP15	IB150	IN9	
	•	•	
	•	•	
	•	•	
	IB157	IN16	
IP16	IB160	IN1	P4-EXT.I/O (optional: for multi-axis controller only)
	•	•	
	•	•	
	•	•	
	IB167	IN8	
IP17	IB170	IN9	
	•	•	
	•	•	
	•	•	
	IB177	IN16	

Example 1: "INP D012 = IP11"

◊ Substitute the status of general input port IP11 to a data register D012.

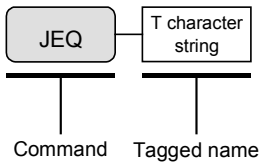
Example 2: "INP D020 = IB117"

◊ Substitute the status of general input port IN16 to a data register D020.

INT Reserved (Do not use.)

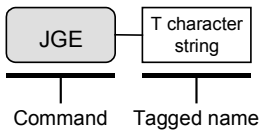
IRET Reserved (Do not use.)

JEQ Conditional jump (=)



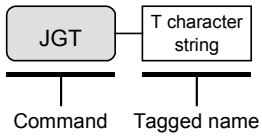
- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [=].
- When the condition is not met, the program does not make a jump and executes the next step.
- Tagged name: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

JGE Conditional jump (≥)



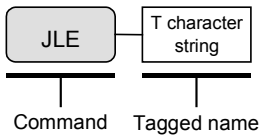
- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [≥].
- When the condition is not met, the program does not make a jump and executes the next step.
- Tagged name: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

JGT Conditional jump (>)



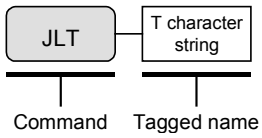
- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [>].
- When the condition is not met, the program does not make a jump and executes the next step.
- Tagged name: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

JLE Conditional jump (≤)



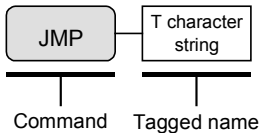
- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [≤].
- When the condition is not met, the program does not make a jump and executes the next step.
- Tagged name: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

JLT Conditional jump (<)



- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [<].
- When the condition is not met, the program does not make a jump and executes the next step.
- Tagged: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

JMP Unconditional jump

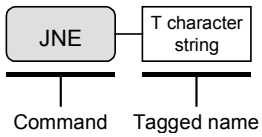


- The program jumps to a step with specified tagged name.
- Jumps to the marked position by TAG command and changes sequence. Character string of the tag shall be less than 8 letters of alphabets and numbers.
  (Under bar is not included in the characters.)

Example: "JMP _START"

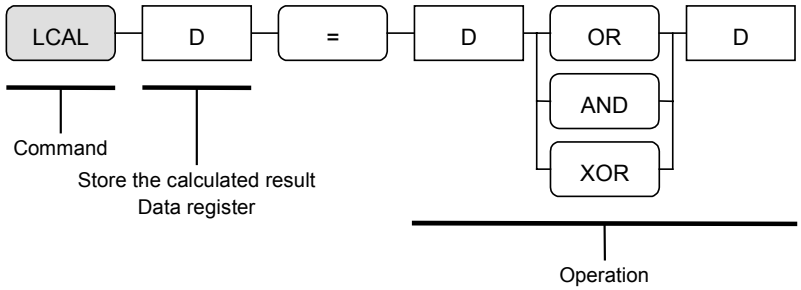
◊ Jump to a step tagged with "_START".

JNE Conditional jump (≠)



- The program jumps to a step with specified tagged name in accordance with the result of comparison by CMP command.
- Jumps when the result of comparison is [≠].
- When the condition is not met, the program does not jump and executes the next step.
- Tagged name: Set characters less than 8 using alphabets and numbers.
  (Under bar is not included in the characters.)

LCAL Logical operation of data



- This is to execute operation between specified data and substitute the result for a data register during programmed operation.

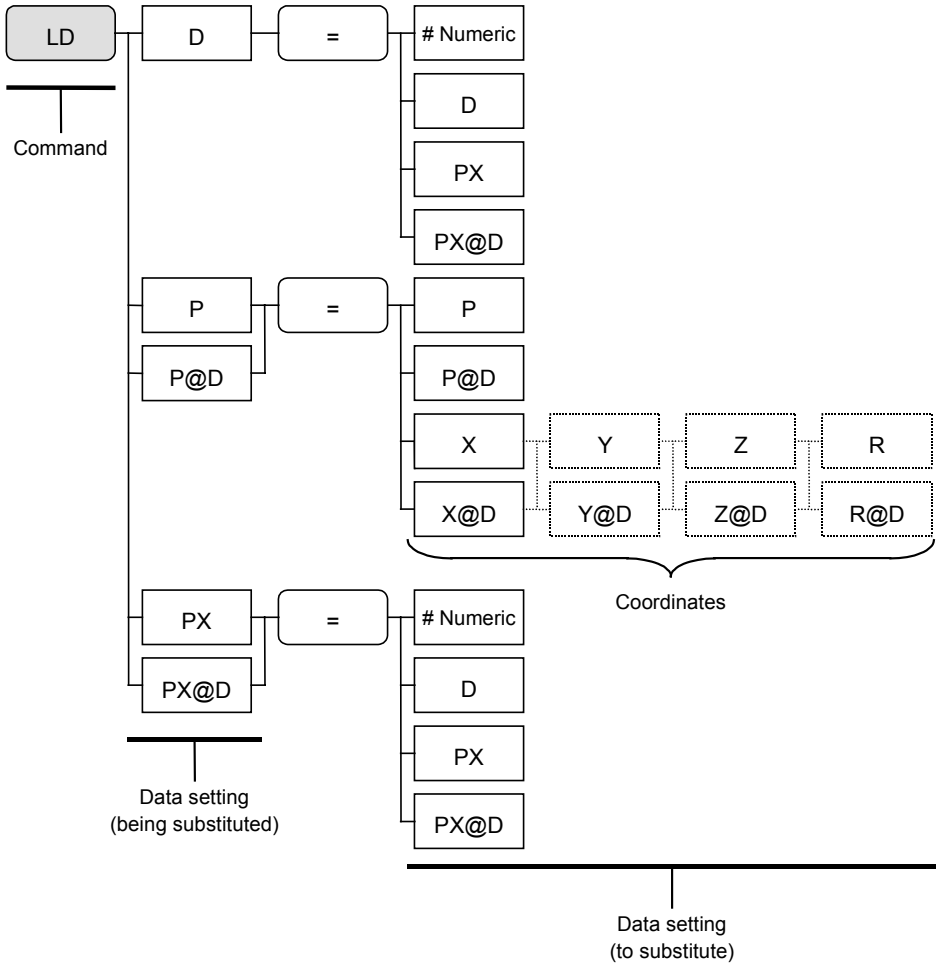
- Operation
  "OR" -------- Operates logical sum.
  "AND" ------ Operates logical multiplication.
  "XOR" ------ Operates exclusive OR.

- The contents of data register shall be the integer 0 to 255. Contents of data register are converted into 8 bit binary number, and the logical operation is executed to each bit.

Example: "LCAL D003 = D001 XOR D002"

◊ Operates 'exclusive OR' between data registers D001 and D002 and substitute the result for D003.

LD Data setting



- This command is to substitute the data and the contents of a data register for a specified data register in the program.

Example 1: "LD D001 = #1"

◊ Substitute numeric number 1 for data register D001.

Example 2: "LD P0001 = P0000"

◊ Substitute the point register P0000 for the point register P0001.

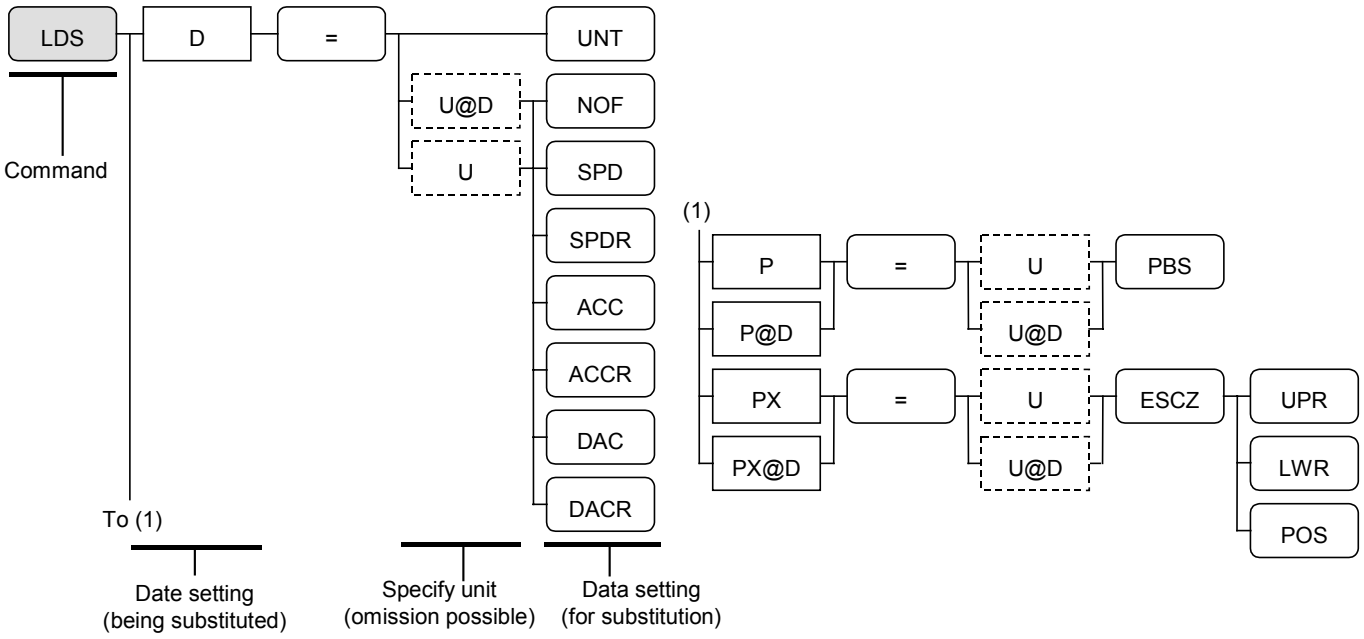
Example 3: "LD P@D001" = P0000"

◊ Substitute the contents of D002 as a X coordinate for a point register of which contents are inputted in the point register D001.

Example 4: "LD P@D001 = X@D002 Y0130.12" (in case of a multi-axis combination)

◊ Substitute the contents of D002 for X axis and 130.12 for Y axis, which are in the point register that shows the contents of data register D001.

LDS Read out motion data setting



- Reads out the motion data setting and substitute for the specified data.

- Substitution setting

  UNT ----- Setting motion unit
  NOF ----- Offset setting, point register number
  SPD ------ Setting of motion speed (direct setting)
  SPDR ---- Setting of motion speed (percentage)
  ACC ----- Setting of motion acceleration (direct setting)
  ACCR --- Setting of motion acceleration (percentage)
  DAC ----- Setting of motion deceleration (direct setting)
  DACR --- Setting of motion deceleration (percentage)
  PBS ------ Setting of working reference point position
  ESCZ ---- Setting of off-limits boundary of Z axis

  Read out upper boundary by UPR, lower boundary by LWR and turnout position by POS.

Note: If turnout position (ESCZ POS) is not specified on the system state setting, readout will be 0000.00.

Example 1: "LDS D000" = SPD

◊ Substitute the setting value of motion speed for the data register D000.

Example 2: "LDS D000 = UNT"

◊ Substitute setting of motion unit for data register D000.

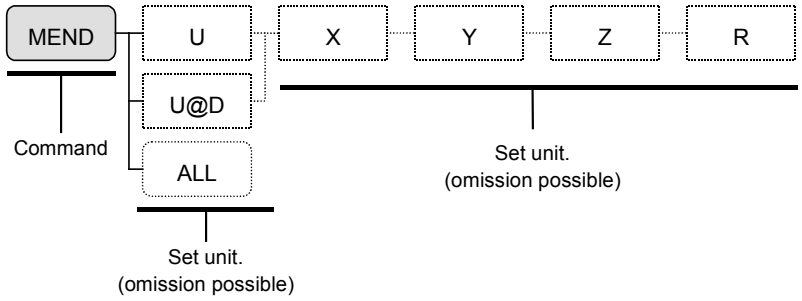
Example 3: "LDS D001 = U@D002 ACC"

◊ Substitute the acceleration setting of unit, which is shown by data register D002, for data register D001.

Example 4: "LDS PX0001 = U1 ESCZ POS" (in case of a multi-axis combination)

◊ Substitute off-limits area of Z axis of Unit 1 for X axis setting of point register P0001.

MEND Wait for end of motion



- When executing multitask operation, this command is to wait the start until the motion unit stops.
- In case of multi-axis combination, this command can be set to a unit or an axis unit. When MEND is set to all axis units, the program does not go to the next step until they complete motion.
- If hold or cycle stop of operation occurs during waiting for end of motion, the command will be temporary interrupted.
- "ALL": Wait until all axis units complete their motion.

Example 1: "MEND"

◊ Wait until specified unit completes motion.

Example 2: "MEND U1"

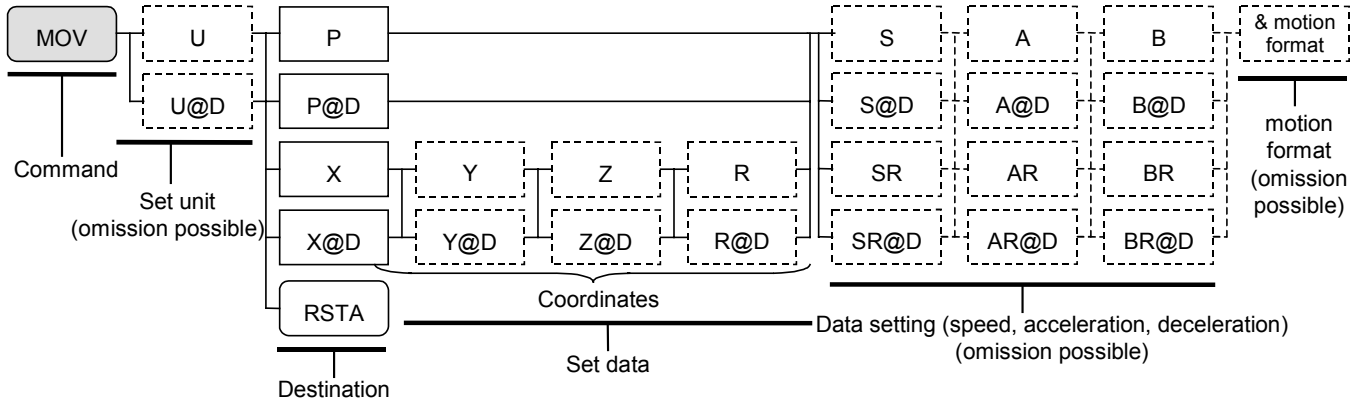
◊ Wait for the end of motion of Unit1.

Example 3: "MEND U1 Y R" (in case of a multi-axis combination)

◊ Wait for the end of motion of Y and R axis units of Unit1.

MOV Start linear interpolation

(For a single axis controller this command is regarded as "start linear motion.")



- This command starts a linear interpolation. In case of a single axis controller the command is for a linear motion, while the command is for up to three axes linear interpolation for a multi-axis controller.

  Note: An intended linear interpolation cannot be executed or its accuracy may be degraded if the coordinate setting does not meet the requirements for linear interpolation.

- Use the data specified by SPD and TYP normally for motion speed, acceleration / deceleration and motion format.

- It is possible to change the motion speed, the acceleration and motion formats to the MOV command. Add the parameters after the setting of point register. These settings are valid in the step only.

- It is possible to set only the motion codes that are necessary to change when specifying a motion format. (Refer to "15.2.7.1. 12 .")

- When "W" (normal processing command) is set as a motion format, the program does not go to the next step until the current step completes. When "P" (parallel processing command) is set, the program operates the subsequent steps in parallel while executing current step. However, if a motion command is programmed in subsequent steps to an axis unit currently in motion, an alarm arises. In such a case, set MEND to wait for completion of motion, or set MSTS command to confirm the motion not to duplicate the operating axis unit.

- "RSTA" ----- Use the command to return to the position just before the programmed operation was interrupted after the power is turned on again. Refer to description of "RSTA" in this chapter.

- When NOF command is set, be careful for changes of point register numbers.

- When PBS command is set, be careful for changes of point register numbers.

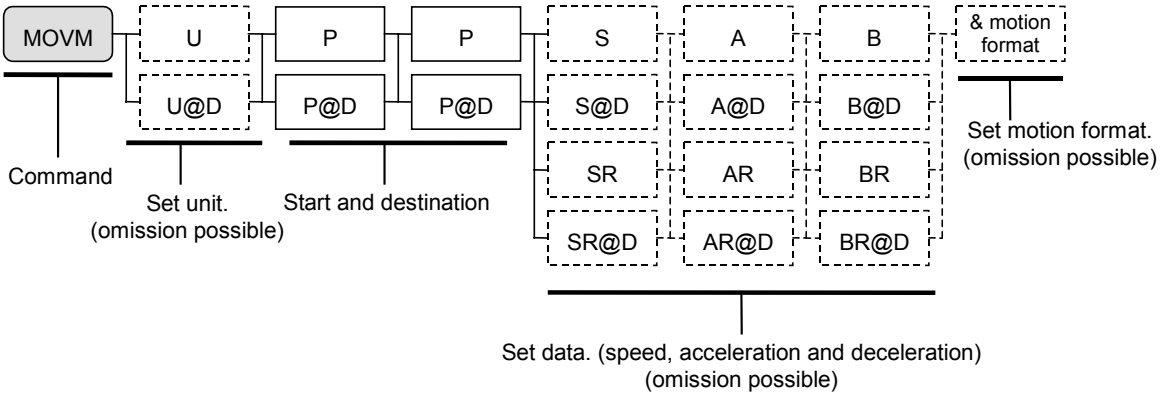
Example 1: "MOV P0001"

◊ Move to the point of which coordinates are indicated in a point register P0001.

Example 2: "MOV U1 P0001"

◊ Move Unit 1 to the point which is indicated by point register P0001.

*MOVM Linear interpolation (multi-points)** ** Linear motion (multi-point) in case of a single controller.*



- Specify continuous points of point number and start linear interpolation (multi-point). In case of a single axis system, it is linear motion. For a multi-axis combination, it is an interpolation with up to three axes.

  Note: An intended linear interpolation cannot be executed or its accuracy may be degraded if the coordinate setting does not meet the requirements for linear interpolation.

- Use specified data by SPD and TYP commands normally for motion speed, acceleration / deceleration and motion format.

- It is possible to change the motion speed, the acceleration and motion formats to the MOV command. Add the parameters after the setting of point register. These settings are valid in the step only.

- It is possible to set only the motion codes that are necessary to change when specifying a motion format. (Refer to "15.2.7.1. 12.")

- When "W" (normal processing command) is set as a motion format, the program does not go to the next step until the current step completes. When "P" (parallel processing command) is set, the program operates the subsequent steps in parallel while executing current step. However, if a motion command is programmed in subsequent steps to an axis unit currently in motion, an alarm arises. In such a case, set MEND to wait for completion of motion, or set MSTS command to confirm the motion not to duplicate the operating axis unit.

- When NOF command is set, be careful for changes of point register numbers.

- When PBS command is set, be careful for changes of point register numbers.

Example 1: "MOVM P0100 P0120 &P"

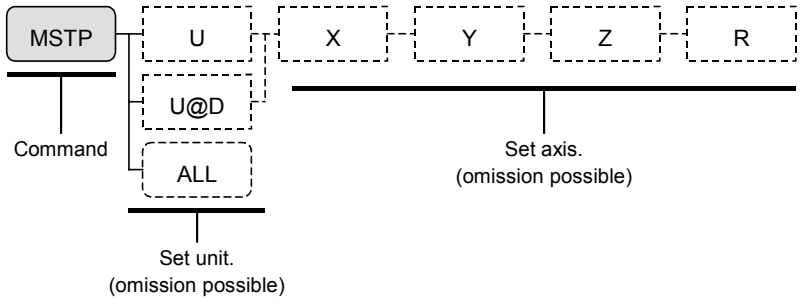
◊ Start a linear interpolation (multi-point) as a parallel processing command. P0100 as the starting point and P0120 as the destination. Passing points are P010, P0102, P0103 and P0119 as the last.

Example 2: "MOVM U1 P0100 P0120 &P"

◊ Start continuous linear interpolation of Unit1 as a parallel processing operation. Set the point register P0100 as the starting point and P0120 as the destination. Passing points are P010, P0102, P0103 and P0104 as the last.

MSPD Reserved (Do not use.)

MSTP Motion stop



- This command is to interrupt or stop motion unit in the middle of an operation, such as a multitask operation.
- This command does not work for the home return.
- "ALL": Stops the motions of all unit and axis units.
- In case of a multi-axis combination, when an axis unit which is involved in a multi-axis interpolation is interrupted, the other axis units stop as well.

Example 1: "MSTP"

◊ Stops motion of an operating unit.

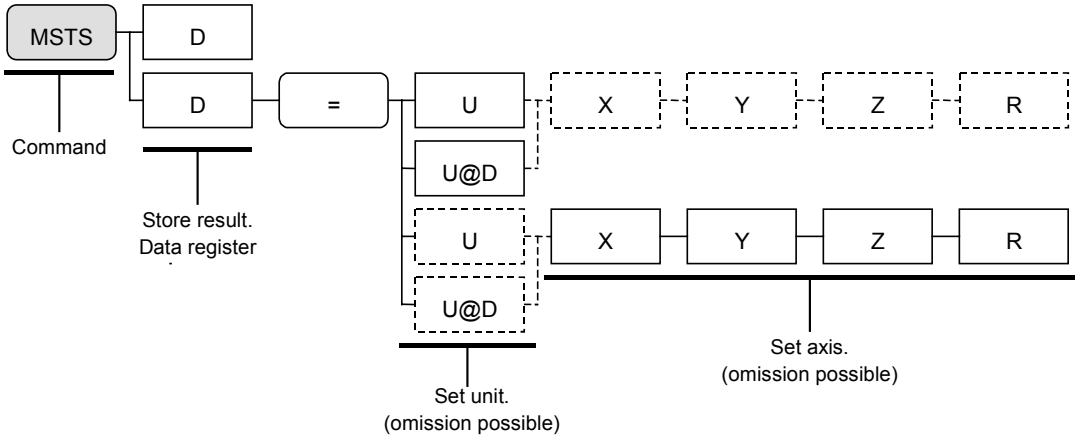
Example 2: "MSTP XY" (in case of a multi-axis combination)

◊ Stops X and Y axis of motion unit specified by UNT command.

Example 3: "MSTP ALL" (in case of a multi-axis combination)

◊ Stops motion of all axes.

MSTS Confirm motion state



- This is to confirm motion state of unit during multitask operation.
- Confirmation is performed on a unit or a specified axis unit.
- If all settings after ' = ' are omitted, the motion state of a unit which are currently specified by UNT command will be substituted for a data register.
- The following data will be stored to a data resister in accordance with motion state.
  0 ------ Stopping
  1 ------ In motion
  2 ------ Reserved

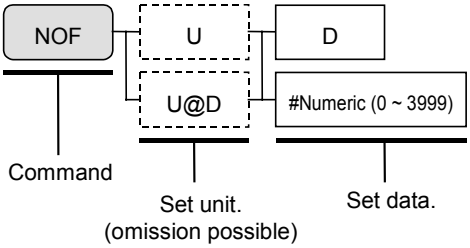
Example 1: "MSTS D001"

◊ Substitute state of motion of a unit, which is currently specified by UNT command, for the data register D001.

Example 2: "MSTS D023 = U1"

◊ Substitute motion state of Unit 1 for data register D023.

NOF Offset of point register number



- It is possible to set an offset to a point number for a command to generate pulses that uses the point register such as MOV command. Actual point number will be the sum of the number of specified point and the setting data of NOF.

- The offset by NOF command are effective for the following commands.
  Single axis system : MOV and MOVM commands
  Multi-axis combination : MOV, MOVM, CIR and ARV commands

- Set NOF command to "0" (zero) again to clear the offset that has been set in the same program.

Example 1: "NOF D000"

◊ Contents of data register D000 is set as the offset of point number.

* If the setting of data register D000 is 100, the actual point number to be use shall be P0200 when P0100 is specified by MOV command.

Example 2: "NOF #1"

◊ Set 1 as the offset of point number.

* If the setting of the data register D000 is 1 (one), the actual point register to be used is P0101 when P0100 is specified for Unit 1 by MOV command.

Example 3: "NOF U1 D000"

◊ Set the content of data register D000 to the offset of point number of Unit 1.

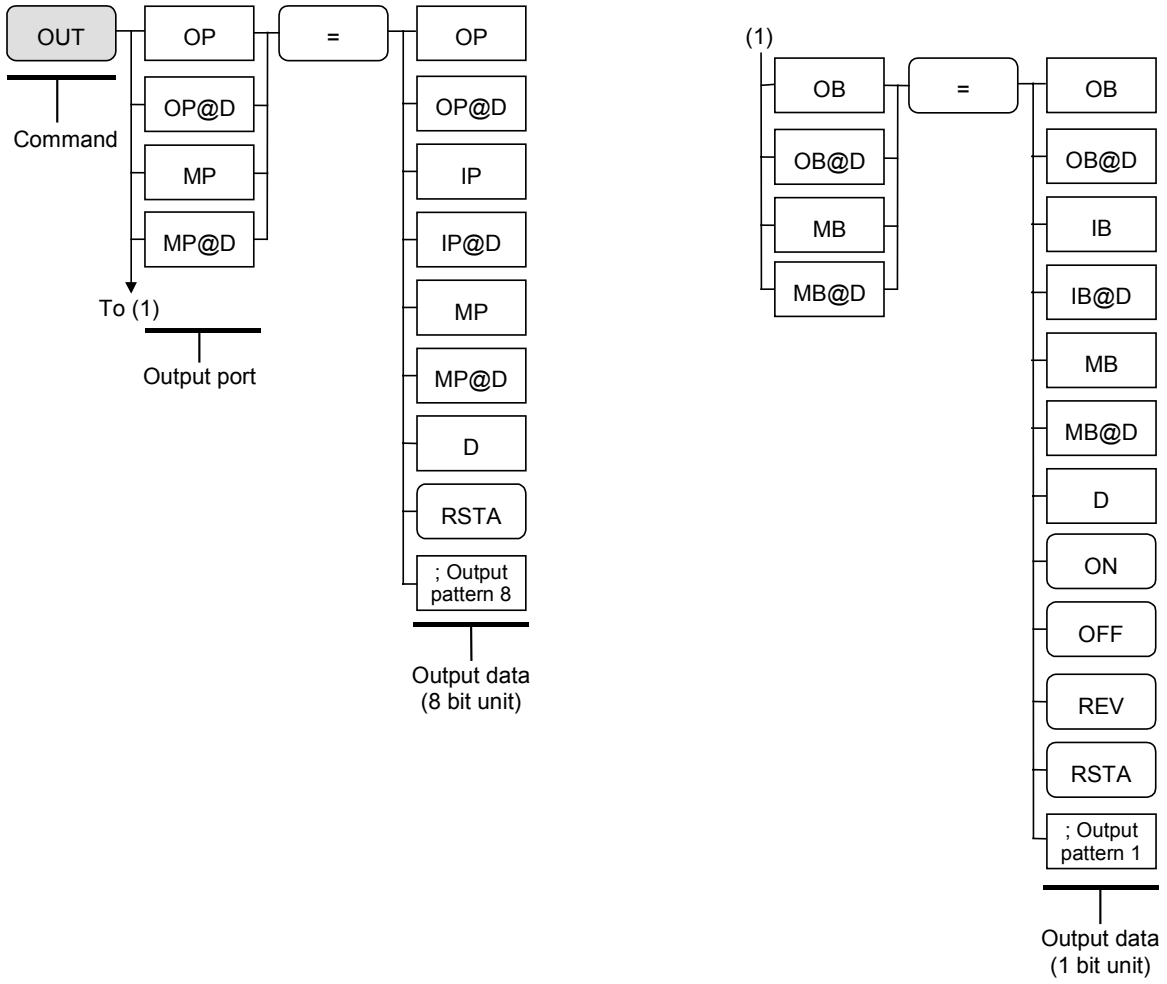
* The actual point number to be used will be P0200 when the content of D000 is 1 (one) and P0100 is specified by MOV command.

Example 4: "NOF U@D001 #1"

◊ Point number offset is sat to 1 (one) for a unit indicated by D001.

* The actual point register to be used will be P 0101 when MOV command D001 is 1 (one) and MOV command specifies P0100 for a unit 1.

Out *Output to port*



- This is to output data from general output (EXT. I/O) or hold the data to the imaginary input / output.
- Setting range of data register shall be the integers between 0 ~ 255 when outputs the contents of data registers.
- Output port may be selected in either 8 bit or one bit unit.
- Description of output data
  "ON" ------- Sets output state on.
  "Off" ------- Sets output state off.
  "REV"------ Reverses output state.
  "RSTA"---- Sets output state in "stop operation."
  (Effective while processing initialization by RSTA command.)
- ; output pattern 8 ------ Output pattern in 8 bit. (Refer to "15.2.7.1. [8].")
- ; output pattern 1 ------ Output pattern in 1 bit. (0 is off and 1 is on.)

● Correspondence of the data to the signals of output port are shown in the table below.

Name of output		EXT.I/O connector pin	Name of connector
8 bit unit	1 bit unit		
OP10	OB100	OUT1	P1-EXT.I/O (standard)
	OB101	OUT2	
	OB102	OUT3	
	OB103	OUT4	
	OB104	OUT5	
	OB105	OUT6	
	OB106	OUT7	
	OB107	OUT8	
OP11	OB110	OUT9	
	OB111	OUT10	
	OB112	OUT11	
	OB113	OUT12	
	OB114	OUT13	
	OB115	OUT14	
	OB116	OUT15	
	OB117	OUT16	
OP12	OB120	OUT1	P2-EXT.I/O (optional)
	•	•	
	•	•	
	•	•	
	OB127	OUT8	
OP13	OB130	OUT9	
	•	•	
	•	•	
	•	•	
	OB137	OUT16	
OP14	OB140	OUT1	P3-EXT.I/O (optional, multi-axis only)
	•	•	
	•	•	
	•	•	
	OB147	OUT8	
OP15	OB150	OUT9	
	•	•	
	•	•	
	•	•	
	OB157	OUT16	
OP16	OB160	OUT1	P4-EXT.I/O (optional, multi-axis only)
	•	•	
	•	•	
	•	•	
	OB167	OUT8	
OP17	OB170	OUT9	
	•	•	
	•	•	
	•	•	
	OB177	OUT16	

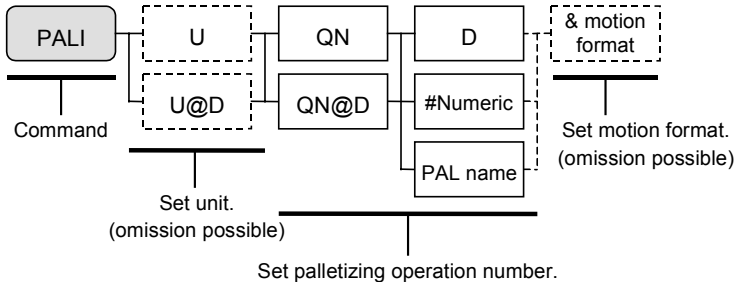
Example 1: "OUT OP1 = ; 00010010"

◊ Set the pattern of general output port OP11 to "00010010." (1 : ON, 0 : OFF)

Example 2: "OUT OB117 = ; 1"

◊ Turn on a general output port OUT16.

PALI Initialize: palletizing operation [Multi-axis controller]

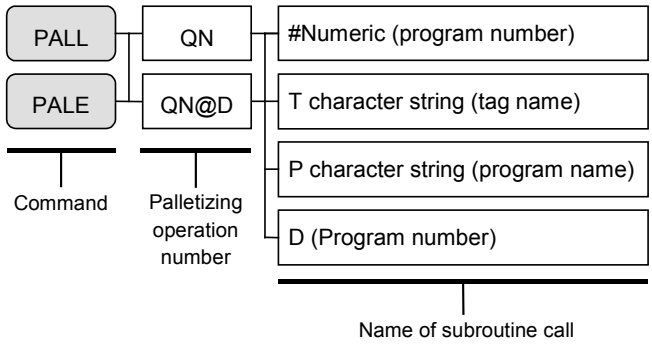


- This command is to initialize palletizing operation.
- It correlates a specific palletizing data with a palletizing operation number to make it usable for a specific motion unit.
- Palletizing data must be made separately. (Refer to "16.1. Palletizing.")
- Only motion format of &A or &I is possible to set.
- The way of handling the palletizing position data changes in accordance with motion format. When "&A" is set, coordinates of position data is absolute position, while the position data is relative coordinates from the current position when &I is specified.
- If PALI command specifies the same palletizing operation number again, which is currently operating, the new palletizing information overwrites the former information of the palletizing operation.

Example: "PALI U1 QN00 #0000"

◊ This command is to correlate palletizing operation number 00 of Unit 1 with palletizing data 0000.

PALL, PALE Palletizing operation : call subroutine [Multi-axis controller]

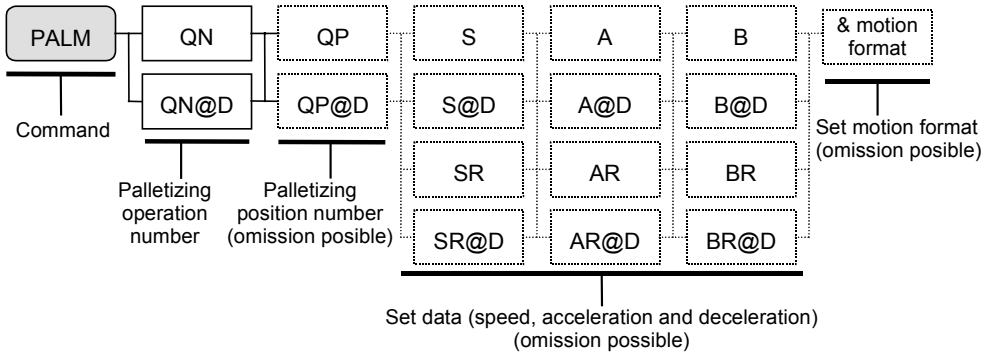


- This command is to call subroutine while executing a palletizing operation.
- Conditions of calling subroutine of PALL and PALE commands are different as shown below.
  PALL command: Call subroutine after completion of palletizing operation.
  (After completed all positioning to the pallet positions).
  PALE command: Call subroutine during palletizing operation.
- This is used to call the subroutine such as changing or loading a pallet.
- A subroutine to change a pallet ends by RET command and the program returns to the next step of PALL and PALE. The palletizing position number is reset when the next subroutine is called.

Example: "PALL QN00 _LODTAG"

◊ When the palletizing of operation number 00 is completed, call local subroutine which starts a step with the tag named "_LODTAG."

PALM Palletizing operation: Positioning [Multi-axis controller]



- This command is to execute positioning to palletizing position of specified palletizing operation number.

- The intended palletizing position number may be selected in the automatic renewal style or the specific renewal style. When QP×××× is specified to a palletizing position number, it is the specific renewal style.
  Automatic renewal style------Motion to a position of palletizing position number that is automatically renewed from 0 (zero).
  Specific renewal style---------Motion to a position of specified palletizing position number.

- Palletizing position number increases one by one (increment) at an execution of PALM command of palletizing operation number in the same program. It starts from 0 (zero) in case of automatic renewal style, while it starts from the specified palletizing position number in case of specific renewal style.

- Palletizing position number may be changed by PALM command.

- Motion speed, acceleration, deceleration and motion format use normally the data specified by SPD and TYO commands.

- It is possible to change motion speed, acceleration and motion format in this command. Add the parameters after point register setting. This setting is only effective in a program step.

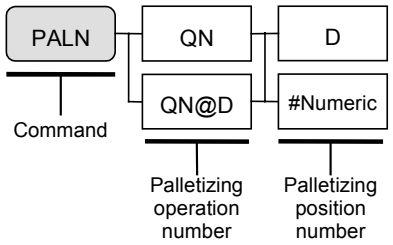
- Applicable motion formats are A, F/N, B/E and L/H only. Motion formats including I, PE and PH are not possible to set.

- If NOF command is set, note that it changes the point register number, and if PBS command is set, note that it changes the positions of positioning.

Example: "PALM QN00"

◊ Execute positioning of palletizing operation number 00.

PALN Palletizing operation: Change palletizing position number [Multi-axis controller]

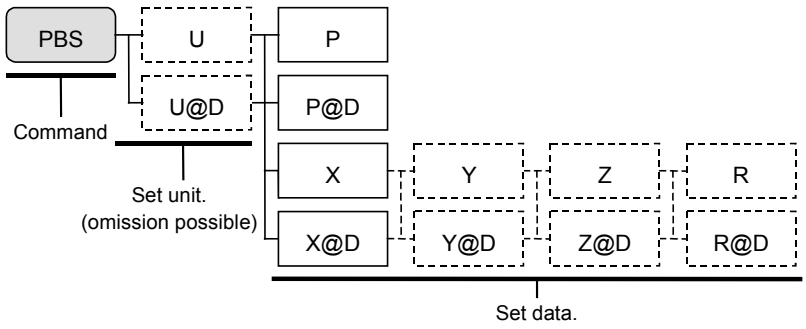


- Changes palletizing position number of specified palletizing operation number in palletizing.

Example: "PALN QN02 #33"

◊ Change position number of palletizing operation number 02 to position number 33.

PBS Set working reference point



- Sets a position of working reference point as an acting origin of coordinates in pulse generating command such as MOV.
- When PBS command executed in the absolute coordinate format, the coordinates of positioning points will take the reference point as the provisional Home position (acting origin).
- To terminate the setting of working reference point, set the command data to zero (0).

Example 1. "PBS P0100"

◊ Set the working reference point to the position indicated by point register P0100.

Example 2. "PBS P0100"

◊ Set the working reference point to the position on [X 100 mm]. Positioning point in the same palletizing program will be the sum of the coordinates of destination and X 100 mm thereafter.

Example 3: "PBS U1 P0100"

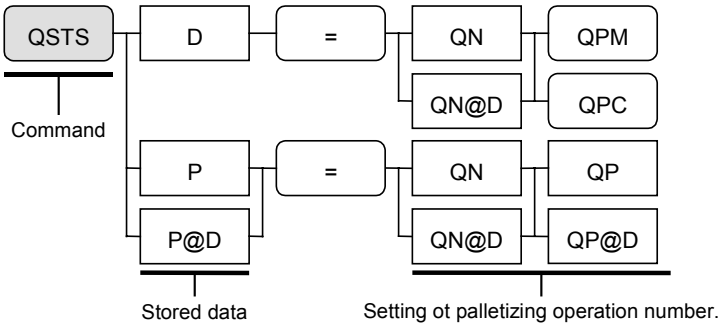
◊ Set the working reference point of Unit1 to the point shown in the point register P0100. After this setting, the coordinates of positioning points in the same program will be the addition of the data of absolute coordinates of destination and the data specified in P0100.

Example 4. "PBS U1 X0100.00 Y 0200.00" (in case of a multi-axis combination)

◊ Set the working reference point of unit 1 to the point on [X 100 mm, Y 100 mm]. The position points of the same palletizing program will be the points of which coordinates are the sum of the coordinates of destination and [X 100 mm].

PDSV Reserved (Do not use.)

QSTS Palletizing operation: Read out palletizing conditions [Multi-axis controller]



- This command is to read out palletizing conditions of specified palletizing operation number.
- Specify data registers as substitutional objectives to get numbers of positioning points and palletizing operation.
  QPM----- Read out total number of positioning points of a pallet.
  QPC------ Read out point number in a palletizing operation.

Example : "QSTS D000 = QN02 QPM"

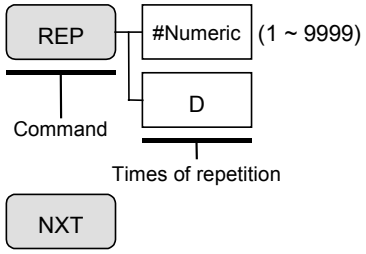
◊ Substitute the total number of positioning points of palletizing operation number 2 to data register D000.

- Specify point registers as substitutional objectives to get each palletizing position number on a pallet.

Example : "QSTS P0001 = QN02 QP0012"

◊ Substitute a coordinate data of palletizing position 12 to P0001.

REP ~ NXT Repeat steps



- Repeats the program steps that are set between REP and NXT for specified number of times. Repetitive times can be set between 1 to 9999.

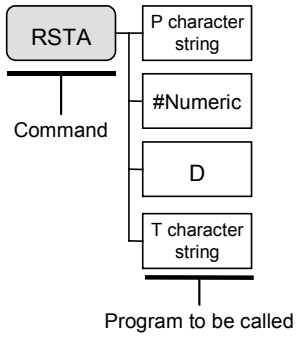
Example 1: "REP #30"

◊ Repeat the steps between REP and NXT for 30 times.

Example 2: "REP D001"

◊ Repeat the steps between REP and NXT for number of times set in data register D001.

RSTA Setting subroutine for recovery.



- This command is to specify the way of handling initialization for recovery when the power is turned on again after cycle stop of programmed operation.
- Set RSTA command on the top of the main program. When cycle stop occurs before execution of this command, this command is not valid for recovery. (The subroutine of initialization won't work.)
- RET command ends the subroutine of initialization.

Example: "RSTA $SUBPRG"

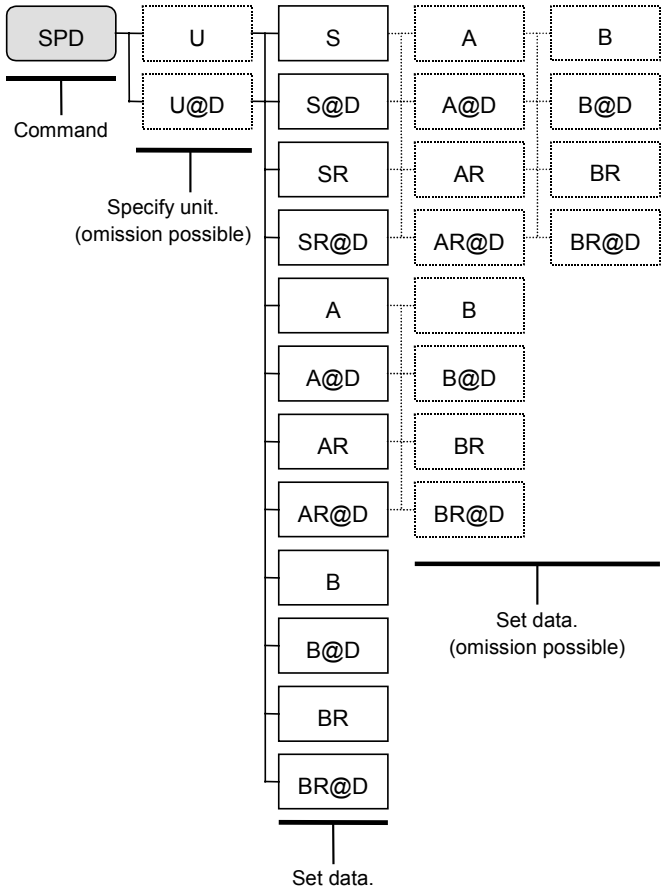
◊ After the power is turned on again to recover from cycle stop in a programmed operation, start after calling the program "SUBPRG" as the subroutine.

* In this initializing process, it is possible to use several special commands.

(1) MOV RSTA----------Move to the position at where the operation has stopped.

(2) OUT OP11 = RSTA -----Set the general output port OP11 to the same state when the operation stopped.

SPD Set motion speed, acceleration and deceleration

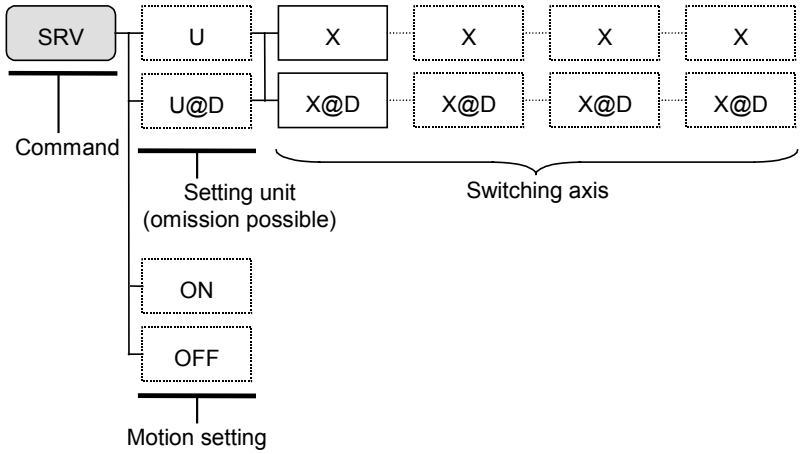


- This command sets motion speed, acceleration and deceleration of a specified unit.
- If motion speed, acceleration and deceleration won't be set to each motion command after set of SPD command in the same program, motions are controlled by this setting.
- Setting range of motion speed is 0.1 to 1200.0 mm /s. However, the actual maximum speed differs in the module main units. Refer to "19.1.2. Specifications."
- Setting range of acceleration is 0.1 to 35.0 m / s^2. However, set acceleration according to an actual load.

Example: "SPD S300"

◊ Set motion speed 300 mm / s to motion units.

SRV Switch servo on / off



- It can set the servo on and off in a program.
- SRV command is valid when following conditions are met.
  - (1) In external control mode: [SVON] of CN3 is ON.
  - (2) In remote control mode : Servo is on by SVON command.
  - (3) When operating through teaching box: Servo is on by the ON key.
- This command is used when switching Servo on/off for all axis units simultaneously.
  "ON" ---- Servo on all axis units simultaneously.
  "OFF" --- Servo off all axis units simultaneously.
- Setting to switch axis
  0 ---------- Servo on
  1 ---------- Servo off
- State of servo on/off does not change for an axis that is not set to switch axis in case of multi-axis combination.

Example 1: "SRV OFF"

◊ Servo of all axes are simultaneously turned off.

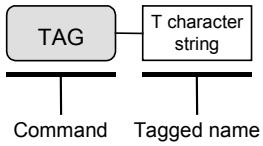
Example 2: "SRV X1"

◊ Servo is off.

Example 3: "SRV U1 X1 Y0" (in case of a multi-axis combination)

◊ Turn servo on of X axis and servo off of Y axis of Unit 1.

TAG Set tag

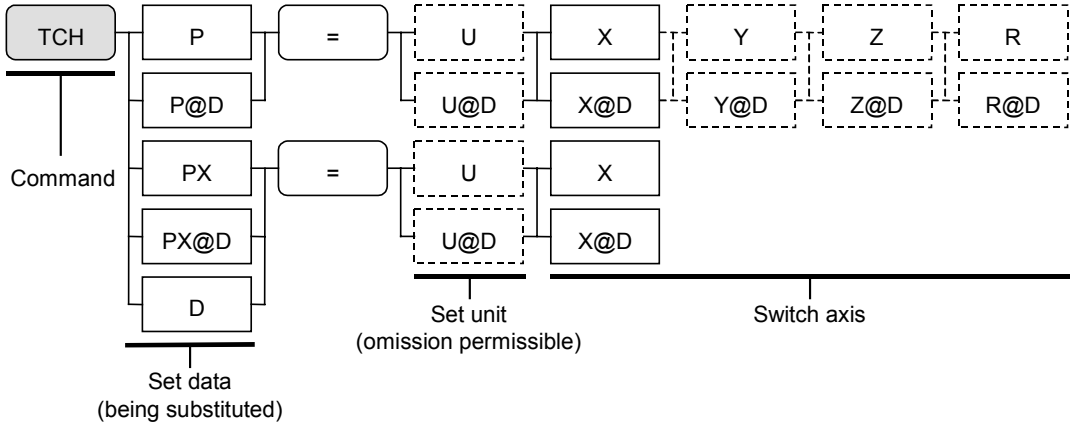


- Set a tagged name in a program.
- It is to set a mark to be used for a "jump" or a "call" command. Set the tag in 8 characters of alphabets and numbers. ["_" (under bar) is not included in the characters.] You cannot duplicate the same tag name in the same program.

Example: "TAG_START"

◊ Name the tag "_START."

TCH Substitute current coordinate data for a specified data.



- This is to set the current position data of specified unit to a specific point register or data register.
- Switch axis
  "0" ------- Sets no action (××××.××).
  "1" ------- Sets current position.
- Set only the coordinate data of specified axis unit, while the coordinate data of other axis units are not changed in case of a multi-axis combination.

Example 1: "TCH P0000 = U1 X1 Z0"

◊ Substitute current position of motion unit 1 for point register P0000. Apply current coordinate data of X axis and none for Z axis, (××××.××). No changes of coordinates of Y and R axes.

- Normally plural axis units are processed simultaneously in a multi-axis combination. However, when the data register is set as a substitution, processing shall be made to the axis units one by one.

Example 2: "TCH PY@D003 = X1"

◊ Substitute the current position of X axis of specified unit for Y coordinate of point register D003. Other axes remain same.

- Initialization of point register in a program is possible when switching axis is set to "0 (zero)" (no motion).
- In a single axis system only X axis may be specified. It does not specify the unit.

Example 3: "TCH P0000 = X1"

◊ Get current X coordinate data of point register P0000.

Example 4: "TCH P@D003"

◊ Set current position of X axis to a point register indicated by D003.

- You can initialize point registers if the axis switch is set to "0" (no action)

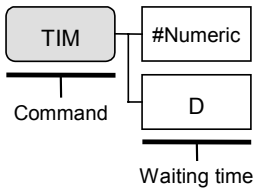
Example 5: "TCH P000 = X0" (in case of a single axis system)

◊ Initialize (××××.××) an X coordinate data in a point registered P000.

Example 6: "TCH P0000 = X0, Y0, Z0, R0 (in case of a single axis system)

◊ Initialize (××××.××) coordinate data of X to R axes in the point register P000.

TIM Set timer

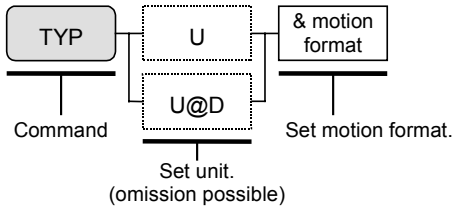


- Program proceeds to the next step after waiting for a time set by the command.
- Time setting range is 0.01 to 999.99 seconds.

Example: "TIM #1.23"

◊ Stop operation sequence for 1.23 seconds.

TYP Set motion format



- Set motion format of a specified motion unit.
- Motion formats set by TYP command will be effective in the commands without motion format, which come under the TYP command in the same program.
- Followings show the initial setting of motion format at the starting of a programmed operation. Set the format only that needs a change.
  In case of a single axis system : &ASFWL (absolute coordinates, smooth modified sine accel/decel. FIN on, normal processing and interpolation)
  In case of a multi-axis combination : &ASFWBL (absolute coordinates, smooth modified sine accel./decel. FIN on, normal processing, no turnout position, no linear interpolation)
- To make TYP command effective, one or more motion formats shall be set.
- Refer to "15.2.7.1. 12 Motion format of unit."
- Motion format must be set in the order of **A/I → T/S → F/N → W/P → B/E* → L/H**.
  A/I-------- Absolute / relative coordinates
  T/S ------- Fixed. Either setting of T or S gives smooth modified sine accel / deceleration
  F/N------- FIN on / off
  W/P ------ Normal processing / Parallel processing
  * B/E------- Turnout on / off
  L/H------- Linear interpolation on / off
  * B/E is not available for a single axis system.

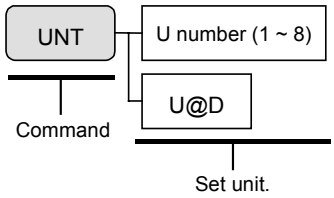
Example 1: "TYP &ASFPL" (in case of a single axis system)

◊ Set the motion formats as:
(1) Absolute coordinates (2) Smooth modified sine acceleration/deceleration
(3) FIN on (4) Normal processing
(5) Linear interpolation

Example 2: "TYP U3 & ASFPEL" (in case of a multi-axis combination)

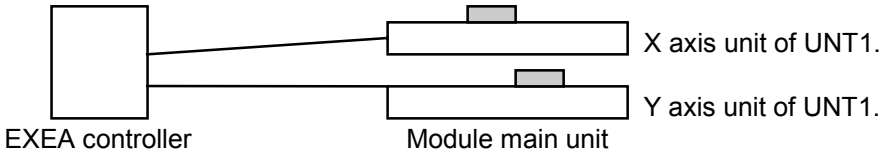
◊ Set the motion format of Unit 3 as :
(1) Absolute coordinates (2) Smooth modified sine acceleration/deceleration
(3) FIN on (4) Normal processing
(5) Turnout position (6) Interpolation

UNT Set motion unit [Multi-axis controller]

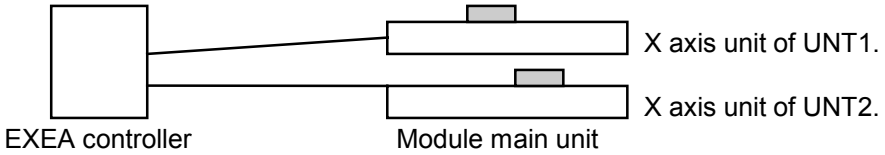


- Sets a motion unit number,
- When the unit setting is omitted to the commands which come after this command in the same program, the unit set by UNT is valid for the respective commands.
- When this command is not set, the number U1 is initially set.
- Unit number for each axis may be set in initial setting. Refer to "9.5. Parameters for Unit Setting."

(1) Initial setting



(2) Set X axis to UNT1 and UNT2.



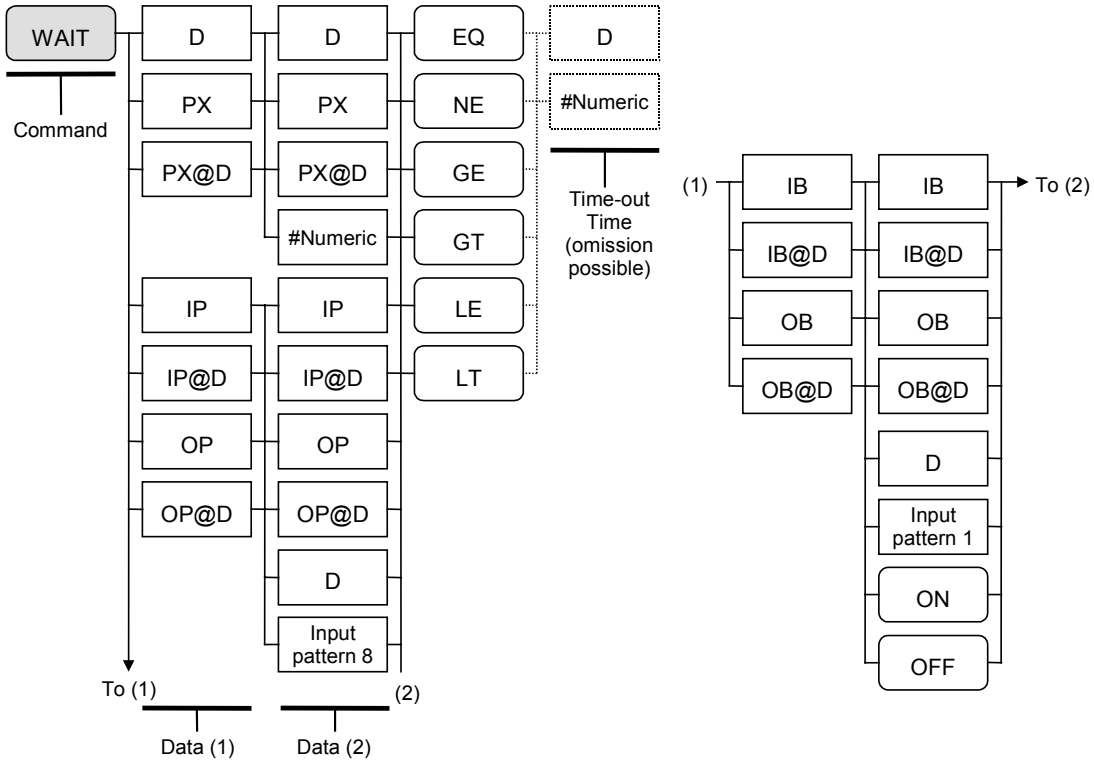
Example 1: "UNT U3"

◊ Set Unit 3 as a motion unit.

Example 2: "UNT U@D001"

◊ Set contents of data register D001 to a motion unit.

WAIT Wait



- This command holds to execute the next step in the middle of a programmed operation while comparison of two data (data1 and data2) meets a specified condition.

- When a time for time-out is set behind the condition, the sequence goes to next step when time expires even the condition is still maintained.

- It is possible to make a branch instruction just after the time-out is effective by setting JEQ command to the next step as the judging condition flag [=] is set when time expires. Setting range of time is 0.01 to 9999.99 seconds.

  * If the time-out is not set, the system interrupts the sequence when 'Hold' or 'Cycle stop' is requested during confirming the condition.

- Provided that right side is data 1 and left side is data 2:
  "EQ"----- Wait if (data 1 = data 2).
  "NE"----- Wait if (data 1 ≠ data2).
  "GE"----- Wait if (data 1 ≥ data 2).
  "GT"----- Wait if (data 1 > data 2).
  "LE" ----- Wait if (data 1 ≤ data 2).
  "LT" ----- Wait if (data 1 < data 2)

Example: "WAIT D003 #001.00 EQ #10.00"

◊ It waits while contents of data register D003 and 0001.00 is "EQ". Time-out setting is 10 seconds.

- Input pattern

  Input pattern 8 ------Input pattern is 8 bit. (Refer to 15.2.7.1. 8 .)

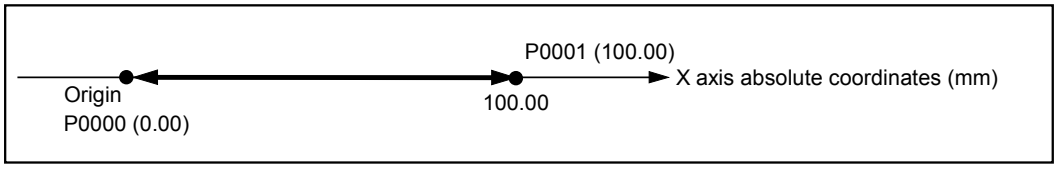
  Input pattern 1 ------Input pattern is 1 bit. (0 is off and 1 is on.)

15.3. Examples of Program

15.3.1. MOV Command : Single Axis System

Outline of motion: Move to a point (0.00, absolute coordinates) from current position (any position)
Move to (100.00, absolute coordinates) and return to (0.00, absolute coordinates)

Figure 15-68: Outline of motion



Example 1: Direct position setting

Advise: It is possible to program intuitively.

Program	Description
	<Initialize: Home return>
MOV X0000.00 S0100.0 A0.5 B0.5 &A	Move to 0.00 (mm) (absolute coordinates) from the current position (any position). (Speed: 100 mm/sec, acceleration / deceleration: 0.5 m/s^2, absolute coordinates)
	<Set parameters of linear motion.>
SPD S0600.0 A5.0 B5.0	Set motion speed and acceleration /deceleration to be used in MOV commands hereafter. (Speed: 600 mm/s, acceleration / deceleration: 5.0 m/s^2)
TYP &A	Set motion format for MOV commands to be used hereafter. (absolute coordinates)
	<Linear motion>
MOV X0100.00	Move X axis to 0100.00 (mm).
MOV X0000.00	Move X axis to 0000.00 (mm).
	<End of motion>
END CSTP	

Example 2: Set positions to point registers.

Advise: It makes easier to handle the coordinates data after the programming because the coordinates data becomes independent as the point register.

Set point data:

```
P000  X0000.00
P001  X0100.00
```

Program	Description
	<Initialize: Move to a point P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to 0.00 (mm) (absolute coordinates) from the current position (any position). (Speed: 100 mm/sec, acceleration / deceleration: 0.5 m/s^2, absolute coordinates)
	<Set parameters of linear motion.>
SPD S0600.0 A5.0 B5.0	Set motion speed and acceleration /deceleration to be used in MOV commands hereafter. (Speed: 600 mm/s, acceleration / deceleration: 5.0 m/s^2
TYP &A	Set motion format for MOV commands to be used hereafter. (absolute coordinates)
	<Linear motion>
MOV P0001	Move to the point P0001.
MOV P0000	Move to the point P0000.
	<End of motion>
END CSTP	

Example 3: Set positions to point registers in the relative coordinates.

Advise: It makes easier to handle the coordinate data after the programming because the coordinate data becomes independent as the point register.

Set point data:

```
P0000 X0000.00
P0001 X0100.00
```

Program	Description
	<Initialize: Move to a point P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to P0000 (absolute coordinates) from the current position (any position). (Speed: 100 mm/sec, acceleration / deceleration: 0.5 m/s^2, absolute coordinates)
	<Set parameters of linear motion.>
SPD S0600.0 A5.0 B5.0	Set motion speed and acceleration /deceleration to be used in MOV commands hereafter. (Speed: 600 mm/s, acceleration / deceleration: 5.0 m/s^2)
TYP &I	Set motion format for MOV commands to be used hereafter. (relative coordinates)
	<Linear motion>
MOV P0001	Move the distance in relative coordinates obtained from the point P0001 data.
MOV P0000 &A	Move to the point P0000.
	<End of motion>
END CSTP	

Example 4: Set positions to point registers, then set indirectly point registers to be used.

Advise: It will make the program for repetitive motion in less codes.
Also it will make easier to handle the coordinates data after programming as the coordinates data becomes independent as the point register.

Set point data:

```
P0000 X0000.00
P0001 X0100.00
```

Program	Description
	<Initialize: Move to a point P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to 0.00 (mm) (absolute coordinates) from the current position (any position). (Speed: 100 mm/sec, acceleration / deceleration: 0.5 m/s^2, absolute coordinates)
	<Set parameters of linear motion.>
SPD S0600.0 A5.0 B5.0	Set motion speed and acceleration /deceleration for to be used in MOV commands hereafter. (Speed: 600 mm/s, acceleration / deceleration: 5.0 m/s^2)
TYP &A	Set motion format for MOV commands to be used hereafter. (absolute coordinates)
LD D000 = #1	Substitute numeric value 1 for the data register D000.
	<Linear motion>
MOV P@D000	The motion follows the contents of point register number which is referred to the data register D000.
MOV P0000	Move to the point P0000.
	<End of motion>
END CSTP	

15.3.2. MOV Command in Two Axes Motion

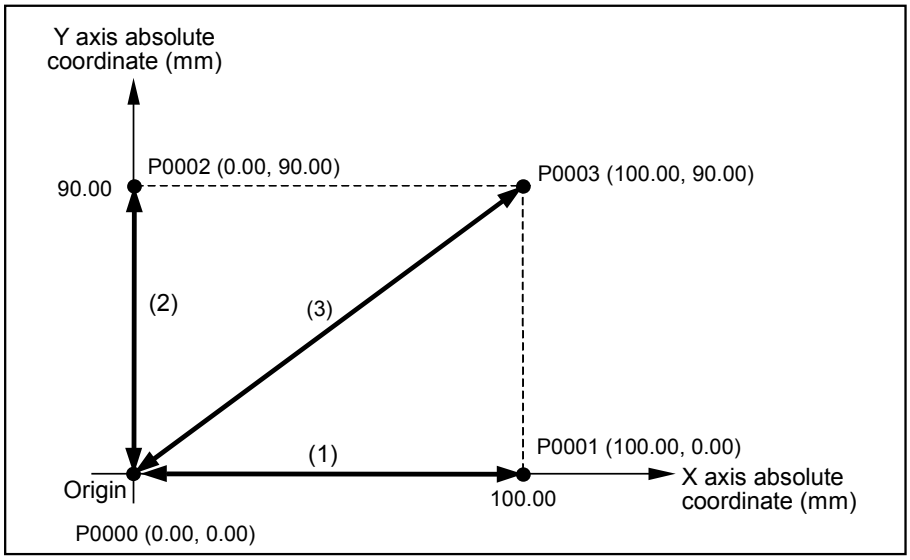
Outline of motion: Move to a point (0.00, 0.00, absolute coordinates) from current position (any position).

(1) Move to (100.00, 0.00, absolute coordinates) then return to (0.00, 0.00, absolute coordinates).

(2) Move to (0.00, 90.00, absolute coordinates), then return to (0.00, 0.00, absolute coordinates).

(3) Move to (100.00, 90.00, absolute coordinates), then move to (0.00, 0.00, absolute coordinates).

Figure 15-69: Outline of motion



Example 1: When specify the position directly

Advice: It is possible to program intuitively.

Program	Description
	<Initialize: Home return>
`MOV X0000.00 Y0000.00`	Move to (0, 0. Absolute coordinates) from current position (any coordinates position).
`S0100.0 A0.5 B0.5 &A`	(Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2)
	<Set parameters for interpolation.>
`SPD S0600.0 A5.0 B5.0`	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5.0 m/s^2)
`TYP &A`	Set the motion format for the interpolating commands hereafter. (Absolute coordinates)
	<Interpolation (1)>
`MOV X0100.00`	Move X axis only to 0100.00 (mm).
`MOV X0000.00`	Move X axis only to 0000.00 (mm).
	<Interpolation (2) >
`MOV Y0090.00`	Move Y axis only to 0090.00 (mm)
`MOV Y0000.00`	Move Y axis only to 0000.00 (mm)
	<Interpolation (3)>
`MOV X0100.00 Y0090.00`	Move to the position (100, 90) on absolute coordinates.
`MOV X0000.00 Y0000.00`	Move to the position (0, 0) on absolute coordinates.
	<End of motion.>
`END CSTP`	

Example 2: When specify the position by the point register

Advice: It makes easier to handle the coordinate data after the programming as the point register makes coordinate data independent.

Set point data:

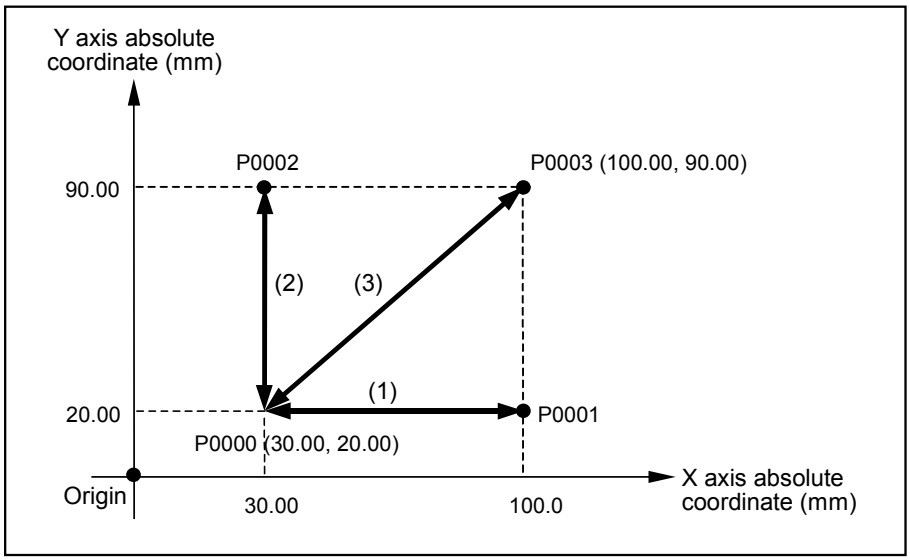
```
P0000 X0000.00 Y0000.00 Z××××.×× R××××.××
P0001 X0100.00 Y××××.×× Z××××.×× R××××.××
P0002 X××××.×× Y0090.00 Z××××.×× R××××.××
P0003 X0100.00 Y0090.00 Z××××.×× R××××.××
```

Program	Description
	<Initialize: Move to P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to P0000 from current position (any coordinates position). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2)
	<Set parameters for interpolation.>
SPD S0600.0 A5.0 B5.0	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
TYP &A	Set motion format for the interpolating commands hereafter. (Absolute coordinates)
	<Interpolation (1)>
MOV P0001	Move to the point P0001.
MOV P0000	Move to the point P0000.
	<Interpolation (2) >
MOV P0002	Move to the point P0002.
MOV P0000	Move to the point P0000.
	<Interpolation (3)>
MOV P0003	Move to the point P0003.
MOV P0000	Move to the point P0000.
	<End of motion.>
END CSTP	

- The motion changes as shown in Figure 15-70 if P0000 is changed to the position described below.

```
P0000 X0030.00 Y0020.00 Z××××.×× R××××.××
```

Figure 15-70: Outline of motion



* Coordinate data ××××.×× denotes that it does not change. (does not move.)
As Y coordinate of point P0001 is ××××.××, Y coordinate remains 20 mm, the coordinate of point (P0000) before make the motion.

Example 3: Specify positions by point register in relative coordinates format.

Advice: It makes easier to handle the coordinates data after the programming as the point register makes coordinates data independent.

Set point data:

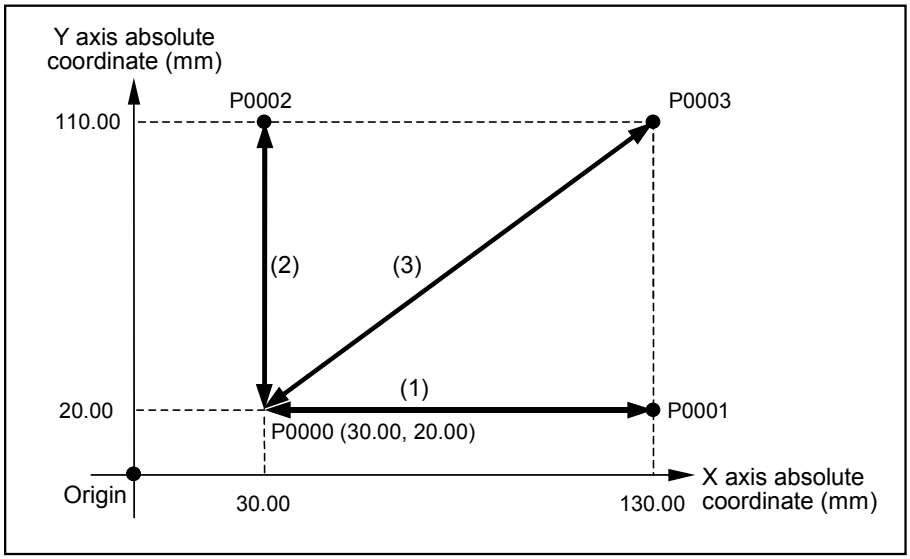
```
P0000 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0001 X0100.00 Yxxxx.xx Zxxxx.xx Rxxxx.xx
P0002 Xxxxx.xx Y0090.00 Zxxxx.xx Rxxxx.xx
P0003 X0100.00 Y0090.00 Zxxxx.xx Rxxxx.xx
```

Program	Description
	<Initialize: Move to P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to P0000 from current position (any coordinates position) (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, in absolute coordinates format)
	<Set parameters for interpolation.>
SPD S0600.0 A5.0 B5.0	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
TYP &I	Set motion format for the interpolating commands hereafter. (Relative coordinates)
	<Interpolation (1)>
MOV P0001	Move the distance in relative coordinate format obtained from the data of P0001.
MOV P0000 &A	Move to the point P0000.
	<Interpolation (2) >
MOV P0002	Move the distance in relative coordinate format obtained from the data of P0002.
MOV P0000 &A	Move to the point P0000.
	<Interpolation (3)>
MOV P0003	Move the distance in relative coordinate format obtained from the data of P0003.
MOV P0000 &A	Move to the point P0000.
	<End of motion.>
END CSTP	

- The motion changes as shown in Figure 15-71 if P0000 is changed to the position described below.

```
P0000 X0030.00 Y0020.00 Zxxxx.xx Rxxxx.xx
```

Figure 15-71: Outline of motion



* As MOV P0001 is a motion in relative coordinate format, X coordinate of P0001after motion is 130 mm, the addition of X coordinate 30.00 mm of current position P0000 and the X teaching coordinate 100.00 mm of point P0001.

Example 4: Specify the positions to point registers, then specify the using point register indirectly.

- Refer to "Figure 15-69: Outline of motion."

Advice: It will make the program for repetitive motion in less code, and make easier to handle the coordinates after the programming as the point register makes coordinates data independent.

Set point data:

```
P0000 X0000.00 Y0000.00 Z××××.×× R××××.××
P0001 X0100.00 Y××××.×× Z××××.×× R××××.××
P0002 X××××.×× Y0090.00 Z××××.×× R××××.××
P0003 X0100.00 Y0090.00 Z××××.×× R××××.××
```

Program	Description
	<Initialize: Move to P0000,>
`MOV P0000 S0100.0 A0.5 B0.5 &A`	Move to the absolute coordinate position (0, 0. Origin) from current position (any coordinate position). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, in absolute coordinate format)
	<Set parameters for interpolation.>
`SPD S0600.0 A5.0 B5.0`	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
`TYP &A`	Set motion format for the interpolating commands hereafter. (Absolute coordinates)
`LD D000 = #1`	Substitute numeric number 1 for a data register D000.
	<Interpolation (1) to (3)>
`REP #3`	Repeat three times between REP to NXT.
`MOV P@D000`	Movement follows the contents of point number referred by data register D000.
`MOV P0000`	Move to the point P0000.
`CAL D000 = D000 + #1`	Add one to the number of point register to be referred. (Add 1 to the data register D000.)
`NXT`	Return to the REP #3. (start of repetition)
	<End of motion.>
`END CSTP`	

15.3.3. ARC Command in Two Axes Motion

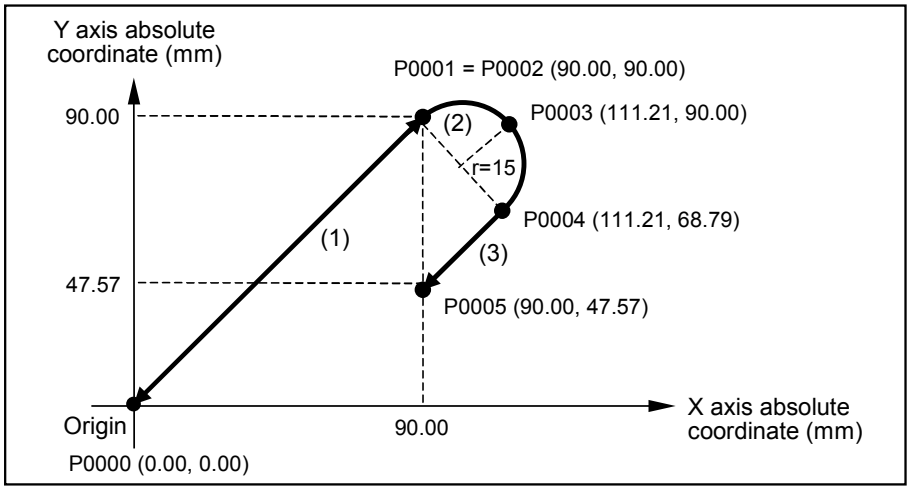
Motion outline: Move to the absolute coordinate position (0.00, 0.00) from current position (any point)

(1) Move to an absolute coordinate position (90.00, 90.00).

(2) Execute a circular arc interpolation. .
Starting point: (90.00, 90.00)
Passing point: (111.21, 90.00)
Ending point: (111.21, 68.79)

(3) Move to an absolute coordinate position (90.00, 47.57).

Figure 15-72: Outline of motion



Example 1: Circular arc interpolation on absolute coordinate (Setting of starting point P0002 may be omitted in the absolute coordinate format.)

Set point data:

```
P0000 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0001 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0003 X0111.21 Y0090.00 Zxxxx.xx Rxxxx.xx
P0004 X0111.21 Y0068.79 Zxxxx.xx Rxxxx.xx
P0005 X0090.00 Y0047.57 Zxxxx.xx Rxxxx.xx
```

Program	Description
	<Initialize: Move to P0000.>
`MOV P0000 S0100.0 A0.5 B0.5 &A`	Move to P0000 from current position (any coordinates position). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	<Set parameters for interpolation.>
`SPD S0600.0 A5.0 B5.0`	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
`TYP &A`	Set motion format for the interpolating commands hereafter. (Absolute coordinates)
	<Interpolation (1): Linear interpolation>
`MOV P0001`	Move to the point P0001.
	<Interpolation (2): Circular arc interpolation >
`ARC P0003 P0004`	Execute circular arc interpolation with passing the point P0003 and ending at the point P0004. (The start point is omitted.)
	<Interpolation (3): Linear interpolation>
`MOV P0005`	Move to the point P0005.
	<End of motion.>
`END CSTP`	

Example 2: Change coordinate format of linear interpolation to relative coordinate in Example 1. (Setting the start point of circular arc interpolation P0002 is required in relative coordinate format.)

- Refer to "Figure 15-72: Outline of motion."

Set point data:

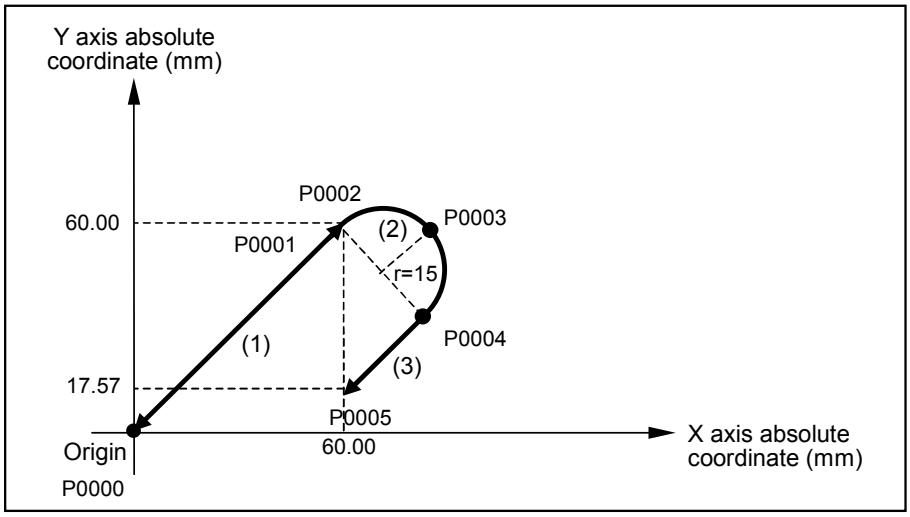
```
P0000 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0001 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0002 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0003 X0111.21 Y0090.00 Zxxxx.xx Rxxxx.xx
P0004 X0111.21 Y0068.79 Zxxxx.xx Rxxxx.xx
P0005 X-0021.21 Y-0021.22 Zxxxx.xx Rxxxx.xx
```

Program	Description
	<Initialize: Move to P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to P0000 from current position (any coordinate positions). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	<Set parameters for interpolation.>
SPD S0600.0 A5.0 B5.0	Set motion speed, acceleration and deceleration that will be used in interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
TYP &I	Set motion format for the interpolating commands hereafter. (Relative coordinates)
	<Interpolation (1): Linear interpolation>
MOV P0001	Move the distance in relative coordinate format obtained from the data of P0001.
	<Interpolation (2): Circular arc interpolation >
ARC P0002 P0003 P0004	Execute circular arc interpolation with P0002, as start point, P0003 as passing point and P0004 as ending point. (In relative coordinate format.)
	<Interpolation (3): Linear interpolation>
MOV P0005	Move the distance in relative coordinate format obtained from data of P000.
	<End of motion.>
END CSTP	

- Motion shown in Figure 15-73 is an example if P0001 is changed as described below.

```
P0001 X0060.00 Y0060.00 Zxxxx.xx Rxxxx.xx
```

Figure 15-73: Outline of motion



Note:

- The start point of circular arc interpolation may be omitted in absolute coordinate format. However, the current position (start position), passing point and ending point shall be on the same circle.

- A circular interpolation in relative coordinate format is an interpolation of which trajectory, that is defined by the starting position, passing point and ending point in absolute coordinate format, shifts to the current position. The absolute coordinate of the starting point shall be the current position.

- The start point cannot be omitted in the relative coordinate format.

15.3.4. CIR Command in Two Axes Motion

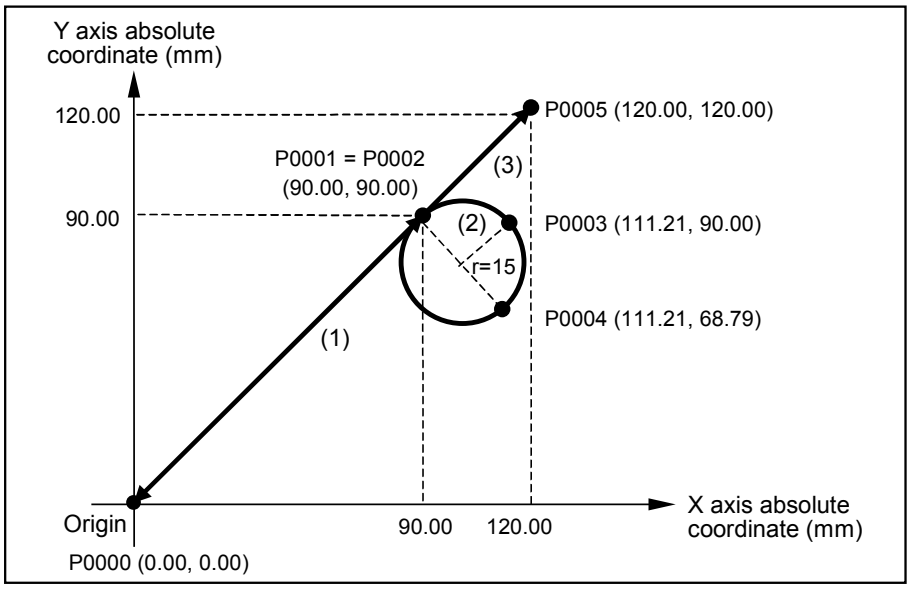
Outline of motion: Move to absolute coordinate position (0.00, 0.00) from current position (any coordinate position).

(1) Move to a point (90.00, 90.00, absolute coordinate position)

(2) Execute a circular interpolation in absolute coordinate format with:
Start point: (90.00, 90.00)
Passing points: (111.12, 90.00) and (111.12, 68.79)

(3) Move to the point (120.00, 120.00) in absolute coordinate format.

Figure 15-74: Outline of motion



Example 1: Circular interpolation in absolute coordinate format (Setting starting point of circular interpolation P0002 may be omitted in absolute coordinate format.)

Set point data:

```
P0000 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0001 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0003 X0111.21 Y0090.00 Zxxxx.xx Rxxxx.xx
P0004 X0111.21 Y0068.79 Zxxxx.xx Rxxxx.xx
P0005 X0120.00 Y0120.00 Zxxxx.xx Rxxxx.xx
```

Program	Description
	<Initialize: Move to P0000.>
`MOV P0000 S0100.0 A0.5 B0.5 &A`	Move to P0000 from current position (any coordinate position). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	<Set parameters for interpolation.>
`SPD S0600.0 A5.0 B5.0`	Set motion speed, acceleration and deceleration that will be used in a interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
`TYP &A`	Set motion format for the interpolating commands hereafter. (Absolute coordinate format)
	<Interpolation (1): Linear interpolation>
`MOV P0001`	Move to the point P0001.
	<Interpolation (2): Circular interpolation >
`CIR P0003 P0004`	Execute circular arc interpolation with passing the point P0003 and ending at the point P0004. (Specifying the start point is omitted.)
	<Interpolation (3): Linear interpolation>
`MOV P0005`	Move to the point P0005.
	<End of motion.>
`END CSTP`	

Example 2: Change coordinate format of linear and circular interpolation in Example 1 to relative coordinate. (Setting of start point P0002 is required in relative coordinate format.)

- Refer to "Figure 15-74: Outline of motion."

Set point data:

```
P0000 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0001 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0002 X0090.00 Y0090.00 Zxxxx.xx Rxxxx.xx
P0003 X0111.21 Y0090.00 Zxxxx.xx Rxxxx.xx
P0004 X0111.21 Y0068.79 Zxxxx.xx Rxxxx.xx
P0005 X0030.00 Y0030.00 Zxxxx.xx Rxxxx.xx
```

Program	Description
	<Initialize: Move to P0000.>
MOV P0000 S0100.0 A0.5 B0.5 &A	Move to P0000 from current position (any coordinate position). (Resultant speed: 100 mm/s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	<Set parameters for interpolation.>
SPD S0600.0 A5.0 B5.0	Set motion speed, acceleration and deceleration that will be used in a interpolation hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 5 m/s^2)
TYP &I	Set motion format for the interpolating commands hereafter. (Relative coordinate format)
	<Interpolation (1): Linear interpolation>
MOV P0001	Move the distance in relative coordinate format obtained from data of P0001.
	<Interpolation (2): Circular interpolation >
CIR P0002 P0003 P0004	Execute circular arc interpolation with start point P002 (current position), passing point P0003 and ending at the point P0004. (Relative coordinates format)
	<Interpolation (3): Linear interpolation>
MOV P0005	Move the distance in relative coordinate format obtained from data of point P0005.
	<End of motion.>
END CSTP	

- It is possible to perform the same interpolation when the above point registers are changed as shown below.

```
P0002 X0000.00 Y0000.00 Zxxxx.xx Rxxxx.xx
P0003 X0021.21 Y0000.00 Zxxxx.xx Rxxxx.xx
P0004 X0021.21 Y-0021.00 Zxxxx.xx Rxxxx.xx
```

Note:

- The start point of circular arc interpolation may be omitted in absolute coordinate format. However, the current position (start position), passing point and ending point shall be on the same circle.

- A circular interpolation in relative coordinate format is an interpolation of which trajectory, that is defined by the starting position, passing point and ending point in absolute coordinate format, shifts to the current position. The absolute coordinate of the starting point shall be the current position.

- The start point cannot be omitted in the relative coordinates format.

15.3.5. Continue Path in Two Axes motion

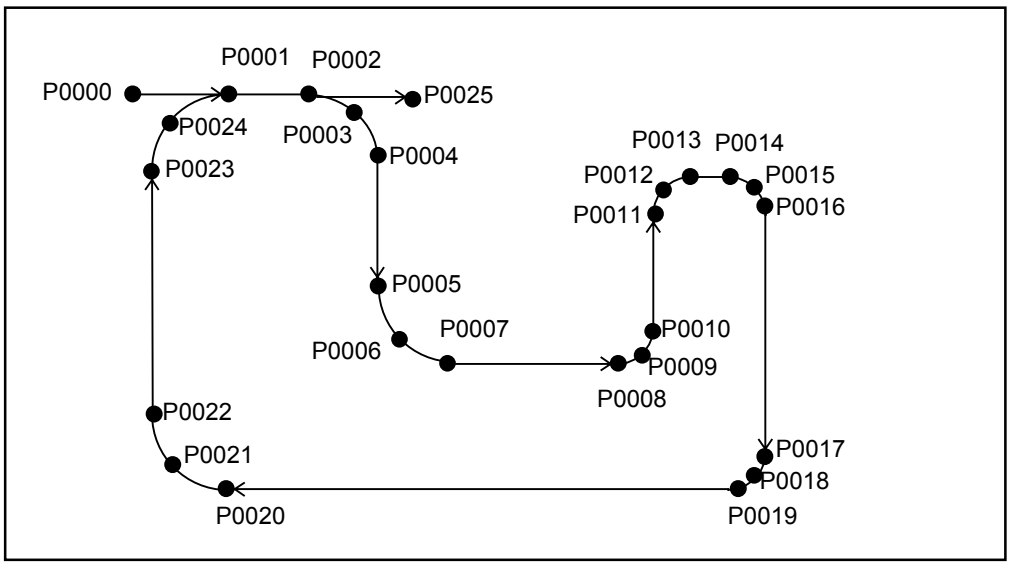
Outline of motion

This program example is for an application such as a sealing machine (apply adhesive) which requires constant speed.

(1) Move to the starting position P0000 from the current position.

(2) Move to the end point (P0000) from the start point (P0000) under constant speed.

Figure 15-75: Outline of motion



- Radius to connect two lines at the inflection point
  Find out the size of radius referring to "Figure 16-21: Size of radius" from allowable acceleration / deceleration. Use the size of radius for an arc to connect two lines smoothly.

Set point data:

```
P0000 X0020.00 Y0120.00 Z××××.×× R××××.××
P0001 X0060.00 Y0120.00 Z××××.×× R××××.××
P0002 X0080.00 Y0120.00 Z××××.×× R××××.××
P0003 X0094.14 Y0114.14 Z××××.×× R××××.××
P0004 X0100.00 Y0100.00 Z××××.×× R××××.××
P0005 X0100.00 Y0070.00 Z××××.×× R××××.××
P0006 X0105.85 Y0055.85 Z××××.×× R××××.××
P0007 X0120.00 Y0050.00 Z××××.×× R××××.××
P0008 X0160.00 Y0050.00 Z××××.×× R××××.××
P0009 X0167.07 Y0052.92 Z××××.×× R××××.××
P0010 X0170.00 Y0060.00 Z××××.×× R××××.××
P0011 X0170.00 Y0090.00 Z××××.×× R××××.××
P0012 X0172.92 Y0097.07 Z××××.×× R××××.××
P0013 X0180.00 Y0100.00 Z××××.×× R××××.××
P0014 X0190.00 Y0100.00 Z××××.×× R××××.××
P0015 X0197.07 Y0097.07 Z××××.×× R××××.××
P0016 X0200.00 Y0090.00 Z××××.×× R××××.××
P0017 X0200.00 Y0030.00 Z××××.×× R××××.××
P0018 X0197.07 Y0022.92 Z××××.×× R××××.××
P0019 X0190.00 Y0020.00 Z××××.×× R××××.××
P0020 X0060.00 Y0020.00 Z××××.×× R××××.××
P0021 X0045.86 Y0025.86 Z××××.×× R××××.××
P0022 X0040.00 Y0040.00 Z××××.×× R××××.××
P0023 X0040.00 Y0100.00 Z××××.×× R××××.××
P0024 X0045.86 Y0114.14 Z××××.×× R××××.××
P0025 X0110.00 Y0120.00 Z××××.×× R××××.××
```

Program

Program	Description
	<Initialize: Move to point P0000.>
MOV P0000 S0300.0 A10.0 B10.0	Move to point P0000 from current position (any coordinate position). (Resultant speed: 300 mm/s, acceleration / deceleration 10.0 m/s^2, absolute coordinate format)
	Start a continue path operation.
CPS	
	<Set the parameters for continue path operation.>
SPD S0100.0 A5.0 B5.0	Set motion speed and acceleration and deceleration that are used in interpolations hereafter. (Resultant speed: 100 mm/s, acceleration and deceleration 5.0 m/s^2)
	<Interpolation>
MOV P0001	Move to point P0001.
	<General output>
OUT OP10 = ; 00000001	Close bit 0 of the general output port 10. (Start an adhesive to flow.)
	<Interpolation>
MOV P0002	Move to point P0002.
ARC P0002 P0003 P0004	Execute a circular arc interpolation starting from P0002 (current position), passing a point P0003 and ending at point P0004.
MOV P0005	Move to point P0005
ARC P0005 P0006 P0007	Execute a circular arc interpolation starting from P0005 (current position), passing point P0006 and ending at point P0007.
MOV P0008	Move to point P0008.
ARC P0008 P0009 P0010	Execute a circular arc interpolation starting from P0008 (current position), passing point P0009 and ending at point P0010.
MOV P0011	Move to point P0011.
ARC P0011 P0012 P0013	Execute a circular arc interpolation starting from P0011 (current position), passing point P00012 and ending at point P0013.
MOV P0014	Move to point P0014.
ARC P0014 P0015 P0016	Execute a circular arc interpolation starting from P0014 (current position), passing point P0015 and ending at point P0016.
MOV P0017	Move to point P0017.
ARC P0017 P0018 P0019	Execute a circular arc interpolation starting from P0017 (current position), passing point P0018 and ending at point P019.
MOV P0020	Move to point P0020.
ARC P0020 P0021 P0022	Execute a circular arc interpolation starting from P0020 (current position), passing point P0021 and ending at point P0022.
MOV P0023	Move to point P0023.
ARC P0023 P0024 P0001	Execute a circular arc interpolation starting from P0023 (current position), passing point P0024 and ending at point P0001.
	<General output operation>
OUT OP10 = ; 00000000	Open bit 0 of general output port 10. (Stop an adhesive to flow.)
	<Interpolation>
MOV P0025	Move to point P0025.
	End of a continue path.
CPE	
	<End of operation.>
END CSTP	

Note :

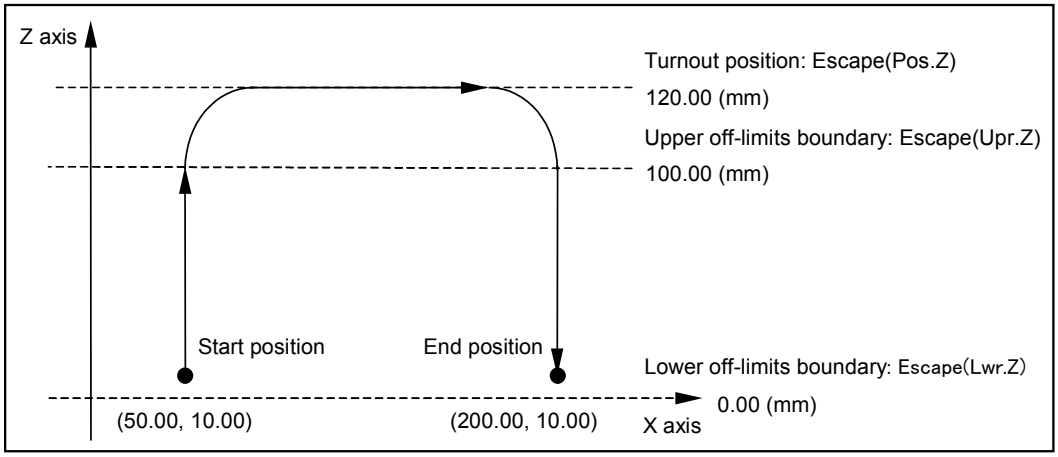
- Number of motion or motion steps between CPS and CPE commands are limited to 100. You cannot set steps 101 or over to a continue path.
- Number of steps increases two more if motion speed is set to MOV and MOVM commands.
- If 2 or more destinations are set to a MOMVM command, the number of steps increases by one for a destination.
- The motion speed limits the radius of connecting two lines. Refer to "Figure 16-21: Size of radius."

15.3.6. Arch Motion in Two Axes Motion

Outline of motion

- Set an upper and lower off-limits boundories and turnout position not to get in the off-limits area.
- An arch motion is executed only when starting or ending position of a motion is in the off-limits area.
- Effective axis unit for the arch motion commands is Z axis unit only in a multi-axis combination.

Figure 15-76: Outline of motion



Example 1: Set off-limits boundary and turnout position directly at the initial setting

- Specify the off-limits boundary and the turnout position of Z axis at the initial setting.
- Refer to "9.3.4. Parameters for Position and Coordinates" for the setting.

Table 15-3: Setting turnout position

Setting parameter	
Escape (upr. Z) Off-limits boundary (Z axis higher limit)	: 100.00
Escape (lwr. Z) Off-limits boundary (Z axis lower limit)	: 0.00
Escape (pos.Z) Turnout position (Z axis position)	: 120.00

Program	Description
	< Initialize: Move to starting point. >
MOV X0050.00 Z0010.00 S0100.0 A0.5 B0.5	Move to a point (50.0, 10.0, in absolute coordinate format) from current position (any position). (Resultant speed: 100 mm /s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	< Set the parameters for interpolation.>
SPD S0600.0 A3.0 B3.0	Set motion speed, acceleration and deceleration that are used to the interpolations hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 3.0 m/s^2)
	< Arch motion >
MOV X0200.00 &E	Move by arch motion to a point (200.0, 10.0 in absolute coordinate format).
	< End of motion >
END CSTP	

Example 2: Setting turnout position in the program

- Use the command (ESCZ) to set "off-limits boundary" of Z axis.
- Refet to "Figure 15-76: Outline of motion."

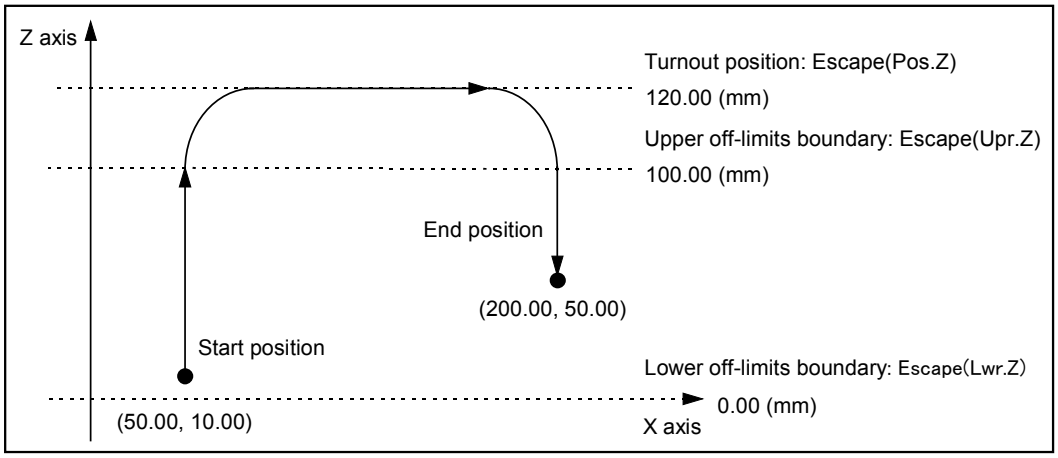
Program	Description
	< Initialize: Move to start point. >
MOV X0050.00 Z0010.00 S0100.0 A0.5 B0.5	Move to a point (50.0, 10.0, in absolute coordinate format) from current position (any position). (Resultant speed: 100 mm /s, acceleration and deceleration: 0.5 m/s^2, absolute coordinate format)
	< Set the parameters for interpolation.>
SPD S0600.0 A3.0 B3.0	Set motion speed, acceleration and deceleration that are used to the interpolations hereafter. (Resultant speed: 600 mm/s, acceleration and deceleration: 3.0 m/s^2)
	<Set parameter of the turnout position.>
ESCZ #0000.00 #0100.00 #0120.00	Set lower off-limits boundary to 0.0 mm, higher boundary to 100.00 mm and turnout position to 120.00 mm.
	<Arch motion >
MOV X0200.00 &E	Execute an arch motion to go to a point (200.0, 10.0, in absolute coordinate format).
	<End of program>
END CSTP	

Example 3: Changing the end position

- End position of arch motion can be changed as shown in Figure 15-77 by setting arch motion command to the program of the examples above.

```
MOV X0200.00 &E  →  MOV X0200.00 Z0050.00 &E
```

Figure 15-77: Outline of motion



Note:

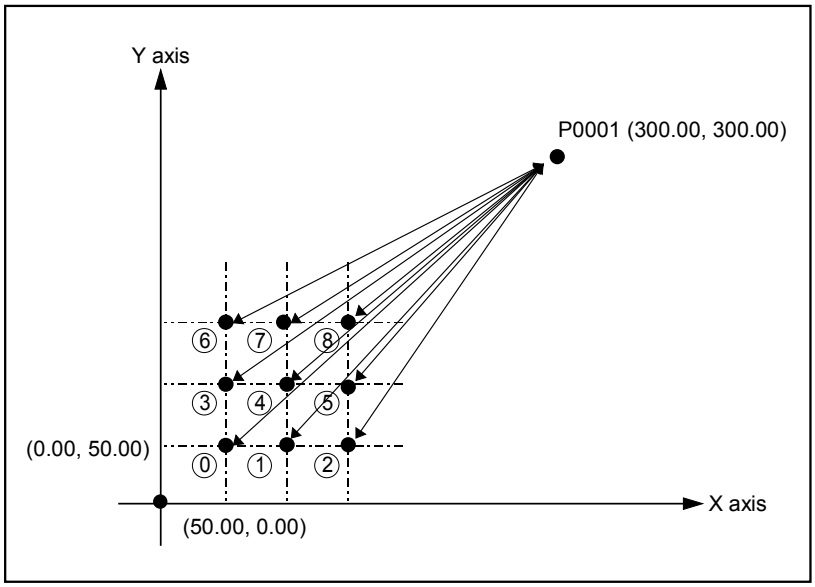
- Add an allowance of 10 mm approximately to the setting of off-limits boundary (higher / lower) and confirm that there is no interference with the objects in the off-limits area.
- When upper off-limits boundary and turnout position are too much apart, or acceleration is set too low, the trajectories of starting and ending of arch motion may not be the same.
- When both start and end points are not in the off-limits area, Z axis moves linearly to the end position. Be sure to make the start point or end point in the off-limits area.

15.3.7. Palletizing in Two Axes Motion

Outline of motion

- Execute a palletizing motion from P0001 to the positions of ⓪ to ⑧.

Figure 15-78: Outline of motion



Setting palletizing operation

- Use a number 00 as a pallet number in this programming example. Set palletizing operation to the palletizing operation number 00.
- Refer to "16.1. Palletizing" for details of setting.

[Outline of setting]

- Select the palletizing mode (PAL mode) then select Palletizing operation number 00.
- Enter to EDT mode next, then set the items listed in Table 15-4.

Table 15-4: Setting palletizing motion

Setting item		Description
Memo (Name of palletizing operation)	: DEMO	Input "DEMO."
Type (Select positioning pattern of palletizing operation.)	: MLT	Multiple step positioning pattern
Axis (Set operating axis)	: XX YY	Operate X and Y axis.
Start Position (Set start position.)	: X0050.00 Y0050.00	Coordinates (50, 50) : Position (0) in Figure 15-78.
Width (Set step distance)	: X0050.00 Y0050.00	50 mm each, X and Y axis
Size (Number of positioning intervals)	: X0002 Y0002	Two times, X and Y axis
Move Pattern (Set operation format.)	: XNRM	X axis takes precedence. Moving to one direction.

Set point data:

```
P0001 X0300.00 Y0300.00 Z××××.×× R××××.××
```

Program	Description
	<Initialize: Palletizing operation>
PALI QN00 $DEMO	Initialize the palletizing operation number 00.
	<Set repeating motion tag.>
TAG _00	
	<Interpolation>
MOV P0001 S0300.0	Move to a point P0001. (resultant speed 300 mm/s)
	<Palletizing operation>
PALM QN00 S0300.0	Palletizing operation (resultant speed: 300 mm/s)
TIM #0001.00	Timer: 1 second
	<Get palletizing status.>
QSTS D000 = QN00 QPC	Substitute next point register number of palletizing operation for data register D000. (* Note: Point register number starts from 0 (zero).
	<Confirm completion of palletizing operation.>
CMP D000 #0000 JNE _00	Jump to TAG_00 if the numeric number in data register D000 (number of next point register) is not 0 (zero).
	<Interpolation>
MOV P0001 S0300.0	Move to a point P0001. (Resultant speed: 300 mm/s)
	<End of program>
END CSTP	

Note:

- Palletizing position number starts from 0 (zero) and increases by one for every execution of PALM command.
- When the last palletizing operation is completed, the palletizing position number returns to 0 (zero).

15.3.8. Multitask

15.3.8.1. Example of Program 1

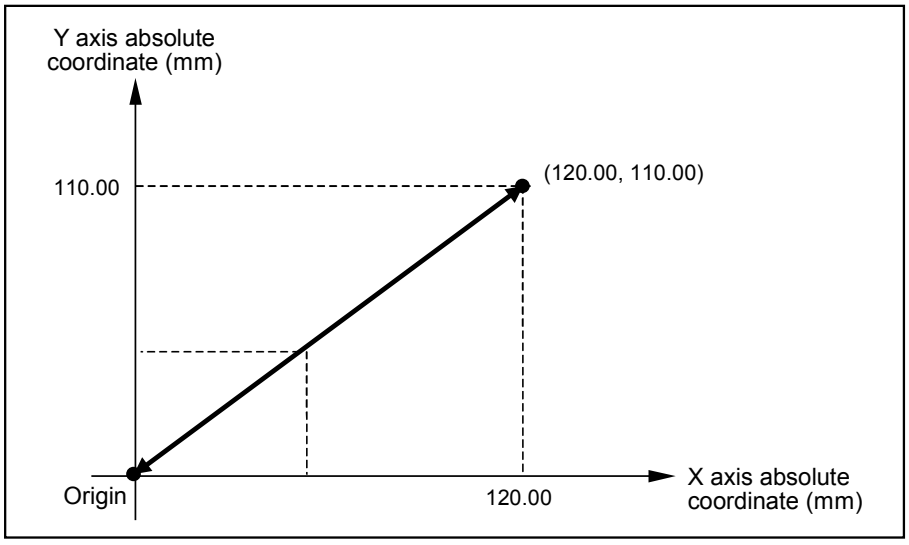
Outline of motion

- Executes repetitive interpolation starting from current position.
  Repeat linear interpolation to the destination point (120.00, 110.00)
  Repeat linear interpolation to the destination (000.00, 000,00)

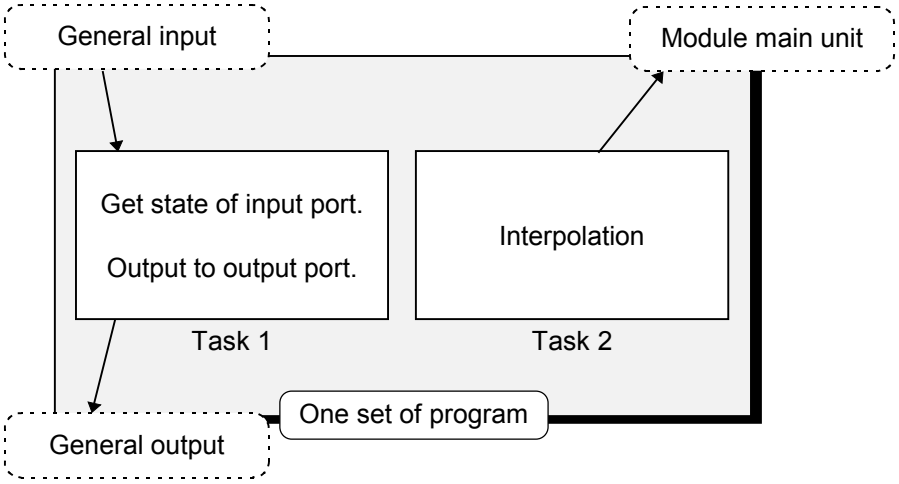
- Closes Bit 0 of general output port OP 10 when bit 0 of general input port IP10 is ON in the middle of the interpolation.

- Opens Bit 0 of general output port OP10 when bit 0 of general input port IO10 is off in the middle of the interpolation.

Figure 15-79: Outline of motion

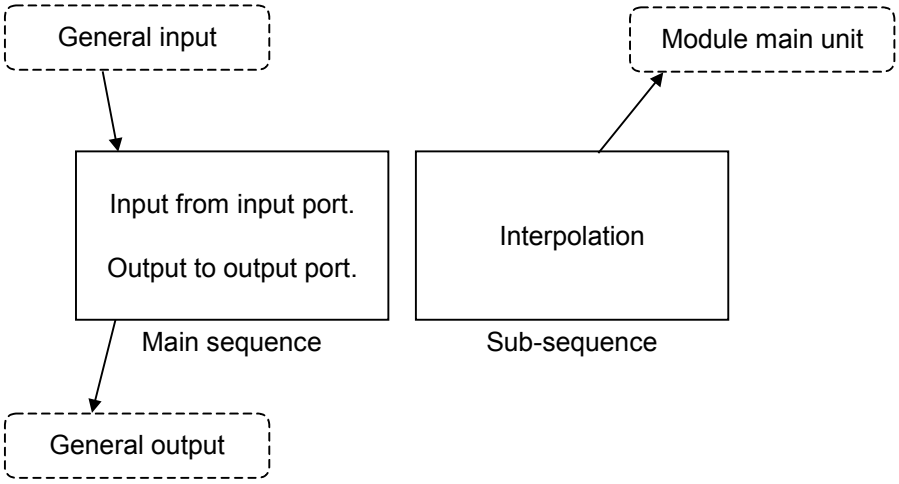


Example 1: Set parallel processing as a motion format to MOV command



Program	Description
	<Initialize>
LD D001 = #1	Set the flag to switch "go" and "return" of interpolation. (#-1: go, #1 return)
	Interpolation
TAG _LOOP	
MSTS D000 = X Y	Read out state of interpolation: (#0: stopping #1: moving)
CMP D000 #0 JNE _OUTPUT	Output when started interpolation is not completed.
CAL D001 = D001 * #-1	Switch the flag for "go and return" motion.
CMP D001 #1 JEQ LOCUS2	Switch to "return" the interpolation.
	<Interpolation "go">
TAG _LOCUS1	
MOV X0120.00 Y0110.00 &P	Outputs to start the interpolation as the parallel processing motion.
JMP OUTPUT	
	<Interpolation "return">
TAG _LOCUS2	
MOV X0000.00 Y0000.00 &P	Outputs to start the interpolation as the parallel processing motion.
	<Output>
TAG _OUTPUT	
OUT OB100 = IB100	Output the state of bit 0 of general input port IP10 to bit0 of general output OP10.
JMP _LOOP	
END CSTP	

Example 2: Divide into main sequence and sub-sequence



Main sequence: Output to I/O port.

Program	Description
	<Sub-sequence>
CHLD $LOCUS	Start sub-sequence $LOCUS.
	<Output>
TAG _LOOP	
OUT OB100 = IB100	Output the state of bit0 of general input port IP10 to bit 0 of general output port OP10.
JMP _LOOP	Jump to TAG_LOOP. (Repeat)
END CSTP	

Sub-sequence (Name of program: $LOCUS): Only operates interpolation.

Program	Description
	<Interpolation>
TAG _LOOP	
MOV X0120.00 Y0110.00	Linear interpolation
MOV X0000.00 Y0000.00	Linear interpolation
JMP _LOOP	Jump to TAG_LOOP. (Repeat)
ENDC	

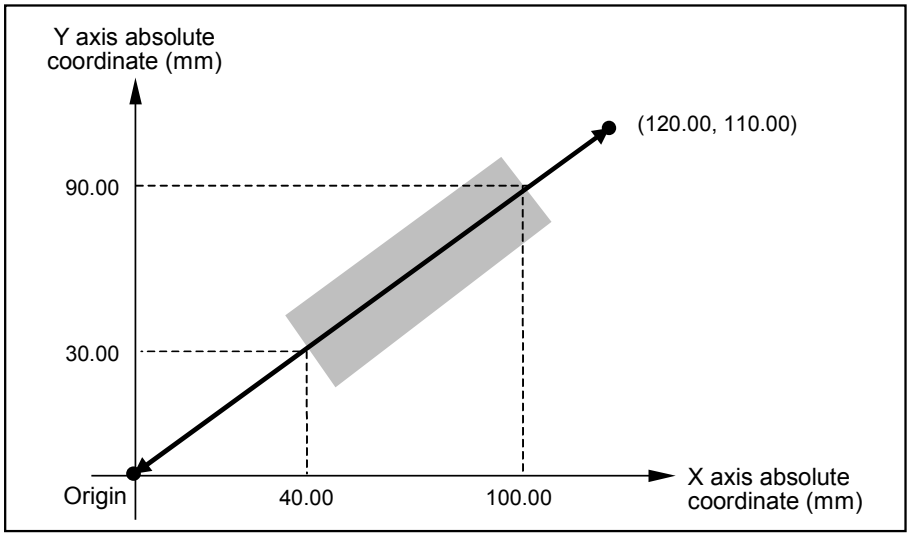
15.3.8.2. Example of Program 2 (2 axes motion)

Outline of motion

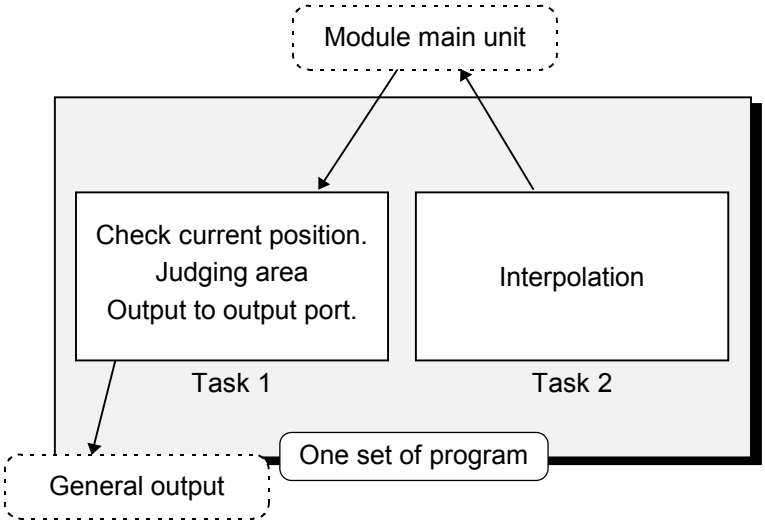
- Execute a repetitive interpolation starting from current position.
  Repeat linear interpolation to the destination (120.00, 110.00).
  Repeat linear interpolation to the destination (000.00, 000.00).

- Closes bit 0 of general output port OP00 while moving in the hatched area shown in Figure 15-80. It opens in the other area.

Figure 15-80: Outline of motion

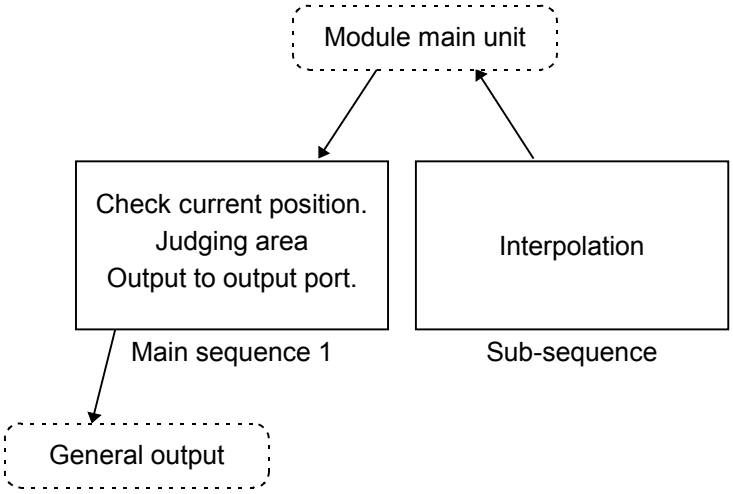


Example 1: Specify a parallel processing as a motion format to MOV command



Program	Description
	<Initialize>
LD D001 = #1	Set the flag to switch "go" and "return" of interpolation. (#-1: go, #1 return)
LD P0100 = X0040.00 Y0030.00 LD P0101 = X0100.00 Y0090.00	Store the coordinate data of area judging to P0100 and P0101.
	<Interpolation and judging area>
TAG _LOOP MSTS D000 = X Y	Get information on the state of interpolation: (#0: stopping #1: moving)
CMP D000 #0 JNE _JUDGE	Execute judging area when motion of interpolation is not completed.
CAL D001 = D001 * #-1	Switch the flag for "go and return" motion.
CMP D001 #1 JEQ LOCUS2	Switch the interpolation to "return."
	<Interpolation "go">
TAG _LOCUS1 MOV X0120.00 Y0110.00 &P JMP JUDGE	Outputs to start the interpolation as the parallel processing motion. Judging area
	<Interpolation "return">
TAG _LOCUS2 MOV X0000.00 Y0000.00 &P	Start interpolation as a parallel processing.
	<Area judging>
TAG _JUDGE TCH P0000 = X1 Y1	Store current position to P0000.
CMP PX0000 PX0100 JLT _OUTRNG CMP PY0000 PY0100 JLT _OUTRNG CMP PX0000 PX0101 JGT _OUTRNG CMP PY0000 PY0101 JGT OUTRNG	Is current position in the area ? (If not, jump to TAG_OUTRNG.)
	<Output result>
TAG _INRNG OUT OB100 = ;1 JMP _LOOP	In case of "in the area": Close bit 0 of general output port OP10. Jump to TAG_LOOP. (Repeat)
TAG _OUTRNG OUT OB100 = ;0 JMP _LOOP END CSTP	In case of "out of area": Open bit 0 of general output port OP10. Jump to TAG_LOOP. (Repeat)

Example 2: Divide into main sequence and sub-sequence



Main sequence: Judge the area and output to I/O

Program	Description
	<Initialize>
LD P0100 = X0040.00 Y0030.00	Store coordinate data of area judging to P0100 and P00101.
LD P0101 = X0100.00 Y0090.00	
	<Start sub-sequence.>
CHLD $LOCUS	Start sub-sequence $LOCUS.
	<Judging area>
TAG _LOOP	
	<Check current position>
TCH P0000 = X1 Y1	Store current position to P0000.
CMP PX0000 PX0100 JLT _OUTRNG	Is current position in the area ? (If not, jump to TAG_OUTRNG.)
CMP PY0000 PY0100 JLT _OUTRNG	
CMP PX0000 PX0101 JGT _OUTRNG	
CMP PY0000 PY0101 JGT _OUTRNG	
	<Output result>
TAG _INRNG	In case of "in the area":
OUT OB100 = ;1	Close bit 0 of general output OP10.
JMP _LOOP	Jump to TAG_LOOP. (Repeat)
TAG _OUTRNG	In case of "out of area":
OUT OB100 = ;0	Open bit 0 of output port OP10.
JMP _LOOP	Jump to TAG_LOOP. (Repeat)
END CSTP	

Sub-sequence (Name of sub-sequence: $LOCUS): Only operates interpolation.

Program	Description
	<Interpolation>
TAG _LOOP	
MOV X0120.00 Y0110.00	Linear interpolation
MOV X0000.00 Y0000.00	Linear interpolation
JMP _LOOP	Jump to TAG_LOOP. (Repeat)
ENDC	

Note:

- Outputted position signal includes some errors against actual position as the judging area in this program is done by polling.

(Blank Page)

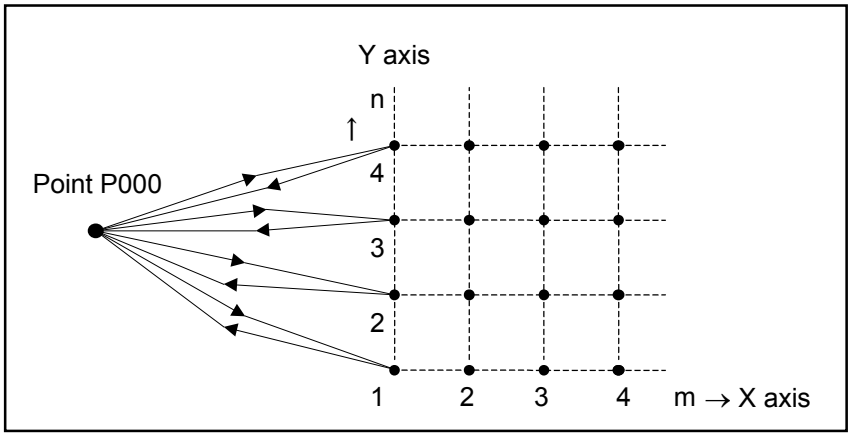
16. Description of Function

16.1. Palletizing (Multi-axis combination only)

16.1.1. Outline of Palletizing Operation

- It is possible to select a setting format of palletizing positions from those described below.
  - ◊ Multiple regular interval format : Palletizing positions are set by the points at equal intervals on X and Y coordinate axes of a pallet.
  - ◊ Divided sides format : Palletizing positions are set by the points on X and Y coordinate axes at divided positions by specified numbers on the pallet sides .
  - ◊ Three corners format : Palletizing positions are determined by the positions of three corners of the pallet.
- Palletizing positions are on a two dimensional flat surface defined by X and Y coordinate axes.

Figure 16-1: Palletizing



Example of motion:

1) Loading pallet

◊ Move an empty pallet A (completed palletizing) to the storage C.
(Palletizing motion shown in Figure 16-2 is an example to put out a work from a pallet and place it on a conveyor.)

◊ Take out pallet B from the storage of loaded pallets.

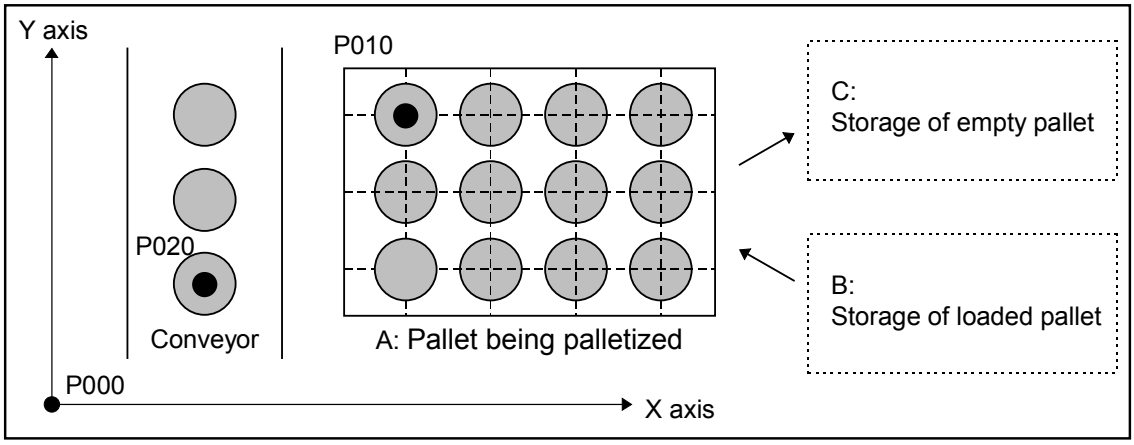
2) Palletizing motion

◊ Take out a work from pallet A. (point P010)

◊ Transfer a work to a conveyor (point P020).

* Repeat these motions to complete the palletizing.

Figure 16-2: Example of palletizing



16.1.1.1. Description of Terms

● Palletizing pattern number:

A palletizing pattern represents a combination of palletizing data that specify the contents of palletizing operation, such as start position of a palletizing, positioning step intervals and number of positions.

It will be numbered in 0 to 15. The palletizing data are edited in the palletizing mode and assigned to respective palletizing pattern numbers. The data is called palletizing pattern data.

● Palletizing operation number

This is a number used in a program to identify a palletizing operation. This number will be correlated by "PALI" command with a palletizing pattern number. Actual palletizing will be operated based on these operation numbers.

[Example]

PALI QN00 #0003 : Initialize palletizing operation number 00 by palletizing data of palletizing pattern 0003. The contents of palletizing operation number 00 are the same contents of palletizing pattern 03 thereafter.

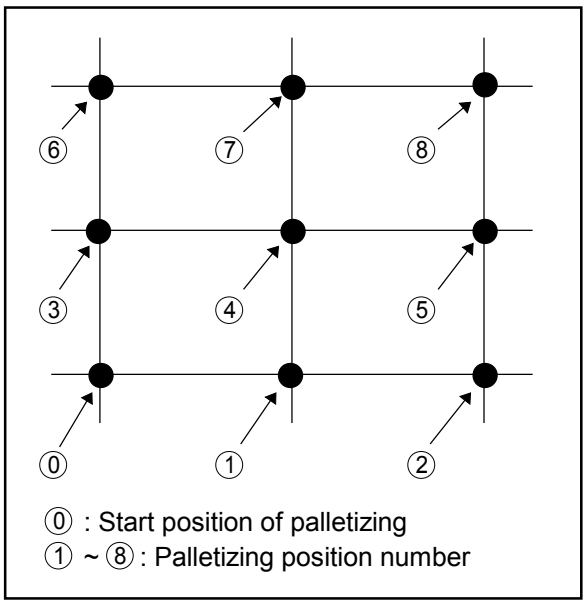
Palletizing operation number

Palletizing pattern number

● Palletizing position number

This is one of a data consists of a palletizing pattern. This is to indicate the order of palletizing position on a pallet. Approximately 10 000 points (100 × 100) are possible to set as the position number. However, coordinate data development over 4 000 is not applicable.

Figure 16-3: Palletizing position number



- Data development

  This is to make a positioning data (the coordinates of palletizing position) to be used for a program, referring to a data specified by a palletizing pattern number.

  Formats of data development, their codes and applications to palletizing are described hereunder.

  NOP No data development
  : Calculate the data based on the palletizing pattern number in the middle of palletizing operation.
  * Select this format as the basic handling way of palletizing operation.

  INI Data development (with initialization)
  : Develops palletizing point data to plural point registers that starts from specified register number. In INI format, the data is developed during execution of the initialize command (PALI).

  TCH Data development (without initialization)
  : Develops palletizing point data to plural point register that starts from specified register number. In TCF format, the development starts when F2 key is pressed in the palletizing pattern setting screen 1. (Refer to Figure 16-6: Setting palletizing pattern.)

16.1.1.2. Programming Procedure for Palletizing

16.1.1.2. Programming Procedure for Palletizing

◆ Palletizing program mode:

- In this mode you can set the data for palletizing motion.
- There are several functions for editing the data.
  - ◊ Naming and adding memo
  - ◊ Selection of palletizing pattern and changing data
  - ◊ Copy or delete palletizing number.
  - ◊ Store the data to the flash memory and read out the data

◆ Programming of palletizing

- Use the commands related to palletizing and program a palletizing operation.
- Refer to "15.2. Programming" for more detail.

Table 16-1: Program command of palletizing

Command	Outline of function
PALI	Command to initialize a palletizing operation. The command correlates the data of specified palletizing pattern number to a palletizing operation number to make the data usable for a specified unit. Be sure to set the command on the top of a program.
PALL	Command to call subroutine. This command calls a subroutine at the end of a palletizing motion. This command may be used to call a subroutine for loading pallet.
PALE	Command to call a subroutine. This command calls a subroutine in the middle of palletizing motion. This command is used when loading a pallet is necessary in the middle of a palletizing.
PALM	Command to move to a palletizing position of specified palletizing pattern number. After the motion is completed, the palletizing position number increases for one.
PALN	Command to change a position number of specified palletizing operation. This command is used when you want to skip a pallet position.
QSTS	This command is to read out palletizing status of a specified pallet number. (When you need to check the next palletizing position number, etc.)

- Refer to "15.3. Example of Program" as well.

16.1.2. Outline of Palletizing Program Mode

- There are 7 modes in the palletizing program mode and it is composed in the tree structure as shown in Figure 16-5.

Figure 16-4: Palletizing program mode

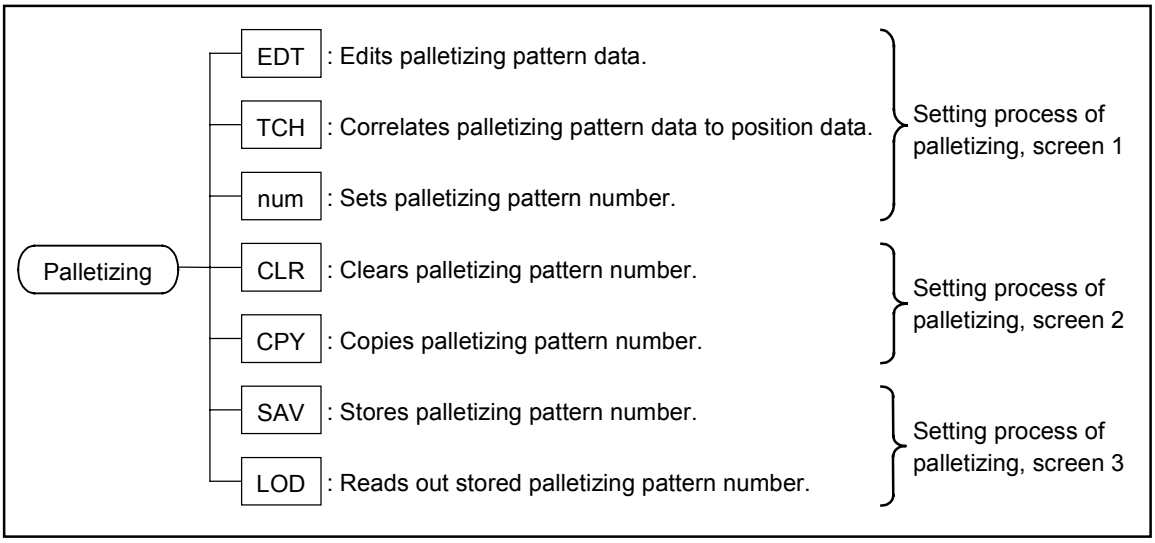
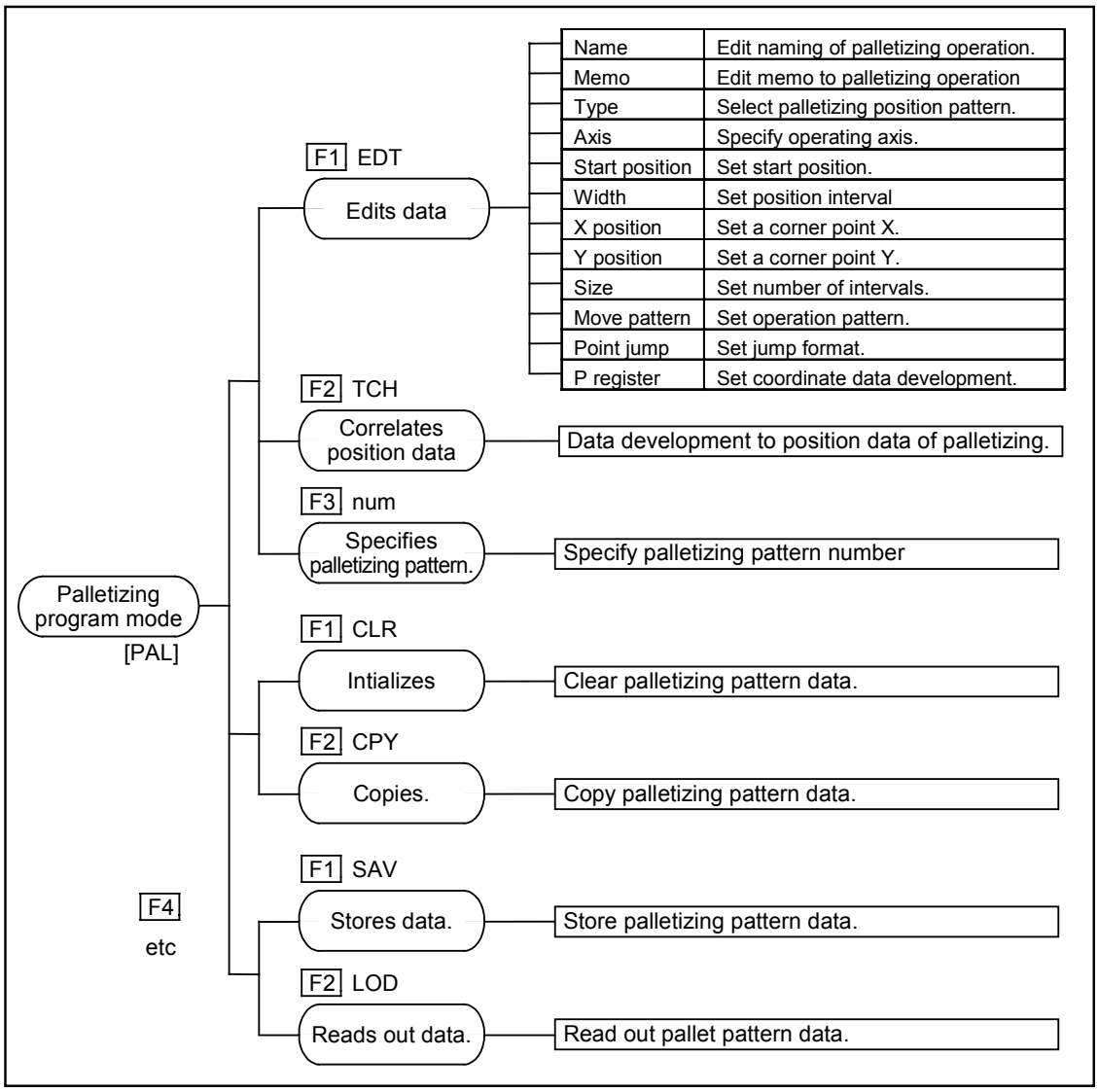
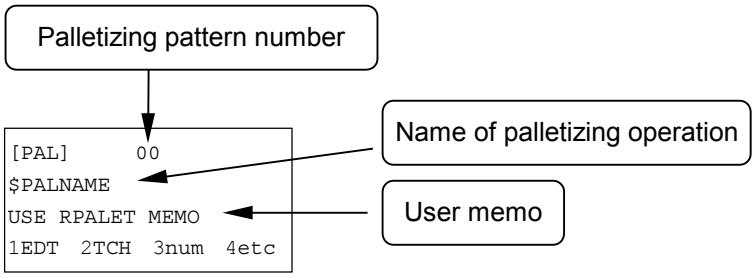


Figure 16-5: Palletizing program mode list



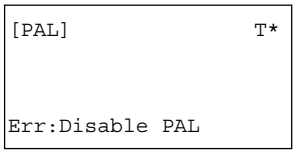
16.1.2.1. Screen of Palletizing Program Mode

- The screen as shown below appears when palletizing program mode is selected. It shows setting state of palletizing data.
- Pallet pattern number is on the first line, name of palletizing operation is on the 2nd line and user memo is on the 3rd line. Maximum number of characters is 8 for naming and 16 for user memo.



* When changing palletizing data is attempted during programmed operation (or operation in any manner), the screen gives message as shown below and the system interrupts the setting palletizing process.

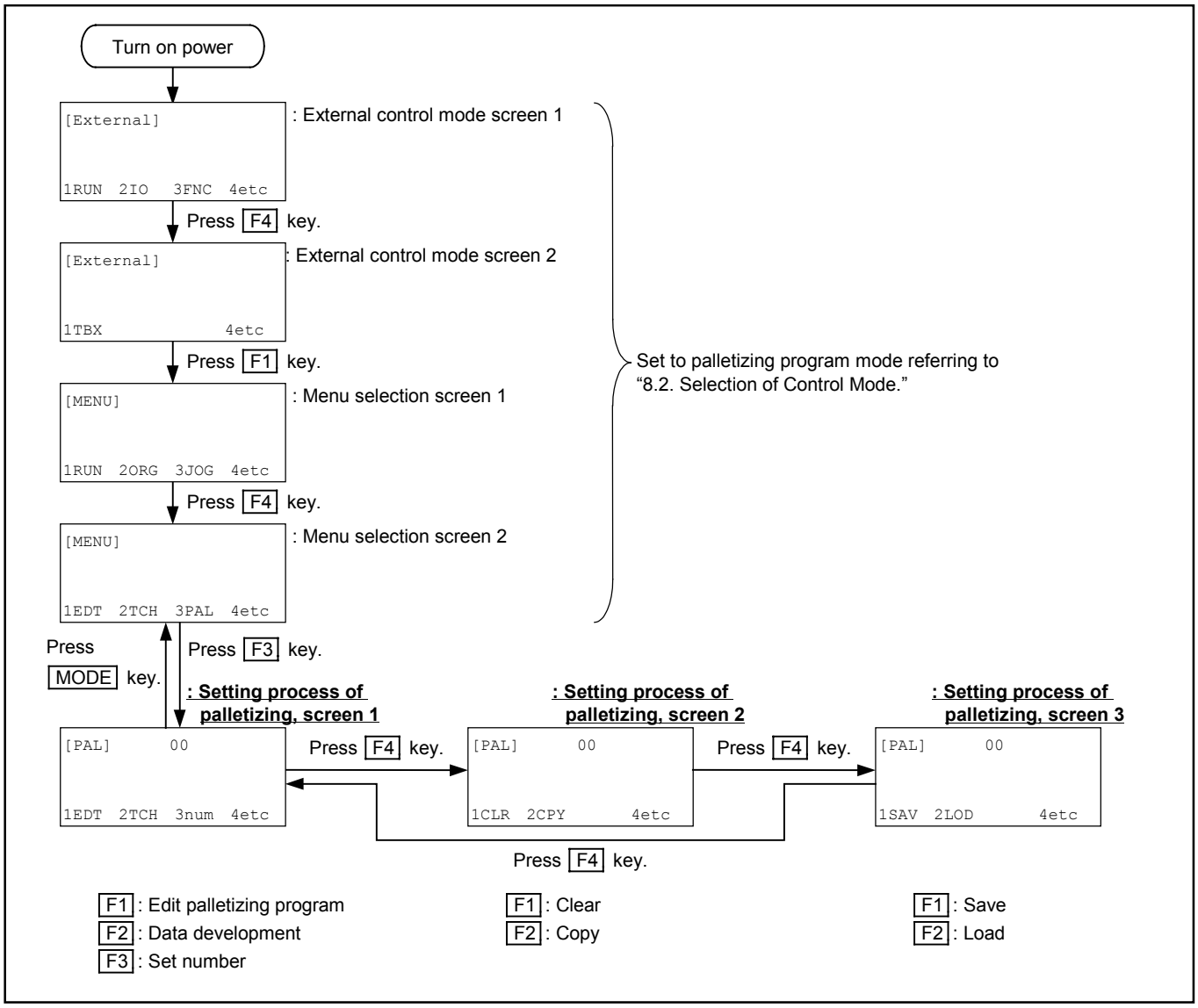
* This state remains until MODE or SET key is pressed.



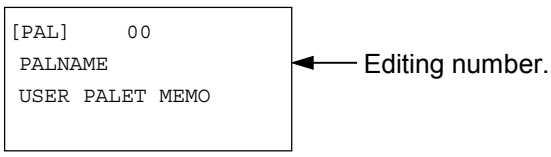
16.1.3. Setting Palletizing Pattern Data

- The four function keys F1 to F4 select the contents of setting process of palletizing in the selecting screen. F4 key is to move to the next setting screen.

Figure 16-6: Setting palletizing pattern



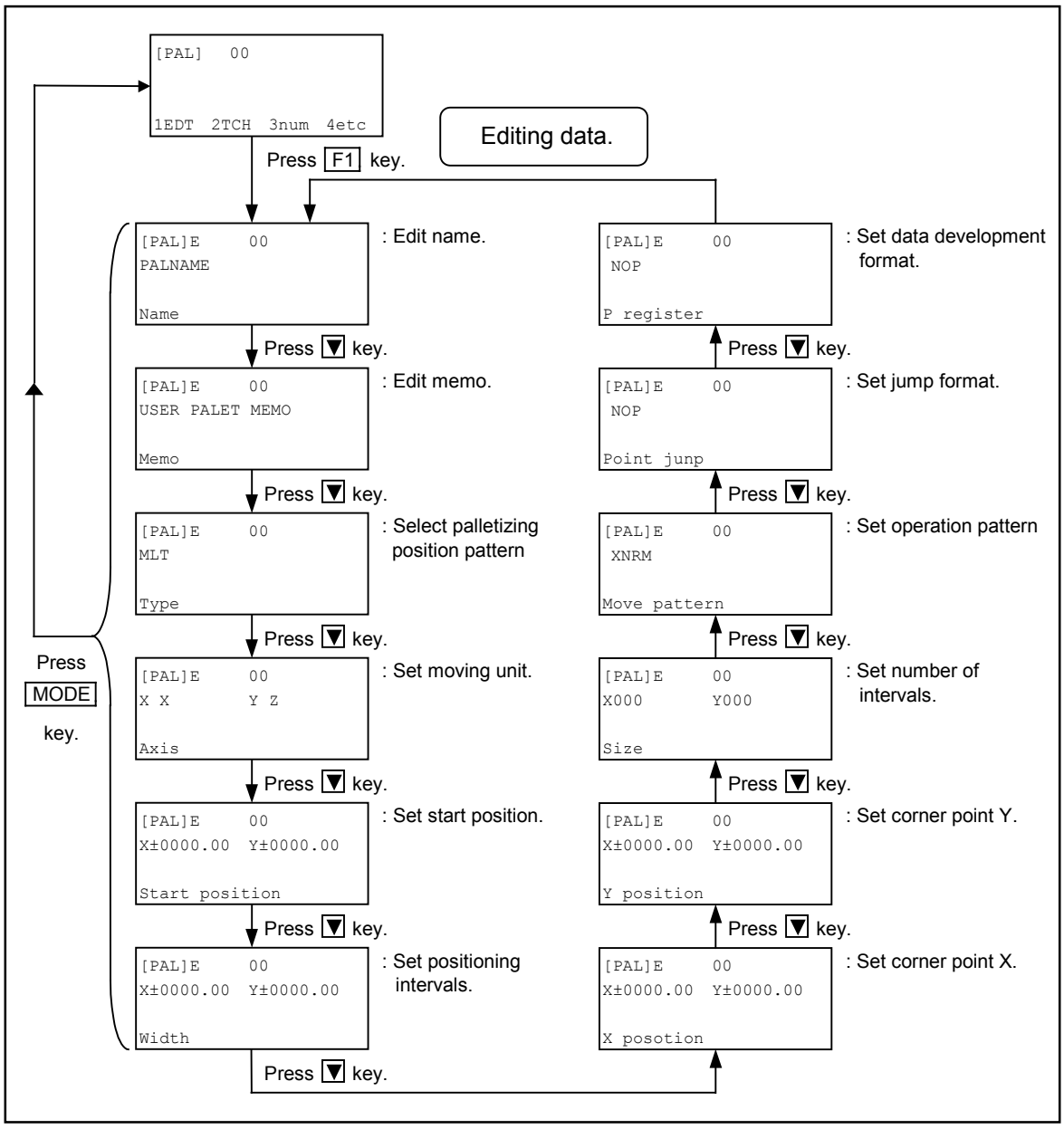
- ▲ and ▼ keys can change palletizing pattern number in the selecting screen of setting process of palletizing.
  The number decreases by one as ▲ key is pressed while it increases by one as ▼ key is pressed.
  The cursor appears on the line of palletizing pattern number when F3 key is pressed in a display showing 'num'. Set number using Numeric key and CLR key.



16.1.4. Editing Palletizing Pattern Data

- Pallet pattern data can be edited in the screens shown in Figure 16-7 that appears when F1 keys is pressed in the selecting screen of setting procedure of palletizing. Use ▲ and ▼ keys to select the editing contents. (Screen scrolls in the order as shown in the figure.) The screen displays a letter E on the right side of [PAL] while editing.

Figure 16-7: Editing data of palletizing pattern



- The cursor, which appears in the screen when ▶ is pressed after the editing objective is displayed, enables to change editing contents. Pressing SET key makes the set contents effective while pressing MODE key interrupts setting.

16.1.4.1. Naming Palletizing Pattern

- The screen shown below appears for naming of palletizing pattern. SET key is to enter character strings and MODE key is to interrupts entering.

```
[PAL]E   00
 PALNAME

 Name
```

- Move the cursor by ◀ and ▶ keys and select characters by ▲ and ▼ key. ▲ and ▼ keys change the characters indicated by the cursor in the order as shown below.

  ▲ key -------A → 9 → 8 ~ 0 → 1 → Z → Y → X---------- B → A → 9 → 8 ~

  ▼ key -------A → B → C ~ X → Y → Z → 0 → 1--------- 8 → 9 → A → B ~

  * Letter A is set when ▲ and ▼ is pressed in a blank.

- CLR key is to clear a character indicated by the cursor. +/- key is to insert a selected letter on the cursor position. The number of letters for naming is limited to 8.

  * Name of palletizing pattern is not set in the initializing state. You may use the name instead of palletizing pattern number in PALI command. Naming does not do anything to execute a programmed palletizing operation.

16.1.4.2. Writing Memo

- The screen shown below appears for putting memo on a program. SET key is to enter a character string and MODE key is to interrupts entering.

```
[PAL]E  00
 PALMEMO

 Memo
```

- Move the cursor by ◀ and ▶ keys and select characters by ▲ and ▼ keys. ▲ and ▼ keys change the characters indicated by the cursor in the order as shown below.

  ▲ key -------A → 9 → 8 ~ 0 → 1 → Z → Y → X---------- B → A → 9 → 8 ~

  ▼ key -------A → B → C ~ X → Y → Z → 0 → 1--------- 8 → 9 → A → B ~

  * Letter A is set when ▲ and ▼ is pressed in a blank.

- CLR key is to clear a character indicated by the cursor. +/- key is to inset a selected letter on the cursor position. The number of letters for memorandum is limited to 16.

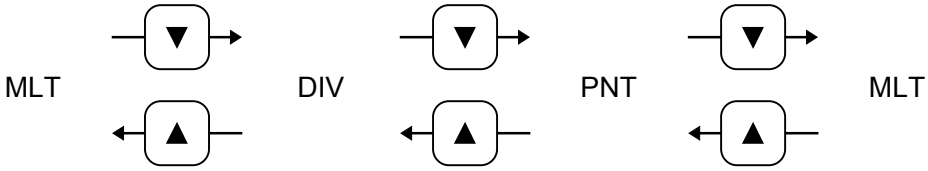
  * Memo to a program is not set in the initializing state. Memo does not do anything to a palletizing operation. Use it as a your note.

16.1.4.3. Format of Palletizing Position Setting

- The screen shown below appears for selecting a palletizing pattern. Press ▶ key to indicate the cursor, then select a pattern by ▲ and ▼ keys. SET key is to enter the selection and MODE key to interrupt entering.

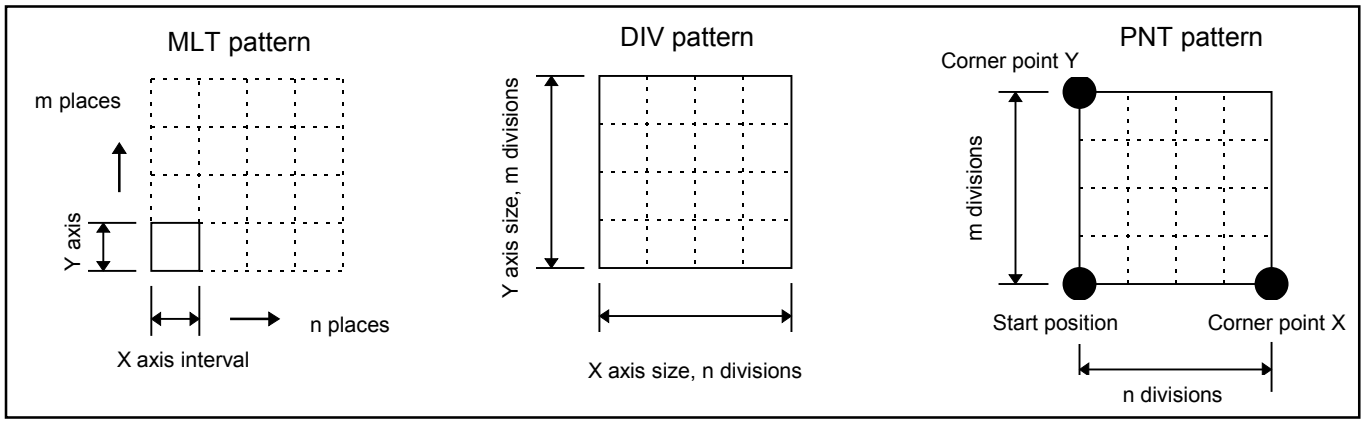
```
[PAL]E   00
MLT

Type
```

- Scroll and set the screen using ▲ and ▼ keys for palletizing position.



- Palletizing position setting formats and their codes are shown below.
  MLT ----- Multiple regular interval
  DIV ------ Divided sides
  PNT------ Three corners

Figure 16-8: Position setting format



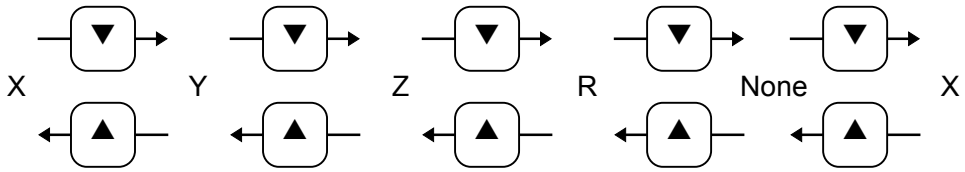
16.1.4.4. Setting Operating Unit

- The screen shown below appears for selecting an operating unit. Select an axis by ◀ and ▶ keys and use ▲ and ▼ keys to select operating unit. SET key is to enter the selection and MODE key is to interrupt entering.

```
[PAL]S   00
 XX          YZ

 Axis
```

This is an example to assign the actual operating axis units X and Y to the coordinate axis X and Z of palletizing pattern data respectively. The screen shows that X axis unit is assigned to X coordinate axis while Y axis unit to Z coordinate axis.

- Scroll and set the screen using ▲ and ▼ keys for name of motion axis.



* Selection of "None" makes single axis palletizing possible.

- This setting specifies how to assign the X and Y coordinate axes of palletizing pattern data to an actual operating axis units.

16.1.4.5. Setting Start Position

- For selecting the start position of palletizing, use a screen to set a point number or direct position data as shown below. Press ▶ key to indicate the cursor, then select ▲ and ▼ keys to switch the screen of register number or direct data.

- The screen shown below appears for setting a point number. Set a number of the start point of palletizing operation by Numeric keys. SET key is to enter the setting and MODE key is to cancel it.

```
[PAL]E   00
 P0000

 Start Position
```
← Point number

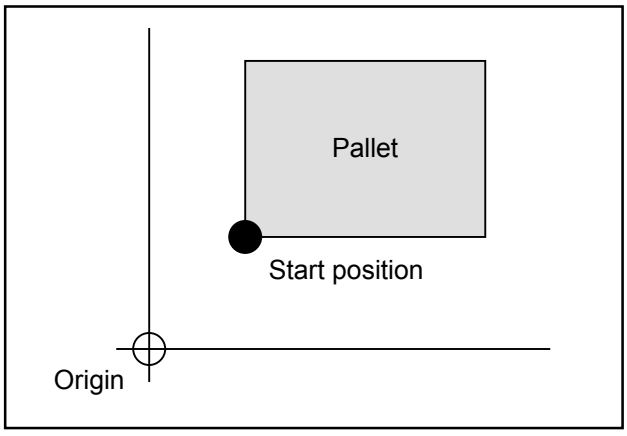
- The screen shown below appears to set a position data directly. Select an axis by ◀ and ▶ keys and set coordinate data of the start position of palletizing using numeric keys. SET key is to enter the setting and MODE key is to cancel it.

```
[PAL]E   00
 X±0000.00  Y±0000.00

 Start Position
```

- In the absolute coordinate format, the coordinate data set in this process is specified as the start position of palletizing operation. In case of the relative coordinate format, the current position is specified as the start point, and from this point a palletizing operation starts following the pallet position number.
  (In case of the relative coordinate format, the start position set in this process is used only for making the palletizing pattern data.)

- When start position is specified by a point number, the data for a unit other than specified operation axis unit does not have the meaning. If the position data of operating axis unit is not set (when data is ××××.××), the palletizing operation is impossible to perform.

Figure 16-9: Start position of palletizing



16.1.4.6. Setting Position Interval

- Use a screen to set point number, or a screen to set position interval directly as shown below for setting position interval. Press ▶ key to indicate the cursor, then use ▲ and ▼ keys to switch the screen of number or interval data.

- In case of setting point number, the screen displays as shown below. Use Numeric keys to select a point number that represents interval data. SET key is to enter the data and MODE key is to cancel it.

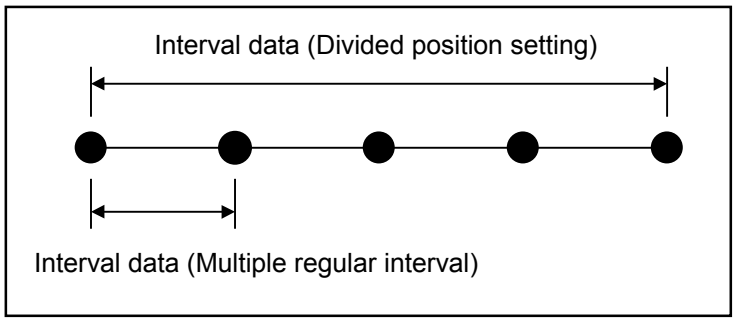
```
[PAL]E   00
 P0000

 Width
```
← Point number

- In case of direct setting of position interval data, the screen displays as shown below. Press ◀ and ▶ keys to set an axis unit and Numeric keys to set interval data. SET key is to enter the data and MODE key is to cancel it.

```
[PAL]E   00
 X±0000.00  Y±0000.00

 Width
```

- When specify the position interval by a point number, the data for a unit not specified as an axis unit does not have any meaning. If the position data of an axis unit is not set (when data is ××××.××), the palletizing operation is impossible to perform. The interval data is valid for DIV (Divided position setting) and MLT (Multiple position interval) format only.

Figure 16-10: Position interval data



16.1.4.7. Setting Position of Corner Point X

- For setting position of corner point X, use a screen to set point number or a screen for direct position data setting as shown hereunder. Press ▶ key to indicate the cursor, then use ▲ and ▼ keys to switch the screen of point number or position data.

- In case of using point number, the screen displays as shown below. Use Numeric keys to select a point number that indicates the position of corner point X. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]E   00
 P0000

X Position
```

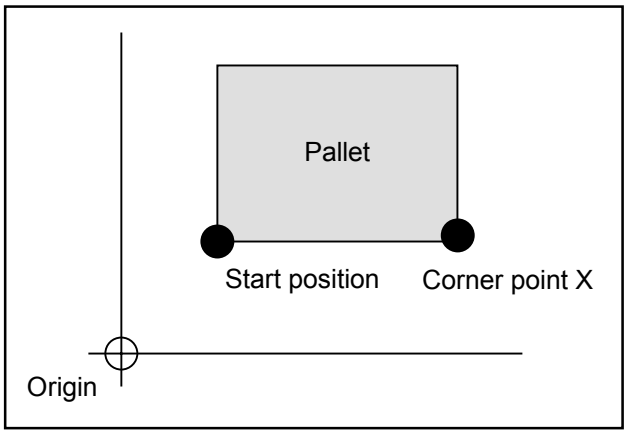
← Point number

- In case of direct setting of position data, the screen displays as shown below. Press ◀ and ▶ keys to set a operating unit and Numeric keys to set the position data. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]E   00
 X±0000.00  Y±0000.00

X Position
```

- When specify the start position by a point number, the data for a unit not specified as an axis unit does not have any meaning. If the position data of an axis unit is not set (when data is ××××.××), it is impossible to perform the palletizing operation. Setting of position of corner point is valid for PNT (three point setting) pattern only.

Figure 16-11: Position of Corner Point X



16.1.4.8. Setting Position of Corner Point Y

- For setting of position of corner point Y, use a screen to set point number or a screen for direct position data setting as shown below. Press ▶ key to indicate the cursor, then use ▲ and ▼ keys to switch the screen of register number or position data.

- In case of using point number, the screen displays as shown below. Use Numeric keys to select a point number that indicates the position of corner point Y. SET key is to enter the data and MODE key is to cancel the data.

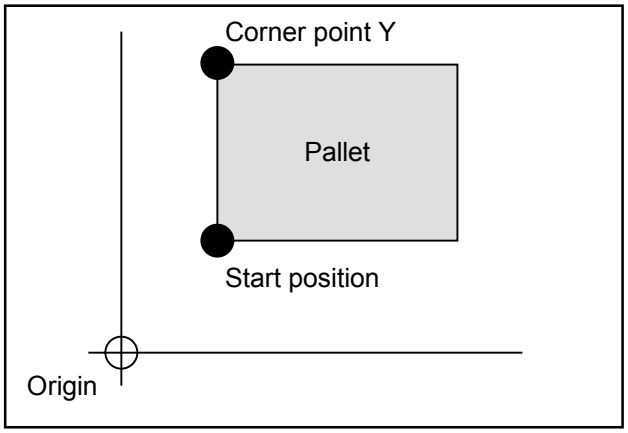
```
[PAL]E  00
 P0000

Y Position
```
← Point number

- In case of direct setting of position data, the screen displays as shown below. Press ◀ and ▶ keys to set a operating unit and Numeric keys to set the position data. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]E  00
 X±0000.00  Y±0000.00

Y Position
```

- When specifying the start position by a point number, the data for a unit not specified as an axis unit does not have any meaning. If the position data of an axis unit is not set (when data is ××××.××), it is impossible to perform the palletizing operation. Setting position of corner point Y is valid for PNT (three point setting) pattern only.

Figure 16-12: Position of corner point Y



16.1.4.9. Setting Number of Position Intervals

- For setting number of position intervals, use a screen to set point number or a screen for direct setting of position interval as shown hereunder. Press ▶ key to indicate the cursor, then use ▲ and ▼ keys to switch screen of register number or position data.

- In case of setting point number, the screen displays as shown below. Use Numeric keys to set a point number that indicates pallet size. SET key is to enter the data and MODE key is to cancel the data.

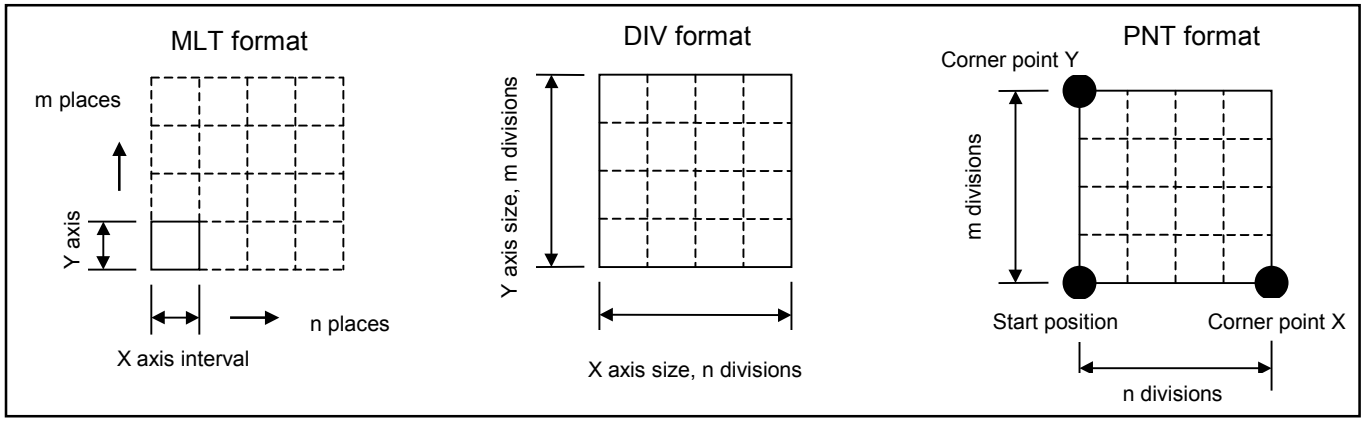
```
[PAL]E   00
 P0000

 Size
```
← Point number

- In case of direct setting of position interval data, the screen displays as shown below. Press ◀ and ▶ keys to set a operating unit and Numeric keys to set number of position interval data. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]E   00
 X000     Y000

 Size
```

- When specifying the number of position intervals by a point number, the data for a unit not specified as an axis unit does not have any meaning. If the position data of an axis unit is not set (when data is ××××.××), it is impossible to perform the palletizing operation. The data is a number of divisions in DIV format (Divided sides format) and PNT format (three corners format), while the data for MLT format (Multiple regular interval) is a multiple value.

Figure 16-13: Position setting format

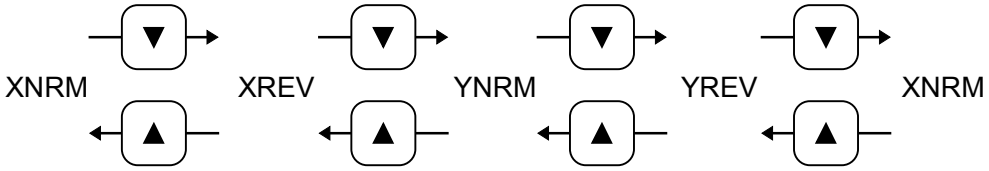


16.1.4.10. Setting Moving Order

- The screen shown below appears to set moving orders. Press ▶ key to indicate the cursor, then use ▲ and ▼ keys to set moving orders. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]S   00
 XNRM

 Move pattern
```

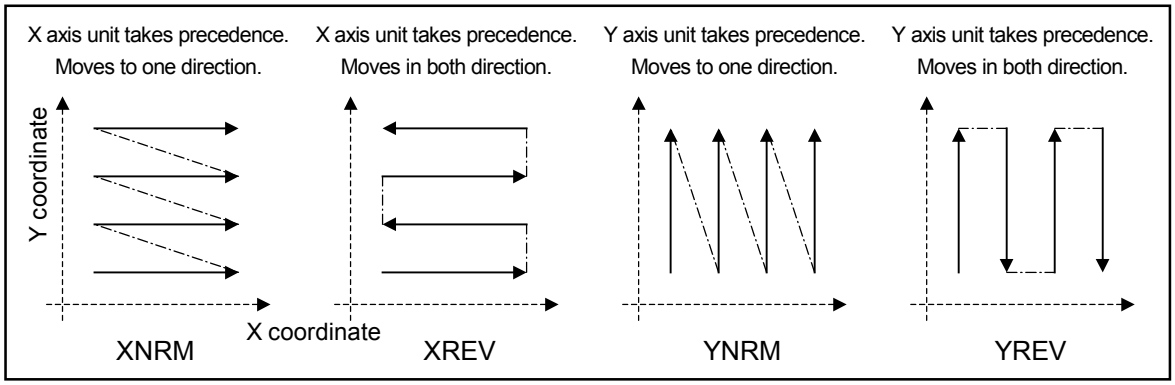
- Scroll screens by ▲ and ▼ keys. The order of screens is shown below.



- Code list of moving order.

  XNRM ------ X axis unit takes precedence. Moves to one direction.
  XREV ------- X axis unit takes precedence. Moves in both direction.
  YNRM ------ Y axis unit takes precedence. Moves to one direction.
  YREV ------- Y axis unit takes precedence. Moves in both direction.

Figure 16-14: Moving direction

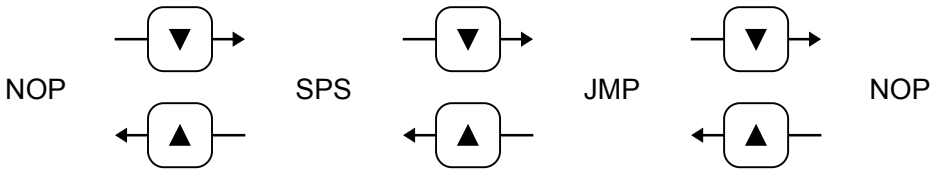


16.1.4.11. Selection of Jump Format

- The screen shown below appears to set jump format. Press ▶ key to indicate the cursor, then select jump format using ▲ and ▼ keys. SET key is to enter the data and MODE key is to cancel the data.

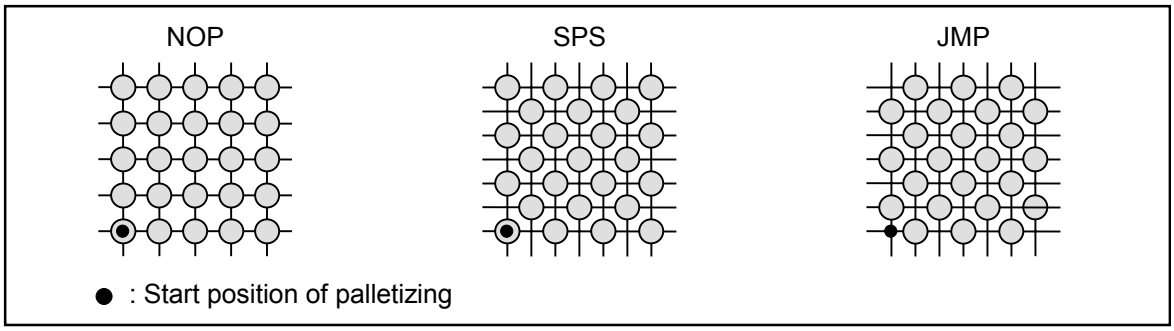
```
[PAL]E   00
NOP

Point Jump
```

- Scroll screens by ▲ and ▼ keys. The order of screens is shown below.



- Code list of Jump format:
  NOP ----- No jump
  SPS ------ Jump (Positioning to start position)
  JMP------ Jump (No positioning to start position)

Figure 16-15: Jump format

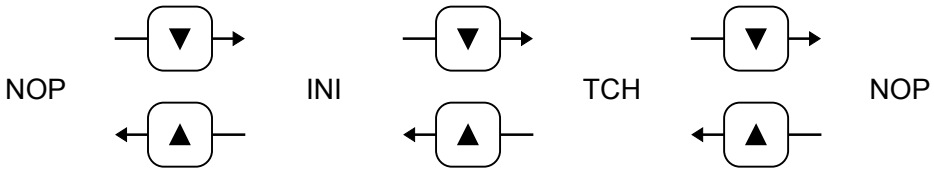


16.1.4.12. Format Selection of Coordinate Data Development

- The screen shown below appears to select format of coordinate data development. Press ▶ key to indicate the cursor, then select a format of the data development using ▲ and ▼ keys. SET key is to enter the data and MODE key is to cancel the data.

```
[PAL]E   00
 NOP

 P register
```

- Scroll screens by ▲ and ▼ keys. The order of screens is shown below.



- Code and the format are described hereunder.

  NOP No data development

  : Develops data based on a palletizing pattern number in the middle of palletizing operation. No point register is used.

  Select this format for normal palletizing operation.

  INI Data development (with initializing)

  : Develops palletizing position data to plural point registers that start from a specified register number. In INI format, the data is developed during execution of the command (PALI) for initialization of palletizing.

  TCH Data development (without initializing)

  : Develops palletizing position data to plural point registers that start from a specified register number. In TCH format, data is developed when F2 key is pressed in "Selecting screen 1 of palletizing setting process" (Refer to 16.1.3). It is possible to change and use the developed data.

  When an actual palletizing position is off from a programmed position, use this format to change the position from developed palletizing position.

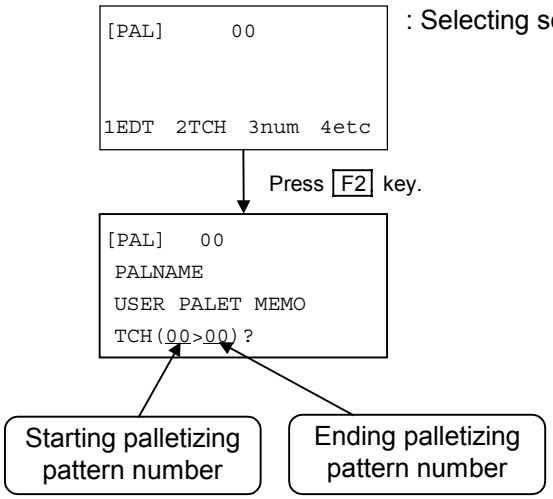
- When the format of data development is INI or TCH, use a point number to specify a point number that starts the data development as shown below. Move the cursor by ◀ and ▶ key to the position of point number and set the number by Numeric keys.

```
[PAL]E   00
 INI        P0000

 P register
```

Caution *: When the format of data development is INI or TCH, write coordinate data to a point register.*
If the teaching data have been written to the register, it will be overwritten. When teaching, keep some room of the data registers for coordinate data development.

16.1.5. Development of Palletizing Data

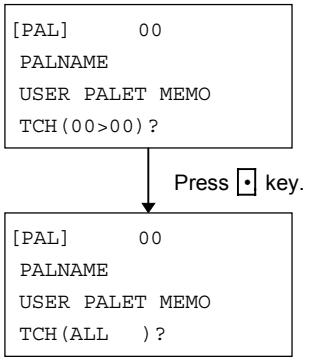
- When TCH format is set in the format selecting screen of developing coordinate data (refer to "16.1.4.12. Format Selection of Coordinate Data Development.") in setting process of palletizing operation, pressing F2 in "Selecting screen 1 of palletizing setting process" (refer to "16.1.3. Setting Palletizing Pattern.") gives the screen below and enables to set the range of palletizing pattern on which the data development is performed. Pressing SET key starts the development.



- The developing range shall be set from the starting (a palletizing pattern number currently displayed) to the ending. (The same number of the starting is displayed for ending number. You can enter a desired number here.)
  You can change the range by Numeric key or ▲ and ▼ keys. Set directly a number of cursor position using Numeric key or set using ▲ and ▼ keys which gives increase/decrease of number by one for every pressing.

  * If a larger ending number than the existing palletizing pattern number is specified, the data development ends at the largest number of existing setting.

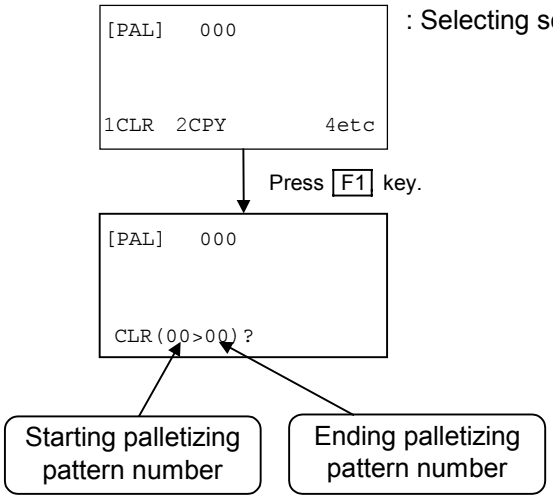
- The cursor moves directly to the starting number when F1 key is pressed, while it moves to the ending number when F2 key is pressed. Pressing • key sets all palletizing pattern number as the range and the screen appears as shown below. To terminate this state and back to numeric setting, press Numeric keys.



Caution : *When TCH format is set for coordinate data development, store the point register data after the developments. Otherwise they will be lost as the power is turned off. Be sure to store the data as well as the teaching data before the power is turned off. Refer to "15.1.4. Saving Point Data."*

16.1.6. Clear Palletizing Data

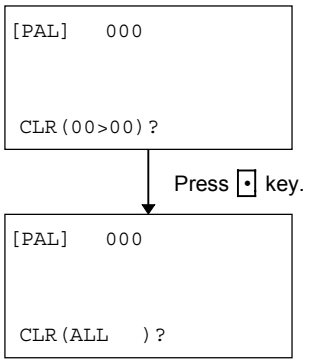
- When you press F1 key (CLR) in the setting screen 2 of process of palletizing setting, the screen shown below appears and you can clear a palletizing pattern data. (refer to "Setting Palletizing Pattern Data.") On the last line, specify a palletizing pattern number, then press SET key for deleting its data. (change to none-data state.)



- The range to be cleared shall be set from the starting (a palletizing pattern number currently displayed) to the ending. (The same number of the starting is displayed for ending number. You can enter a desired number here.)
  You can change the range by Numeric key or ▲ and ▼ keys. Set directly a number of cursor position using Numeric key or set using ▲ and ▼ keys which gives increase/decrease in number by one at every pressing.

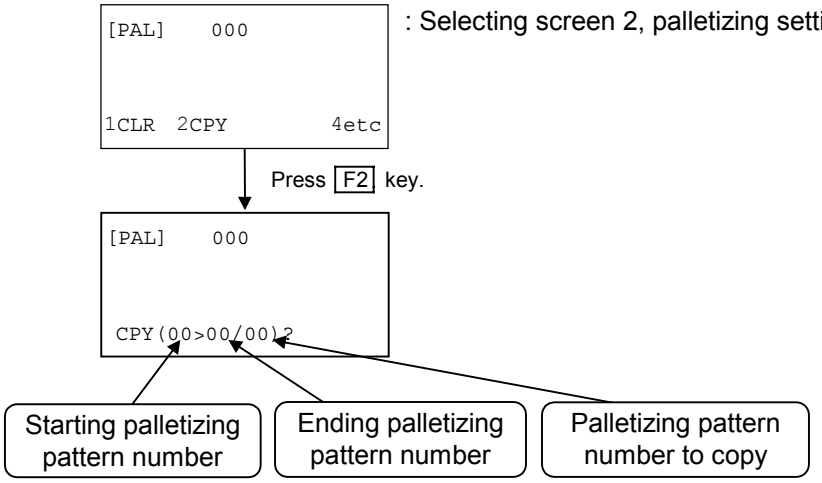
  * If a larger ending number than the existing palletizing pattern number is specified, the largest number is regarded as the end number and the data is processed accordingly.

- The cursor moves directly to the starting number when F1 key is pressed, while it moves to the ending number when F2 key is pressed. Pressing • key sets all palletizing pattern number and the screen appears as shown below. To terminate this state and back to numeric setting, press Numeric keys.



16.1.7. Copy Palletizing Data

- When you press F2 key in the selecting screen 2 of process of palletizing setting ("Setting Palletizing Pattern Data."), the screen shown below appears and you can copy a palletizing pattern data. On the last line, specify a palletizing pattern number, then press SET key for copying its data.



- The range to be copied shall be set from the starting (a palletizing pattern number currently displayed) to the ending. Set a number as the top palletizing pattern number of the range to copy. Move between 'the ending number to be copied' and 'the starting number to copy' using ◀ and ▶.

  You can set each number by Numeric key or ▲ and ▼ keys. Set directly a number of cursor position using Numeric key or set a number using ▲ and ▼ keys which gives increase/decrease in number by one at every pressing.

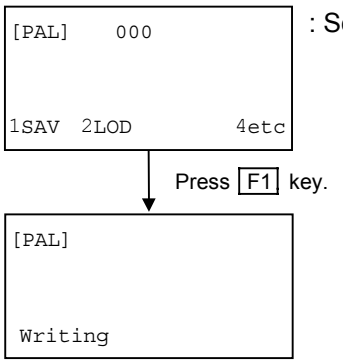
  * The same number is on the ending palletizing pattern number and starting the palletizing pattern number to copy. Set a desired number to start copy.
  * If a larger ending number than the existing palletizing pattern number is specified, the data development ends at the largest number of existing setting.
  * When the copy range is less than the copied range, the over flown data will be lost.
  * When the copy and copied range overlap each other, the previously written data will remain in the area.

16.1.8. Saving Palletizing Data

- Pressing F1 key in the selecting screen 3 of the process of palletizing setting, you can save the editing result to the flash memory. (Refer to "16.1.3. Setting Palletizing Pattern Data.")

- Pressing F2 key in the selecting screen 3 as same as above, you can read out the data in the flash memory.

- Turning off the power before saving the results of editing, you will lose them. The following screens appears during writing to the flash memory.



: Selecting screen 3, palletizing setting process

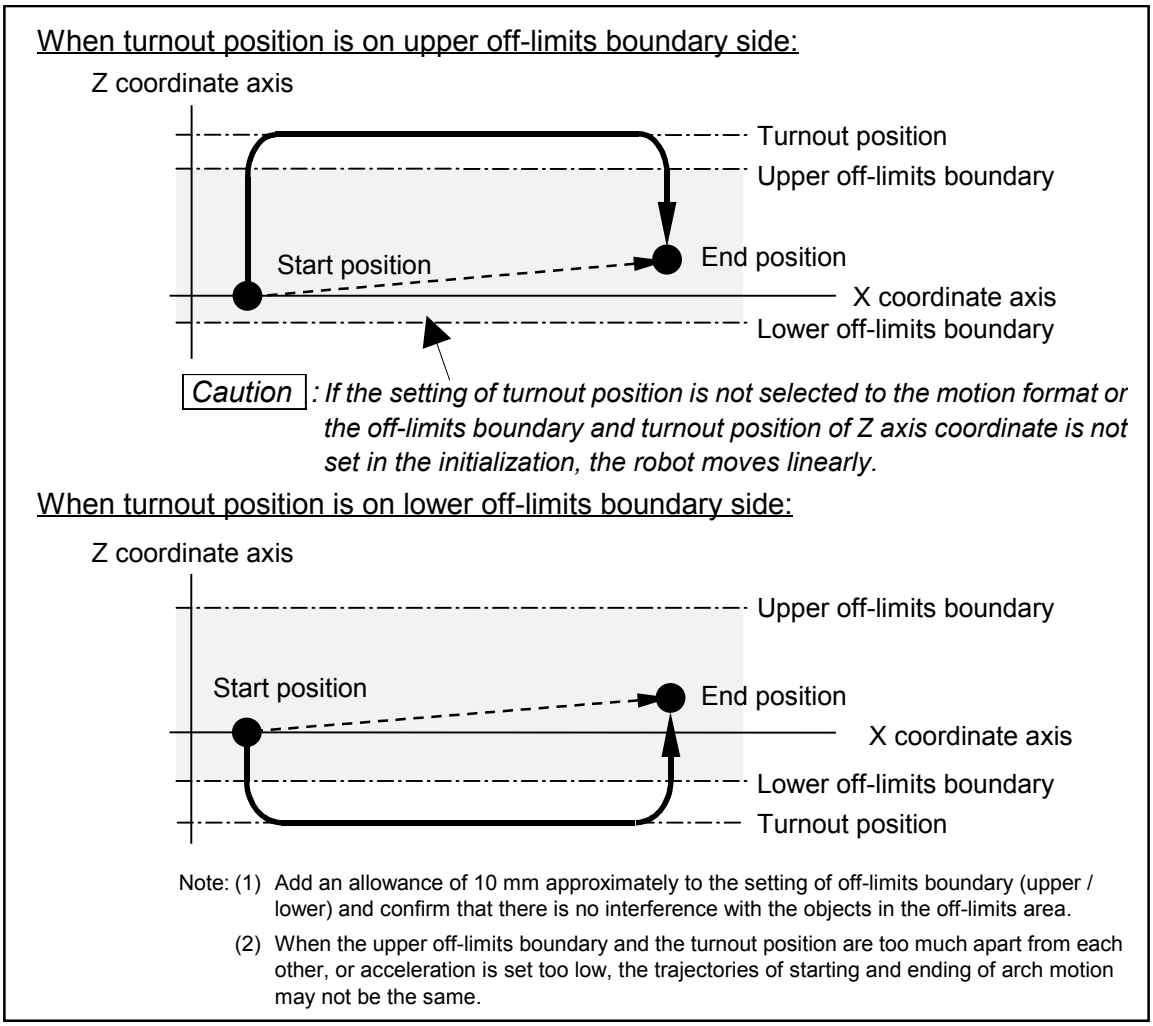
Caution : *Do not turn off the power while "Writing" is on the display. Otherwise memory error alarm will be given.*

16.2. Arch Motion (Malti-axis combination only)

- When a linear motion cannot be performed because an obstacle blocks the motion trajectory, you can program an arch motion by setting an off-limits area. The off-limits area is set by the upper off-limits and lower off-limits boundaries.

- The robot moves to Z axis coordinate direction from the start point, then moves to X axis coordinate direction along the turnout coordinate after it clears upper (or lower) off-limits boundary and moves back in Z axis coordinate direction to the destination.
  (Refer to Figure 16-16.)

Figure 16-16: Outline of arch motion



- The arch motion starts by "MOV" command. Arch motion becomes active if the motion format E (sets turnout.) is specified.

- The arch motion is valid only when Z axis off-limits boundaries and a turnout position is initialized.
  The off-limits area and turnout position of R axis do not function.

  * Refer to "9.3.4. Parameters for Coordinates and Position."

- Set the units to X and Z axis units in case of 2 axes control, while YZ or XYZ for 3 or 4 axes controller.

  * Refer to "9.5. Parameters for Unit setting" for more details.

- Refer to "15.3.6. Arch Motion in Two Axes Motion."

16.2.1. Initial Setting

- Set the off-limits boundary and the turnout position referring to "9.3.4. Parameters for Position and Coordinates."

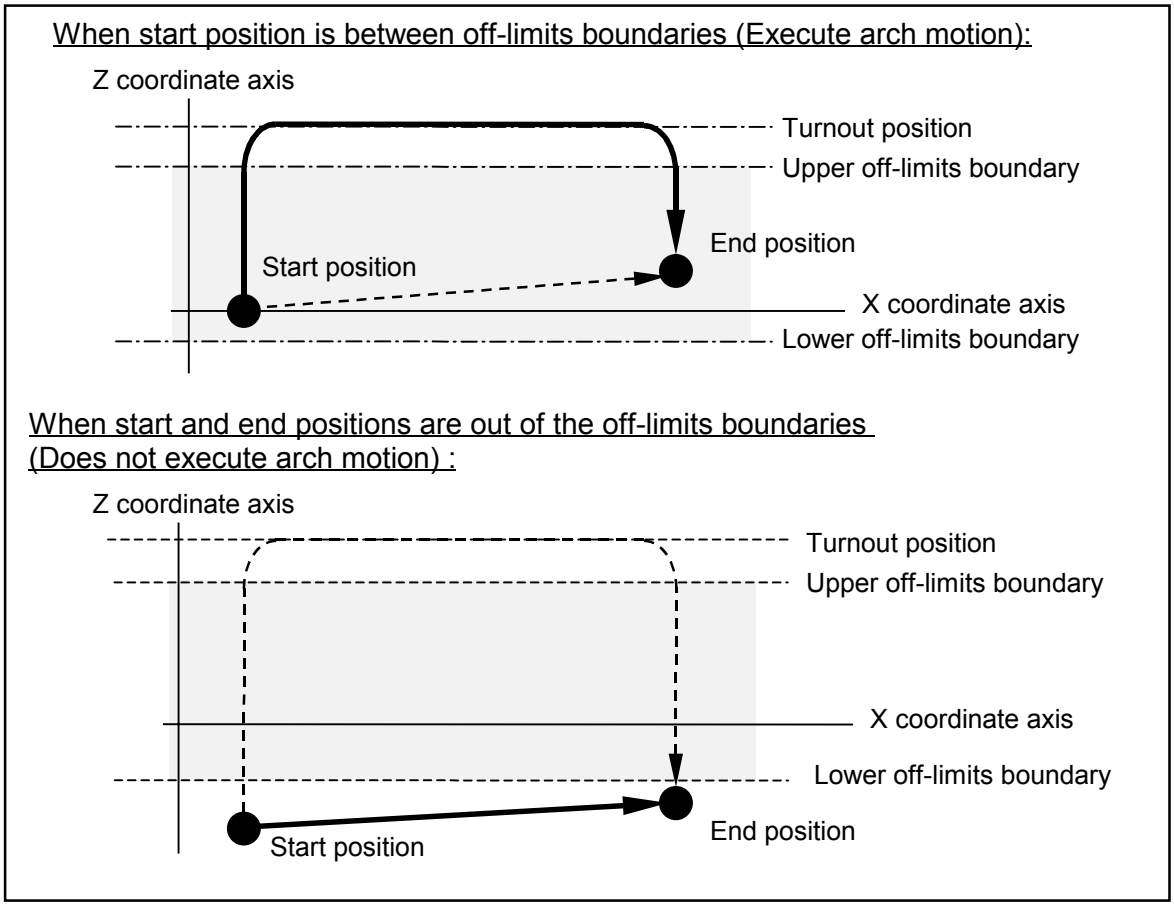
Table 16-2: Off-limits / turnout parameter

Parameter	Description	Unit	Setting range	Factory set
Escape (upr.Z)	• Set the upper off-limits boundary. • Setting of motion range of the arch motion using Z axis unit.	mm	-9999.99 ~ 9999.99	0
Escape (lwr.Z)	• Set the lower off-limits boundary. • Setting of motion range of the arch motion using Z axis unit.	mm	-9999.99 ~ 9999.99	0
Escape (pos.Z)	• Set turnout position of Z axis. • Setting of motion range of the arch motion using Z axis unit.	mm	-9999.99 ~ 9999.99, OFF	OFF
Escape (upr.R)	• Reserved	–	–	–
Escape (lwr.R)	• Reserved	–	–	–
Escape (pos.R)	• Reserved	–	–	–

- The motion to move to the turnout position starts when the start or end position is between the upper and lower off-limits boundary.

- When the start and end positions are out of the boundary, the motion is linear, not arch motion, to the end position.

Figure 16-17: Start position and off-limits boundaries



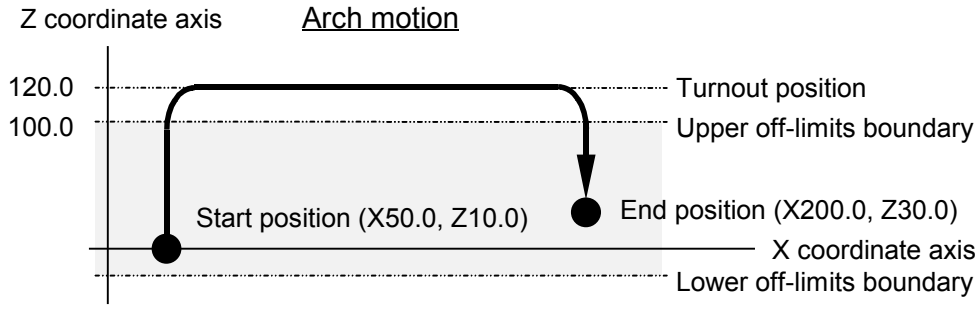
16.2.2. Programming

- Program an arch motion referring to "15. Programming."
- Set ESCZ command to change off-limits boundary and turnout position in the middle of an operation.

  Note : ESCZ is a command to change the off-limits boundary and the turnout position. It is not valid unless an arch motion is set initially (&E).

- Command to specify the arch motion is one of the motion format to MOV command. Set the turnout position (&, E) of a motion format active.

Figure 16-18: Example of program



Program	Description
	<Start: Move to start position.>
MOV X0050.00 Z0010.00 S0100.0 A0.5 B0.5	Move to start position (50.0, 10.0) from current position. (can be anywhere) (Resultant speed: 100 mm / s, Acceleration/ deceleration: 0.5 m / s2, Absolute coordinate format)
	<Set parameter for interpolation>
SPD S0600.0 A3.0 B3.0	Set motion speed and acceleration/deceleration for interpolation hereunder. (Resultant speed: 600 mm / s, acceleration / deceleration: 3.0 m / s^2)
	<Set parameters for arch motion.>
ESCZ #0000.00 #0100.00 #0120.00	Lower off-limits boundary: 0.0 mm. Upper off-limits boundary: 100 mm, Turnout position: 120.00 mm
	<Arch motion start>
MOV X0200.00 Z30.0 &E	Start arch motion to a position of absolute coordinates (200.0, 30.0)
	<End of program>
END CSTP	

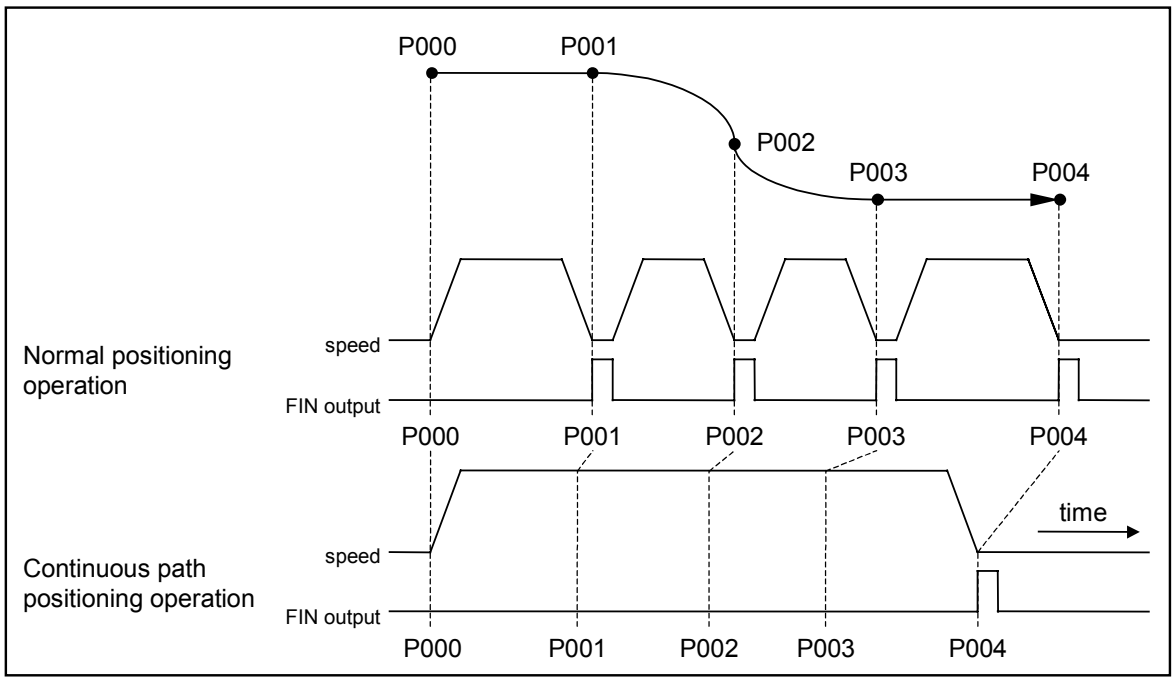
16.3. Continuous Path (Malti-axis combination only)

- Continuous path is a motion in constant speed without changing speed, acceleration and deceleration at the passing points. This is to be used for an operation that need constant speed such as sealing or deburring.
  It is possible to change motion speed and to give an output command to the general output port in the middle of a continuous path operation.

- This motion is performed in two dimensions by 2 axes controller and in three dimensions by three or four axes controller.

- There are some restrictions in a Continuous path program. Be sure to program correctly referring to "16.3.2. Restrictions on Program of Continuous Path."

Figure 16-19: Continuous path



16.3.1. Command for Continuous Path

- There are two commands.
  CPS: Start continuous path.
  CPE: End of continuous path.

- Execute a motion command between CPS and CPE under constant speed.

- Commands that can be set between CPS and CPE are shown in Table 16-3.

Table 16-3: Commands that can be set between CPS and CPE

Command	Description
MOV	Linear interpolation
MOVM	Continuous linear interpolation
CIR	Circular interpolation
ARC	Circular arc motion command
OUT	General output command
SPD	Setting command of speed, acceleration and deceleration.

- See "15.3.5. Continuous Path Two Axes Motion."

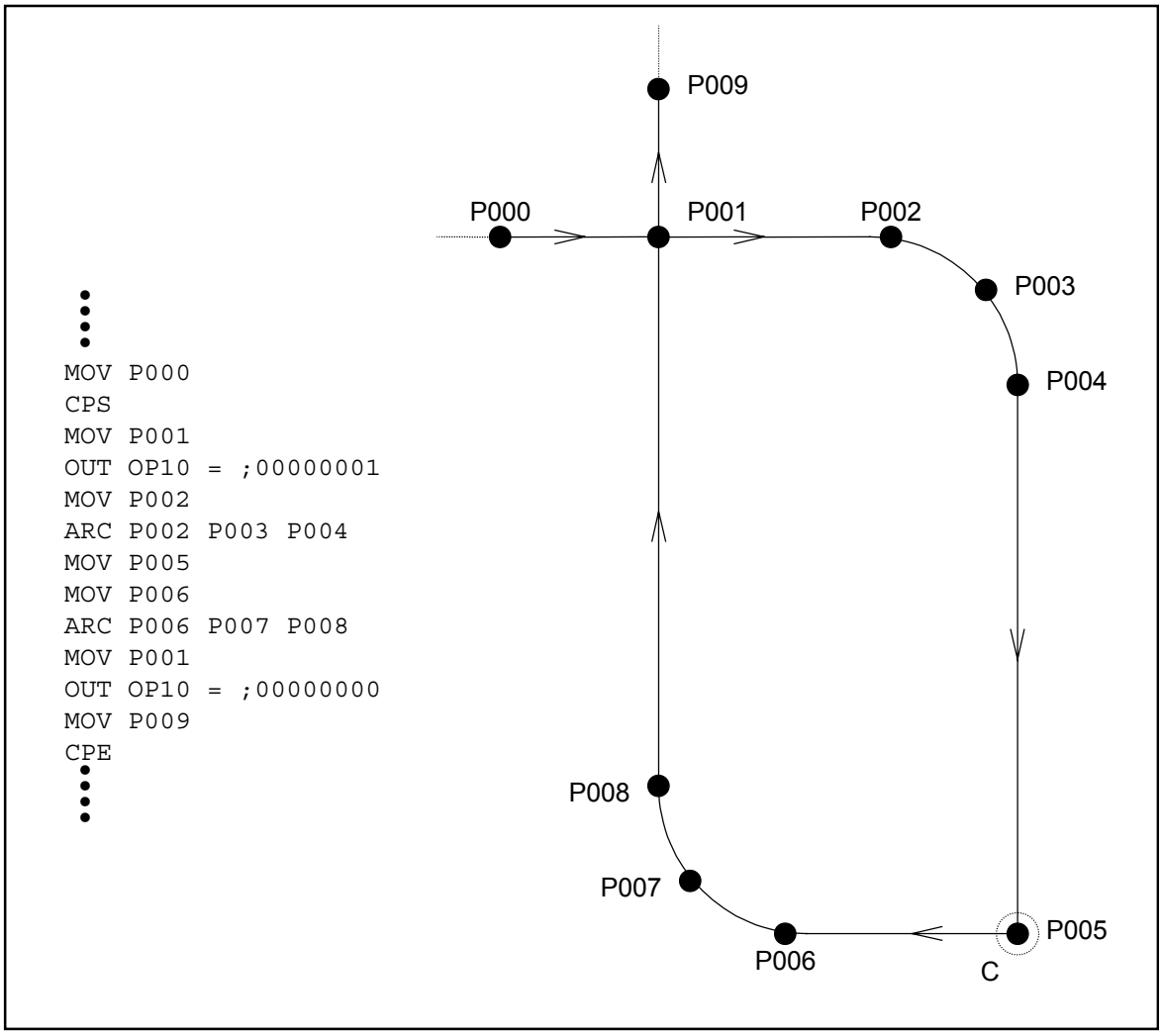
16.3.2. Restrictions on Program of Continuous Path

16.3.2.1. Change Direction of Linear Motion

- When changing direction of linear motion, connect two linear trajectories with an arc and make it smooth as possible. The actual motion trajectories are not accurate under limited power of motor as the speeds of each axis unit will be discontinuous at the inflection points. When the motion speeds differ greatly, an excessive current command to the motor might be given, and consequently, an excessive current alarm arises.

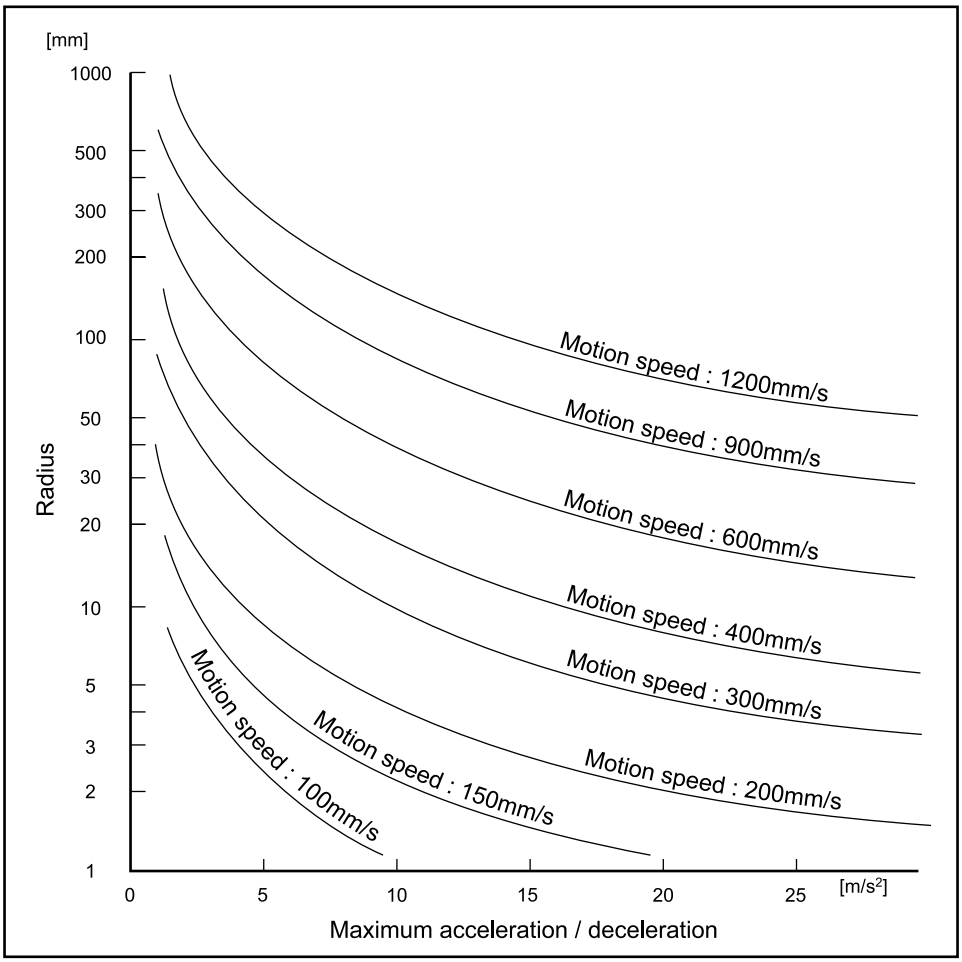
- Size of radius to connect two linear motions:
  Please find out the size of radius from allowable acceleration and deceleration referring to Figure 16-20.
  See "19.1.3. Precautions against Use of Module Main Unit" for allowable acceleration / deceleration of module main unit.

Figure 16-20: Restrictions on program of continuous path: In case of changing direction of linear motion



Note: Connect two linear lines with an arc at point C. See Figure 16-21 for size of radius.

Figure 16-21: Size of radius



16.3.2.2. Number of Steps between CPS and CPE Commands

- Number of steps that can be set between CPS and CPE commands are up to 100. If 101 or more steps are set between them, a program error alarm arises.

- Be aware that the number of steps depends on a program command.

  1) Normally steps for MOV, ARC, and CIR is one. However if motion speed is set, it increases two more steps.

     [Example] MOV P0000 S300: Number of steps: 3

  2) For MOVM command, number of steps number of continuous points.

     [Example] MOVM P0001 P0050 : Number of Steps: 50

16.3.2.3. Others

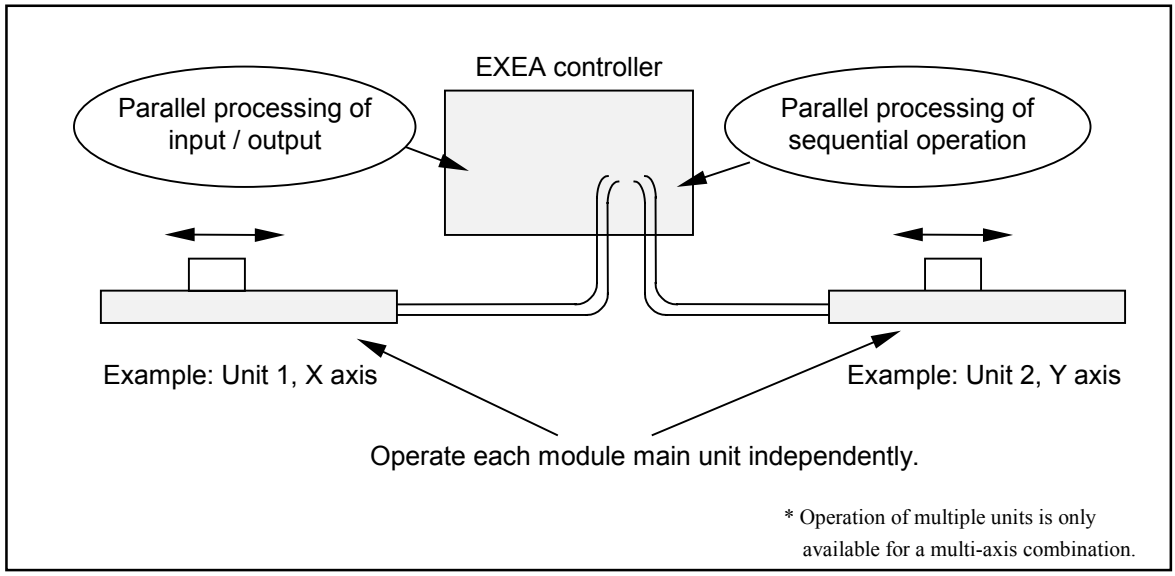
- Operating time (calculation): Before starting a continue path operation, it is required to calculate total motion distance. If there is a lot of steps, it looks as if the robot is pausing momentarily.

- The timing to output a command during a continue path motion is delayed to the time when the robot is passing the previous point. Delay of time differs ways of programming. The delayed time is longer for a multitask operation.

16.4. Multitask

- It is possible to execute several programs simultaneously in a multitask operation, such as controlling input/output or controlling sequence while operating a module main unit.
- Multitask operation can be performed by a main program as a trunk and sub programs as branches and leaves. However, you may conduct a simple multitask operation by setting a parallel operation as a motion format of MOV command. (See "15.3.8. Multitask" for program examples.)
- You may perform a multitask operation by specifying it through a program command.
- It is required to operate a multitask when operate the several units, that are separated into individual units at initial setting, simultaneously and independently. It is possible to operate individual axis units without separating from the unit as well.
- Multi-axis EXEA controller can handle up to 16 programs in the multitask operation, while a single axis EXEA controller can handle 4 programs.

Figure 16-22: Outline of multitask



16.4.1. Multitask Program

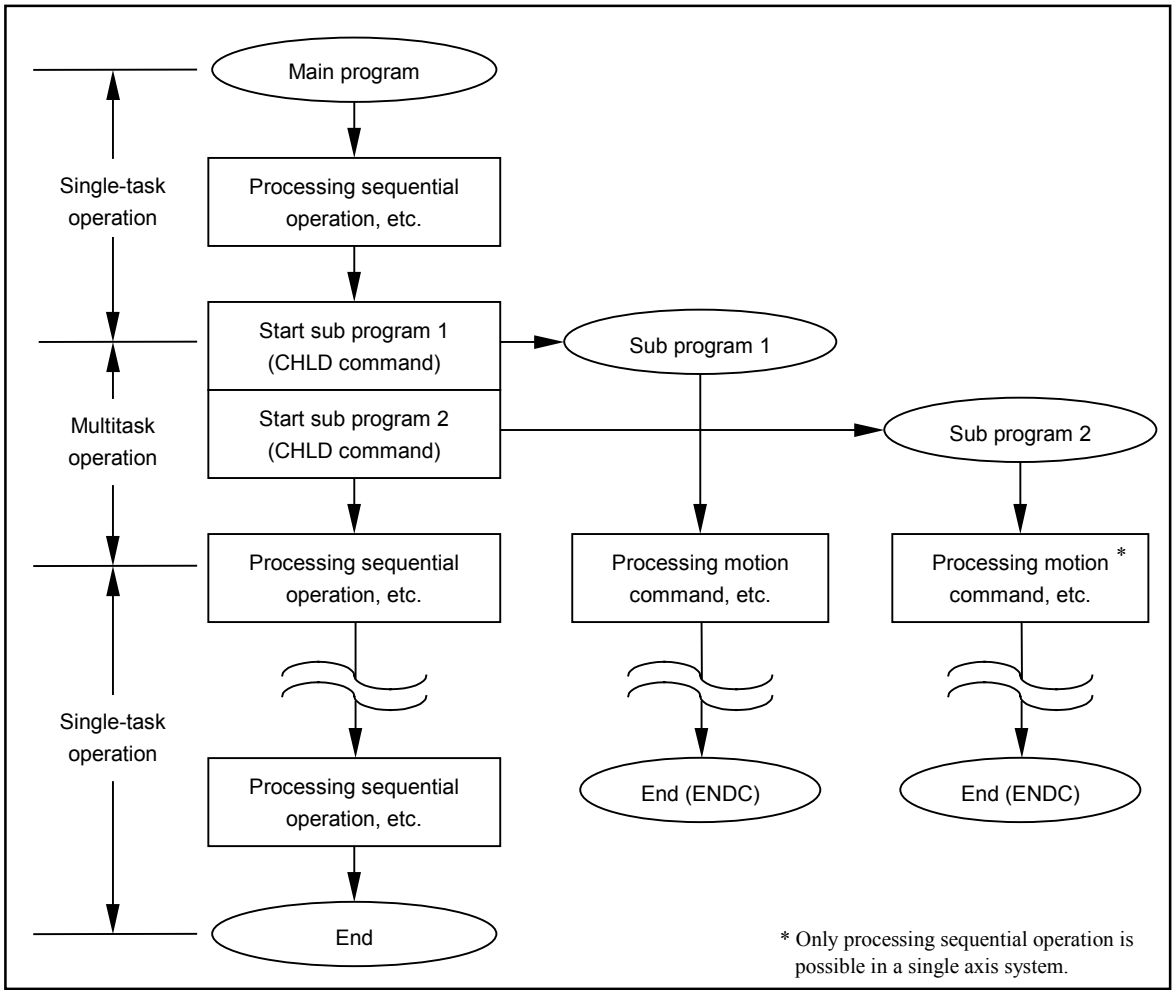
- There are two ways to operate the multitask.
  - ◊ The program comprises a main program as a trunk and sub programs as branches and leaves.
  - ◊ Set the parallel processing of the motion format to a motion command.

* In case of a multi-axis combination it is possible to operate a multitask by a direct operation as well. Refer to "16.5. Direct Operation."

16.4.1.1. Main and Sub Programs

- Multitask program comprises a main program and sub program. Sub-programs can be up to 15 for a multi-axis combination while up to 4 for a single axis system. A multitask starts by a start command from a main program to sub-programs.
- CHLD command starts a sub program.

Figure 16-23: Outline of main and sub programs



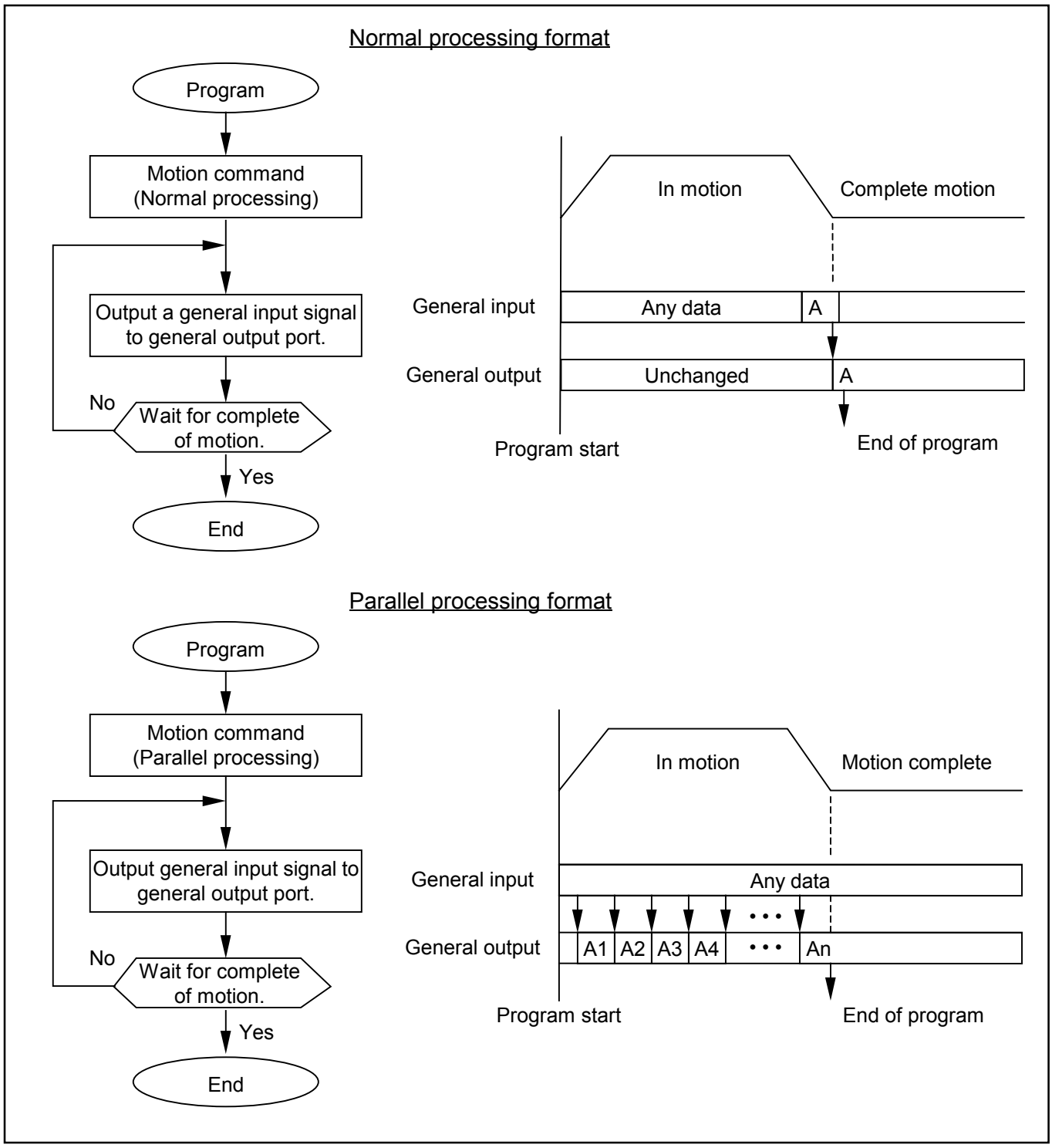
16.4.1.2. Parallel Processing

- It is possible to select parallel or normal processing as motion format of the motion command such as MOV. Selecting the parallel processing as motion format enables to operate a simple multitask in one program.

  Normal processing (Motion format code W) : Program waits to execute next step for a completion of current motion command.

  Parallel processing (Motion format code P) : As soon as the motion command starts, the next step of the program starts. However, the system has to wait till completion of current motion to start a motion unit.

Figure 16-24: Outline of parallel processing



16.5. Direct Operation

- Specify a point register number through I / O in external control mode, then move the robot to a position set on the point register.
- As the programmed operation is set to the shipping set, it is necessary to change setting to operation mode. They cannot be used together.
- The same motion parameters of programmed operation are used for those of a direct operation. Refer to "9.3.1. Parameters for Programmed Operation " for the contents.

1 Set parameters for direct operation

- Table 16-4 shows settings of internal parameter for direct operation

Table 16-4

Parameter	Factory set	Setting for direct operation	Chapter to be referred
RUN mode	PRG	POS	9.7. PMD Parameters for PMD Setting
Input IP10(IB100)	PROG0	* UNTN0	9.8. Parameters for Input Signal Format
Input IP10(IB101)	PROG1	* UNTN1	
Input IP10(IB102)	PROG2	* UNTN2	
Input IP10(IB104)	PROG4	POSN0	
Input IP10(IB105)	PROG5	POSN1	
Input IP10(IB106)	PROG6	POSN2	
Input IP10(IB107)	RSRV	POSN3	
Input IP11(IB110)	USER	POSN4	
Input IP11(IB111)		POSN5	
Input IP11(IB112)		POSN6	
Input IP11(IB113)		POSN7	
Input IP11(IB114)		POSN8	
Input IP11(IB115)		POSN9	
Input IP11(IB116)		POSN10	
Input IP11(IB117)		POSN11	

* Not set for a single axis system.

2 Input / Output of direct operation

Table 16-5

<table>
<tr><th rowspan="2">Signal name</th><th colspan="2">Connector Pin No.</th><th rowspan="2">Description of function</th><th></th></tr>
<tr><th>CN3</th><th>P1-EXT.I/O</th></tr>
<tr><td>SVON</td><td>8</td><td>–</td><td>Input: Servo on</td><td>On: Servo on OFF: Servo off</td></tr>
<tr><td>RUN</td><td>7</td><td>–</td><td>Input: Start programmed operation</td><td>Rising signal detection from off to on starts direct operation.</td></tr>
<tr><td>UNTN0 *
~
UNTN2</td><td>–</td><td>10,
27, 28</td><td>Input: Selection of unit numbers.</td><td>Set number of units (point registers) to be started simultaneously by binary digit from input of UNTN 0 to 2.
0 : 0FF 1 : ON
<table><tr><th rowspan="2">Input</th><th colspan="5">Number of units to be operated.</th></tr><tr><th>1</th><th>2</th><th>3</th><th>· · ·</th><th>8</th></tr><tr><td>UNTN2</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>UNTN1</td><td>0</td><td>0</td><td>1</td><td>· · ·</td><td>1</td></tr><tr><td>UNTN0</td><td>0</td><td>1</td><td>0</td><td></td><td>1</td></tr></table></td></tr>
<tr><td>POSN0
~
POSN11</td><td>–</td><td>3 ~ 8,
20 ~ 26</td><td>Input: Selection of Point register number.</td><td>Select point registers to be operated by binary digit from POSN0 to 11.
0 : OFF 1 : ON
<table><tr><th rowspan="2">Input</th><th colspan="5">Point register No.</th></tr><tr><th>0</th><th>1</th><th>2</th><th>· · ·</th><th>3999</th></tr><tr><td>POSN11</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN10</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN9</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN8</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN7</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN6</td><td>0</td><td>0</td><td>0</td><td>· · ·</td><td>0</td></tr><tr><td>POSN5</td><td>0</td><td>0</td><td>0</td><td></td><td>0</td></tr><tr><td>POSN4</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN3</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN2</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>POSN1</td><td>0</td><td>0</td><td>1</td><td></td><td>1</td></tr><tr><td>POSN0</td><td>0</td><td>1</td><td>0</td><td></td><td>1</td></tr></table></td></tr>
<tr><td>MTN</td><td>34</td><td>–</td><td>Output: In motion</td><td>It is closed when the robot is in motion.
It opens at end of motion.</td></tr>
<tr><td>HOLD</td><td>5</td><td>–</td><td>Input: Hold</td><td>When it is on during direct operation, the robot decelerates and stops.
★Command to start direct operation is not valid while it is on.</td></tr>
<tr><td>STOP</td><td>26</td><td>–</td><td>Input: Forced stop</td><td>When it is on during direct operation, the robot decelerates and stops.
★Command to start direct operation is not valid while it is on.</td></tr>
<tr><td>HOLDA</td><td>13</td><td>–</td><td>Output: On hold</td><td>It is closed while HOLD input is on.</td></tr>
</table>

* Not used for a single axis system.

3 How to use UNTN signal. (Multi-axis combination only)

- UNTN signal specifies number of units (point registers) to be started simultaneously.

[Example]

P0000 X0000.00 Y0000.00 U1
P0001 X0010.00 Y0000.20 U2
P0002 X0030.00 Y0040.00 U3

If there are teaching point data of Unit 1 ~3 in point numbers P0000 ~ P0002 as shown above, designation of a point number to 0 (zero) by POSN0 ~ 11 signals and setting the number of units to 3 by signals UNTN0 ~ 2 start P0000 and P0001 simultaneously and put the main units of U1 and U2 module in parallel operation (multitask operation).
When number of units is set to 2, P0000 and P0001 start simultaneously and U1 and U2 module start parallel operation (multitask operation). Error (F5: unit overlap) arises if the setting of a unit number of point register is duplicated.

4 Procedure of direct operation

1) Turn Servo on referring to "17.4.2. Servo on."

2) Perform Home return referring to "17.4.3. Home Return." Once the Home position is established you do not need Home return in daily bases.

3) Confirm the state of following input and output.
Input off: CSTP, STOP and HLD
Output: MTN opened
You cannot operate direct operation when STOP and HLD inputs are on. Direct operation cannot be started again if MTN output is closed (in the middle of direct operation).

→ Wait for end of direct operation or force to terminate direct operation by inputting STOP. (Turn off STOP input after end of direct operation.)

4) Specify a desired point number to be operate by selecting input of a point number.

5) Turn on RUN input. Direct operation starts and MTN output is closed.

Note: Refer to "17.4.5. Stopping Programmed Operation" for procedure to stop direct operation. However, CSTP input (cycle stop) is invalid in direct operation. (It is ineffective.)

6) MTN output is opened at the end of direct operation.

Note: (1) Selecting input of point number is invalid once a direct operation starts. (during MTN output is closed.) For changing number of point register, it is necessary to wait till the end of direct operation or force to terminate a direct operation by STOP input.

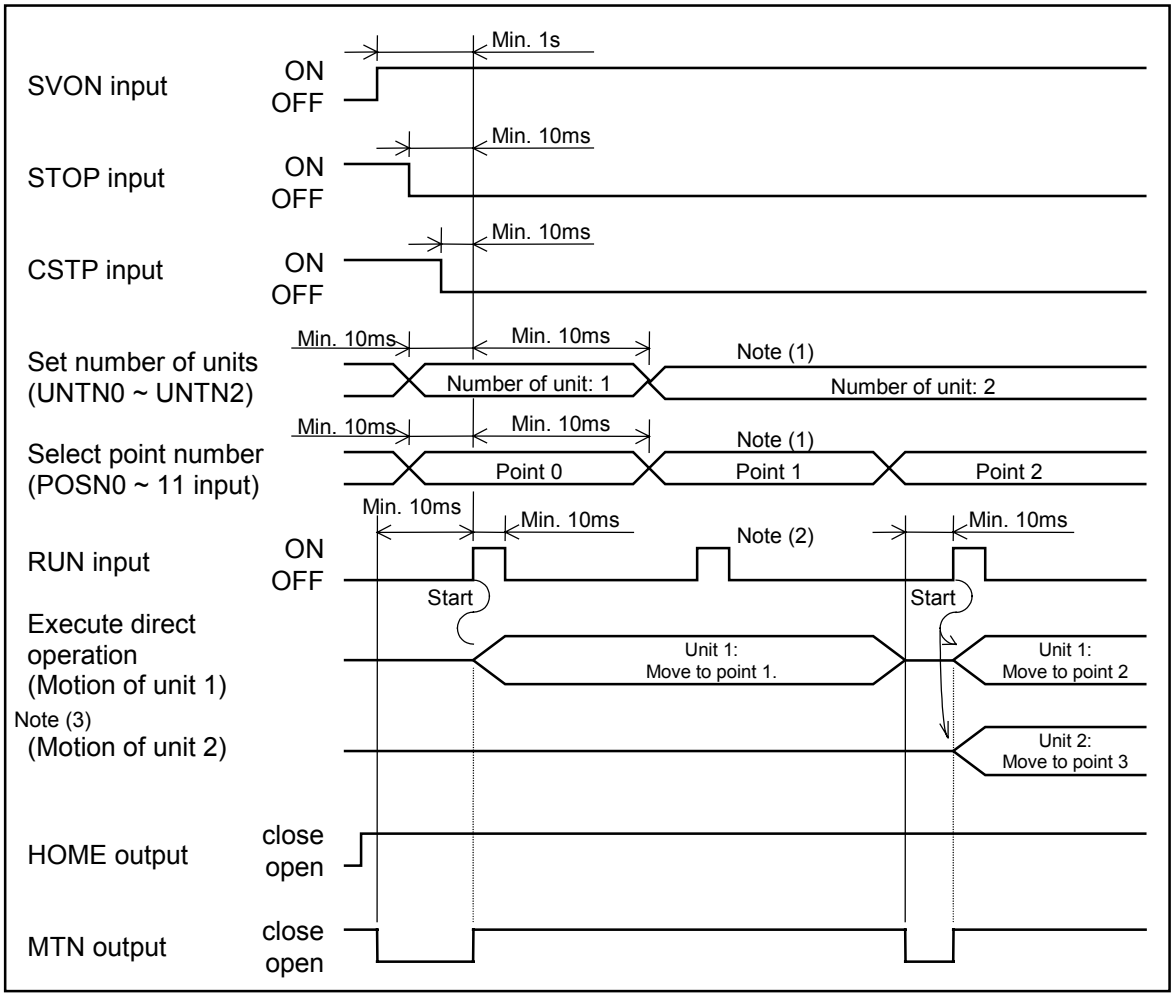
(2) RUN input is detected as ON by recognition of rising current (effective at changing point of off to on). Direct operation cannot start when RUN is on from the beginning.

(3) It is possible to make operation in relative coordinates when (I) (relative coordinates format) is specified to point register setting. Otherwise absolute coordinates format is set. Refer to "15.1. Teaching."

5 Sequential timing of control input / output

- Perform a direct operation following the timing as shown in Figure 16-25.

Figure 16-25



Note: (1) Selection of point number and setting number of units are invalid until end of direct operation once it started (MTN output close).

(2) RUN input is invalid until end of direct operation once it started (MTN out put close).

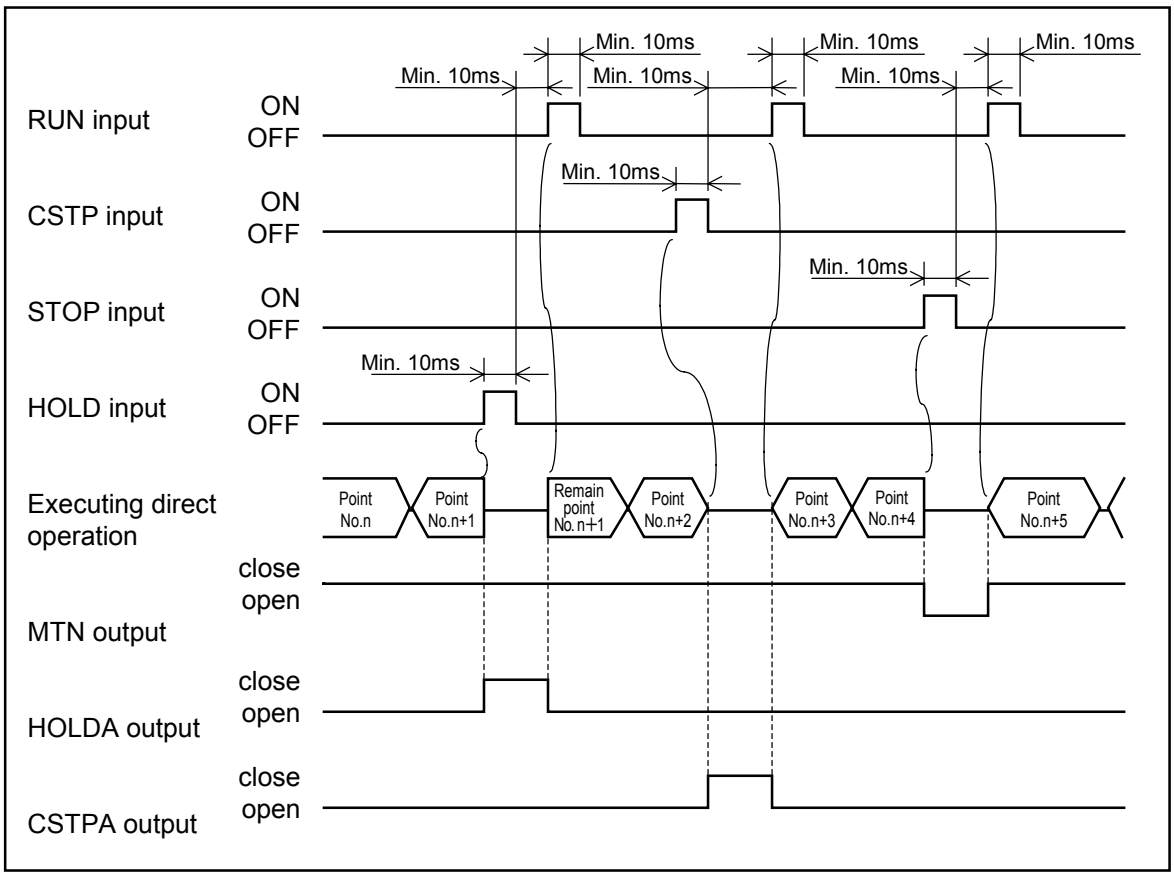
(3) It does not need for a single axis system.

* In the example shown above, the point numbers "0" to "1" are set to Unit 1, while the point number "3" is set to Unit 2.

6 Timing to handle stop commands in the middle of direct operation

- Be sure to observe the timing shown in Figure 16-26.

Figure 16-26



(Blank Page)

17. Operation of Robot Module

- This chapter describes how to operate the robot module system as well as the operational functions that are necessary to operate the system.

  Danger : *Do not enter in the guard fence (provided by user) when the robot system is operating. If you go beyond the guard fence for unavoidable reason, be sure to stay out of working area of module main units and set the speed of module main units to a safety speed (250 mm/ sec or less). Be careful not to be hit or crushed by the robot.*

  Caution : *When you have prepared a new program or changed a program, we recommend to perform a trial operation to check if the system works as you have intended. Make sure to prevent the work piece or end effectors from being damaged by unexpected movement of the robot module system in the trial running. If the program does not work as intended, re-program it until it works properly.*

17.1. Power On and Off

◆ Procedure for turning power on

- See "8.1. Turn on Power" before turning on your robot module for starting up or re-starting after the system modifications.
- In this chapter, the procedure for turning on the robot module for daily operation is presented.

1) Check the cables and connectors for:
   - ◊ Loose or disconnected connectors. Deteriorated or damaged cables.
   - ◊ Wrong axis connection.

2) Check main units for:
   - ◊ Loose bolts.
   - ◊ End effector. (provided by the user)

3) Check that no person nor devices which could be injured or damaged by the module is in the working area of the robot.

4) If everything is satisfactory, turn on DC 24V external power supply first, when DC 24V external supply is used.

5) Turn on main power of EXEA controller if everything is normal.

◆ When turn off power

1) Stop operation of the robot.

◊ During operation using teaching box : Press STOP key.

Note: STOP key is not valid unless programmed operation mode is set.

◊ During external control : Turn on STOP input signal.

◊ During remote control : Input STOP command.

2) Make sure that the slider of main unit (or the main unit when it is moving) stops completely, then turn the servo off.

◊ During the operation using teaching box : Press OFF key.

◊ During external control : Turn off SVON input.

◊ During remote control : Input SVOF command.

3) Turn the power off.

Caution : If the power is turned off in the "SVON" state, the slider (or the main unit when it is moving) of a vertical axis may drop for 5 to 10 mm due to the time lag of excitation of motor brake solenoid.

17.2. Checking before Start Operation

- Before start operation, check the robot system.
- Determine check items centering operational safety in accordance with user's applications.

Danger : *Observe the two checking items described below. If failures of the emergency stop function is overlooked, serious danger could result because the stop functions would not work in an emergency.*

1) Execute jog operation referring to "17.3.2. Jog Operation."

◊ This is to check if the controller and motor system work properly. If the jog operation cannot be made because your application does not require the teaching box, check the systems by alternative way such as a programmed trial operation.

→ Do the systems work smoothly as intended? If not, take necessary measures referring to "14. Troubleshooting."

2) Check stopping functions

◊ Emergency stop function: Emergency stop key of teaching box
: EMST input of CN3 connector
: EMST command (remote control only)

→ Gets into "emergency stop" state when a procedure for emergency stop is executed during jog operation.

* For procedure for emergency stop and its state, refer to "11.3.1. Emergency Stop."

* When the system enters emergency state, F5 alarm (program error) will arise.

◊ Function of dead-man switch: The switch located on the side of teaching box.

→ During jog operation, the system gets in servo off state and stops operation when the dead-man switch is released.

★ Before start the operation, be sure to check that the stop functions work properly.

17.3. Operation by Teaching Box

- Teaching box allows you to execute Home return, jog and programmed operation.
- Procedures to perform operations using the teaching box are described hereafter.

Caution : *When operating the robot module through the teaching box, all control inputs (connector CN3), except EMST input, are invalid.*

→ If EMST input (B contact, normally close) is not ON, you cannot cancel the emergency stop and, thereby, the teaching box does not function.

→ Other inputs are invalid.

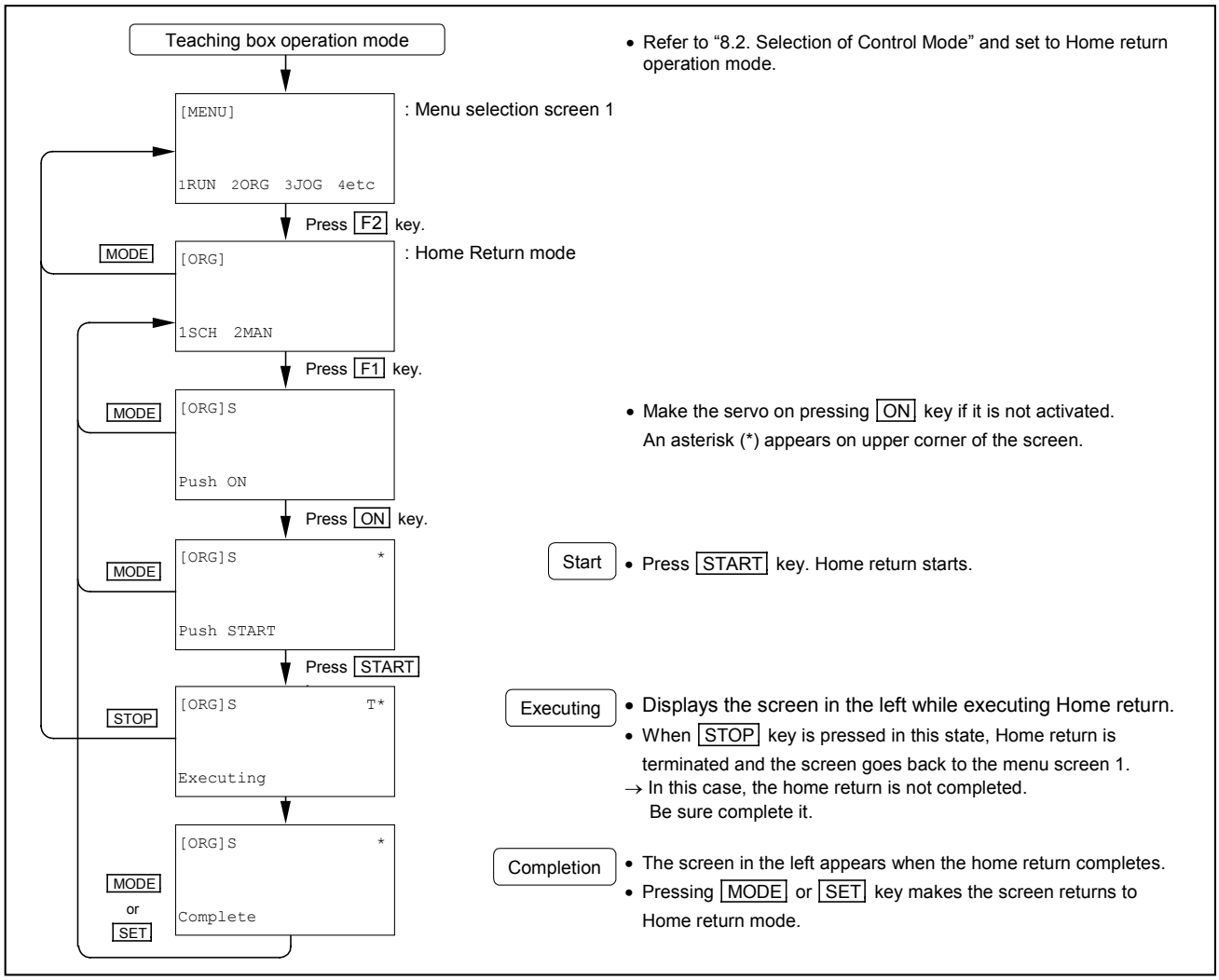
Example: Turning on of STOP input does not stop the robot.

* Same as in the external control mode, the control outputs are outputted during the operations through the teaching box.

17.3.1. Home Return

- Be sure to execute "Home return" when you use EXEA controller for the first time. Otherwise, the Home position of the coordinate cannot be set and you cannot perform the teaching by programmed operation nor jog operation. Once Home position is set, it is not necessary to perform Home return when the power is turning on again.
  [Exception of A5 alarm (Home return incomplete) or disconnection of cables.]

Figure 17-1



17.3.2. Jog Operation

◆ Only the teaching box can execute the jog operation.

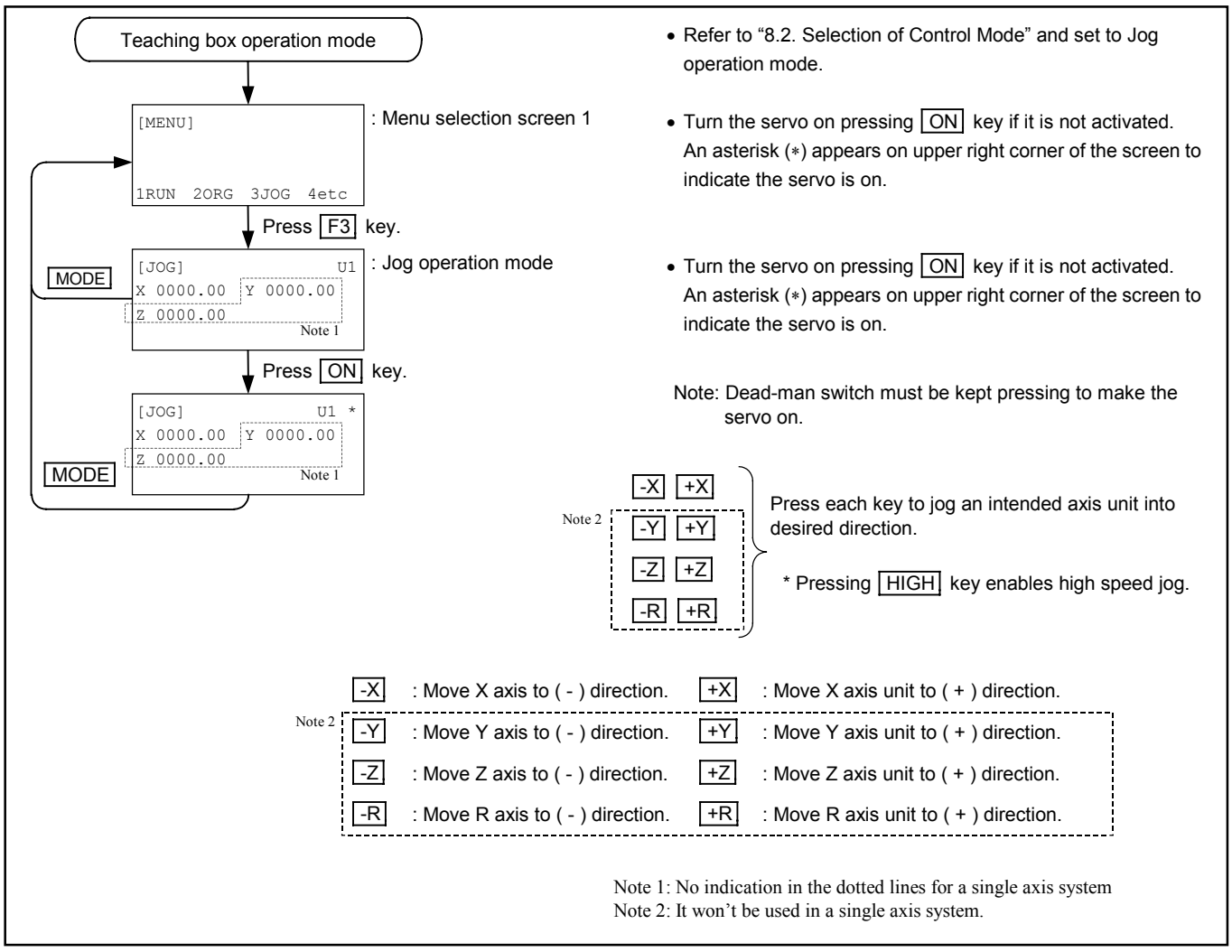
- Execute jog motion while pressing the jog keys (+X -X, +Y -Y ,+Z -Z, and +R -R). Releasing the jog key will stop jog motion.

  Note: If the teaching box has a dead-man switch, you must keep pressing the switch while executing the jog operation. Releasing the switch turns the servo off and stops operation.

- You cannot execute the jog operation for two or more axis units simultaneously. Operate an axis unit one by one.

- Press MODE key for termination of the jog. The screen goes back to the menu selection 1 screen.

Figure 17-2



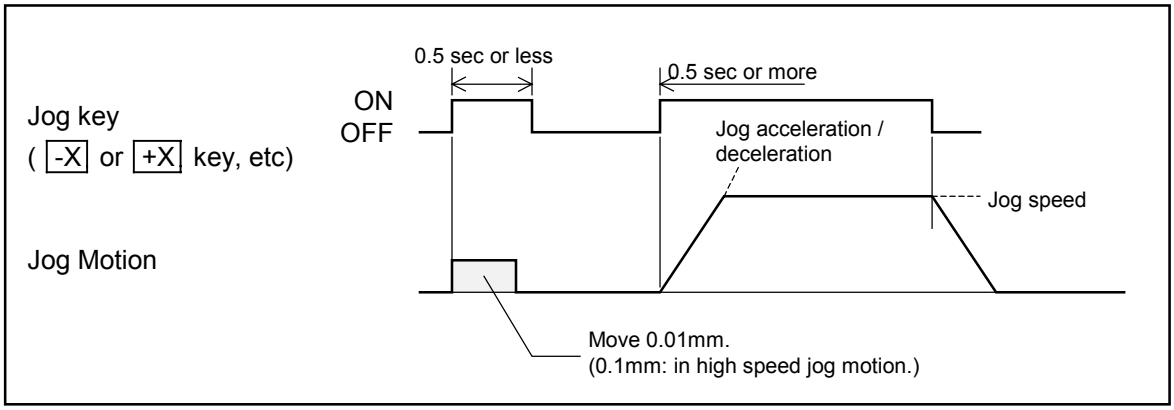
Caution : Directions of Jog operation are as follows.

Plus (+) : Motor side
Minus (-) : Opposite to motor side

However the direction is reversed if the coordinate parameter for reverse coordinate is set to "REV" (reverse). (Refer to "9.4.4. Parameters for Function Setting.")

17.3.2.1. Sequential Timing of Jog Operation

Figure 17-3



- Pitch feed of the jog operation is 0.01 mm when the jog keys are pressed for less than 0.5 seconds. Pitch feed becomes 0.1 mm when HIGH key and the jog keys are pressed simultaneously.

- The jog acceleration / deceleration, jog speed, and high jog speed must be specified to the initial setting. Refer to "9.3.3. Parameters for Jog Operation."

Note: When the response of the slider (start and stop) to on and off of the jog keys is dull, increase acceleration and deceleration of jog motion. It will give a crisp response. Overshoot may be observed if setting of acceleration / deceleration is too high. Set them to the safety speed (250 mm/s or less.)

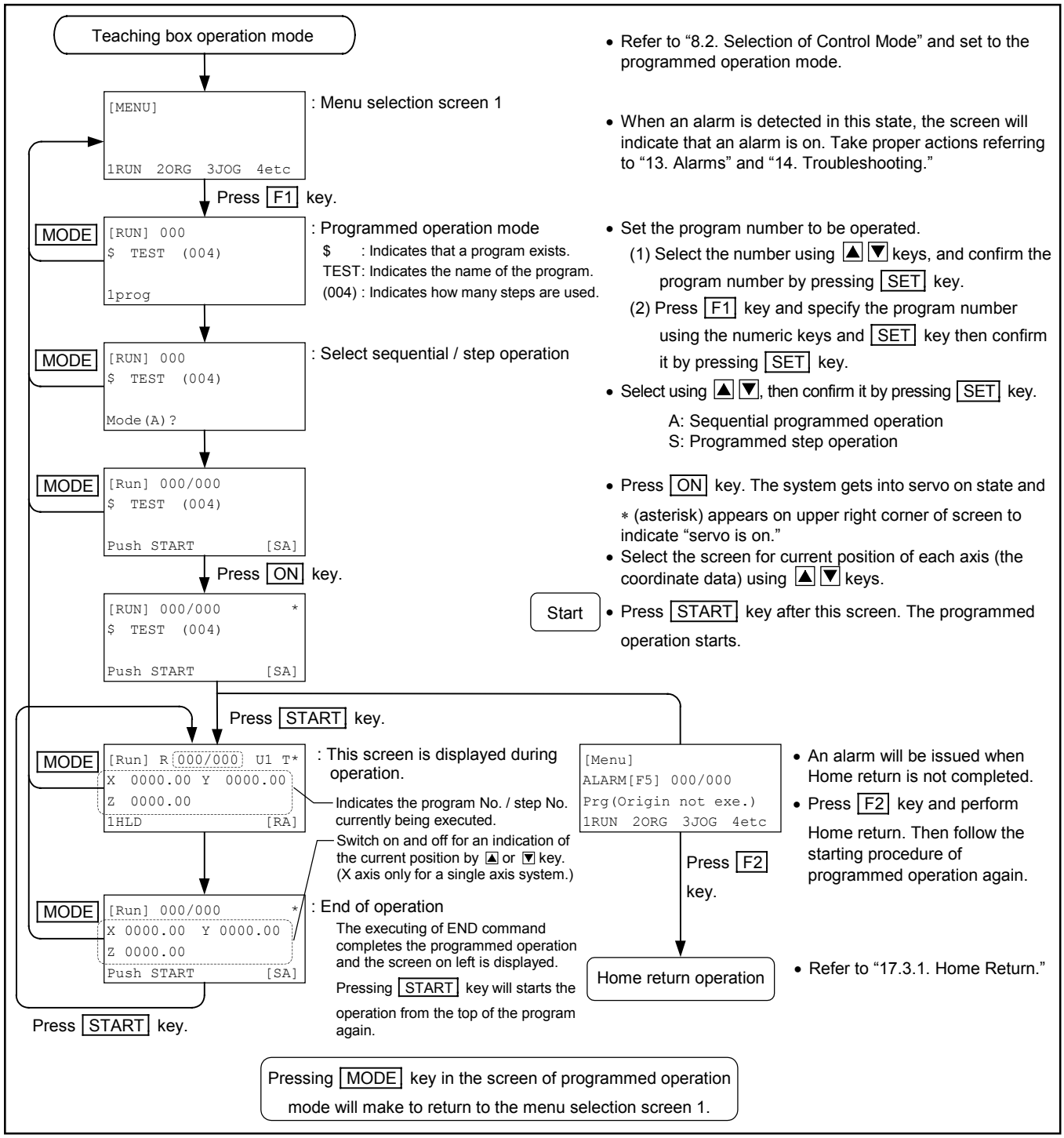
17.3.3. Programmed Operation by Teaching Box

17.3.3.1. Sequential Programmed Operation

- Once the robot module is started, the system continues sequential programmed operation to the end of the program. (to the END command of a program.)
- Step operation is available. Programmed step operation is to execute a step one by one by pressing [START] key.

◆ Starting sequential programmed operation

Figure 17-4



◆ Stopping sequential programmed operation

● Table 17-1 below describes the stopping functions of a sequential programmed operation.

Danger : *The stopping functions listed in Table 17-1, excluding the emergency stop, are valid for programmed operation only. In another operation mode, press EMG key to stop operation in an emergency as the other functions are not valid.*

Table 17-1

Function	Description of function	Procedure
Emergency stop	• Stops a programmed operation immediately and the motor gets into "servo off" state. * For emergency stop function, see "11.3.1. Emergency Stop."	Press EMG key.
Forced stop	• Stops a programmed operation immediately (when moving, decelerates and stops). The system goes back to the top of the program and waits. • The motor is in "servo lock" state. • Pressing START key starts the programmed operation from the top of program. **Note: Execution of "forced stop" resets the data of data registers and internal flags excluding the following.** **• Point coordinate data (P0000 ~ P3999)** **• General output data by OUT command.**	Press STOP key.
Cycle stop	• Stops after completing the step being executed. • The motor gets in "servo lock" state. • Pressing START key resumes the programmed operation from the next step. **Note: If an alarm arises, or STOP or EMG key is pressed during cycle stop, the state of cycle stop is removed and the programmed operation cannot be resumed.** *Caution : If case of a multi-axis combination, cycle stop is not effective during "continuous path" operation. It stops when the continuous path is completed.*	Press CYCSTOP key.
Hold	• Terminates motion command and decelerates and stops, and waits at the current position. • The motor gets in "servo lock" state. • Pressing START key resumes the subsequent instructions (completes motion command) and continues to execute following programmed operation. **Note: If an alarm arises, or STOP or EMG key is pressed during "hold" state, "hold" state is removed and remaining programmed operation cannot be resumed.**	Press F1 key.

Caution : *The display screen goes back to MENU screen by MODE key while a programmed operation continues. The system will immediately stop the operation as soon as the mode is changed to "external control" mode.*

17.4. Operation in External Control Mode

- This mode is to operate the system through control output / input (CN3) and general input / output (EXT. IO).
- Turning on the main power gets into this mode automatically.

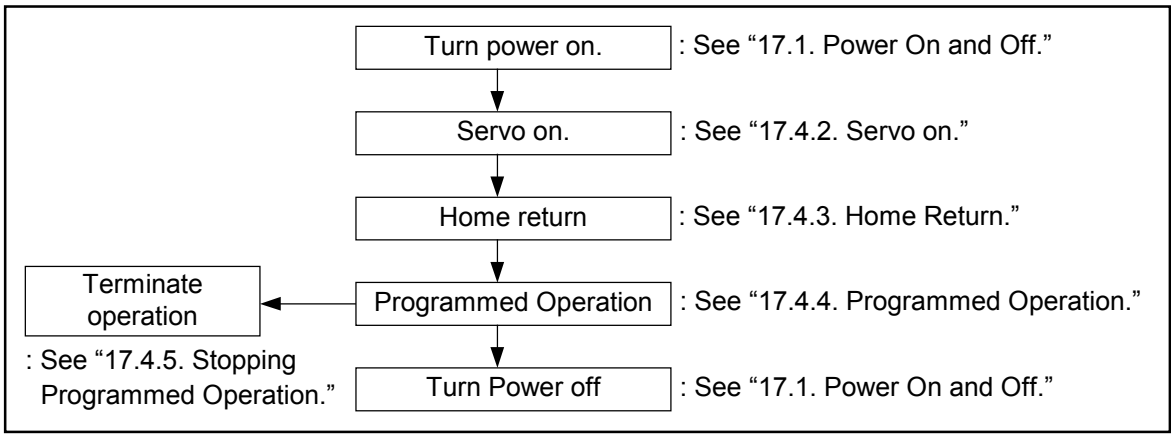
Caution *: In external control mode, the teaching box does not function except the following.*

1) Emergency stop : Pressing EMG key leads to an emergency stop state.
   * Refer to "11.3.1. Emergency Stop."
2) Mode selection : You may switch to teaching box operation mode using keys. Changing mode while executing the external mode will stop operation immediately (if the slider is in motion, it decelerates and stops) and the system gets into servo lock state.
3) Monitoring programmed operation :
   You can monitor the state of programmed operation using F1 to F3 keys.

Caution *: Turn input EREM off to control through CN3 connector.*

17.4.1. Operation Procedures in External Control Mode

Figure 17-5



17.4.2. Servo on

◆ Servo on Input / Output

Table 17-2

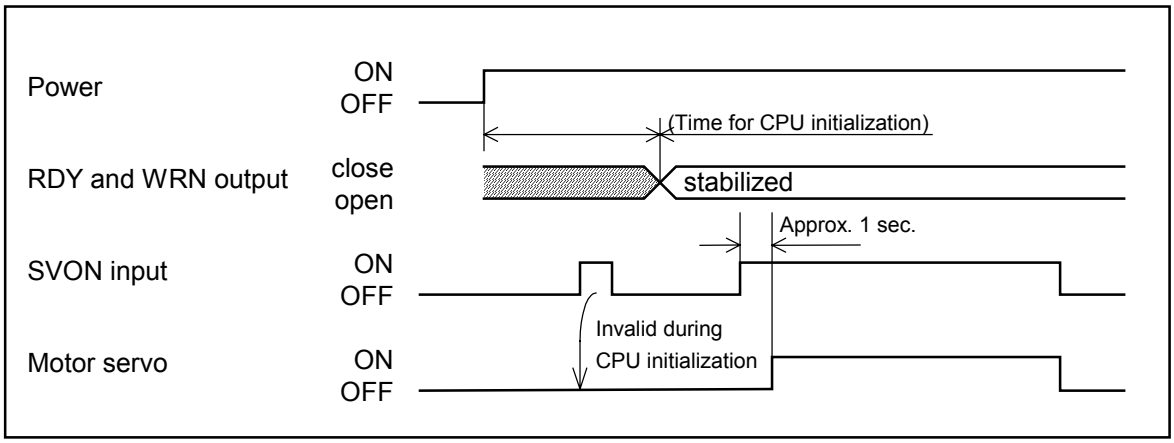
Signal	Pin No.	Description	
SVON	8	Servo on input	On : Servo on. Off: Servo off
RDY	35	Ready output	Opens at serious alarms. Normally closed.
WRN	16	Alarm warning output	Closes at minor alarms. Normally opened.

- The alarm outputs (RDY and WRN) are stabilized within a certain period of time after turning on power. (approximately 4 seconds for initializing CPU) Activate SVON input if no alarm is detected. The motor servo is activated and is ready to receive the operation commands.

  Caution : *If an alarm arises, take proper measures referring to "13. Alarms" and "14. Troubleshooting."*

- In inactive state of SVON, the position error counter is cleared.

Figure 17-6



Note: (1) ON and OFF keys of the teaching box cannot switch Servo on / off in external control mode.

(2) When SVON input is on, execution of "servo off" command in a program makes SERVO off.
When SERVON input is off, "servo on" command in a program is invalid.
(Servo on / off command in a program: SRV command is valid only when SVON input is active.)

(3) When SVON input is deactivated during a positioning operation (Home return or execution of motion command in a programmed operation), the positioning operation is terminated, the servo is off, then an alarm arises.

17.4.3. Home Return

- Please perform "Home return" for the first time you operate the system. Otherwise the system cannot perform the programmed operation because it cannot specify the position coordinate data.

Note: Home return shall be performed in the following occasions. Home return is not necessary in normal operation once the home position is established.

(1) At startup. (When the power is on for the first time.)
(2) More than 20 minutes are passed after disconnection of the controller cable.
(3) A battery (installed internally) for encoder back-up has discharged.
(4) The setting of module main unit type is changed.
(5) The power is turned off before the new setting in 4) is not saved. [SAV command in CTR mode.]
(6) Resetting home position is interrupted by an emergency stop command, an alarm or power failure.

◆ Input / Output of Home return operation

Table 17-3

Signal	Pin No.	Description	
SVON	8	Servo on input	On: Turns Servo on. Off: Turns Servo off.
HOS	25	Input: Start Home return	Rising signal detection to on starts Home return.
HOMS	14	Output: Completion of Home return	Closes when Home return is completed.
MTN	34	Output: Executing operation	Closes during the execution of programmed operation or Home return. Home return is enabled when this signal is opened. (while no programmed operation)
STOP	26	Input: Forced stop	A slider decelerates and stops when this signal is inputted during Home return. Home return cannot start while this signal remains on.

◆ Procedure of Home return

1) Turn the servo on referring to "17.4.2. Servo on."

2) Make sure that STOP input is off and MTN output is open. If STOP input is on, Home return cannot be executed. Also, if MTN is closed (means an operation is underway.) Home return cannot be executed.

→ Wait till the programmed operation ends or forcibly terminate the programmed operation by inputting STOP signal. (Deactivate STOP signal after completion of the programmed operation.)

3) Turn on HOS input. Home return starts.

◊ The motor decelerates and stops when STOP input is turned on during Home return. In such a case, as Home return is not completed yet, turn off STOP input and start Home return again.

◊ Home return will be terminated when EMST input is off during the operation.

* Refer to "11.3.1. Emergency Stop."

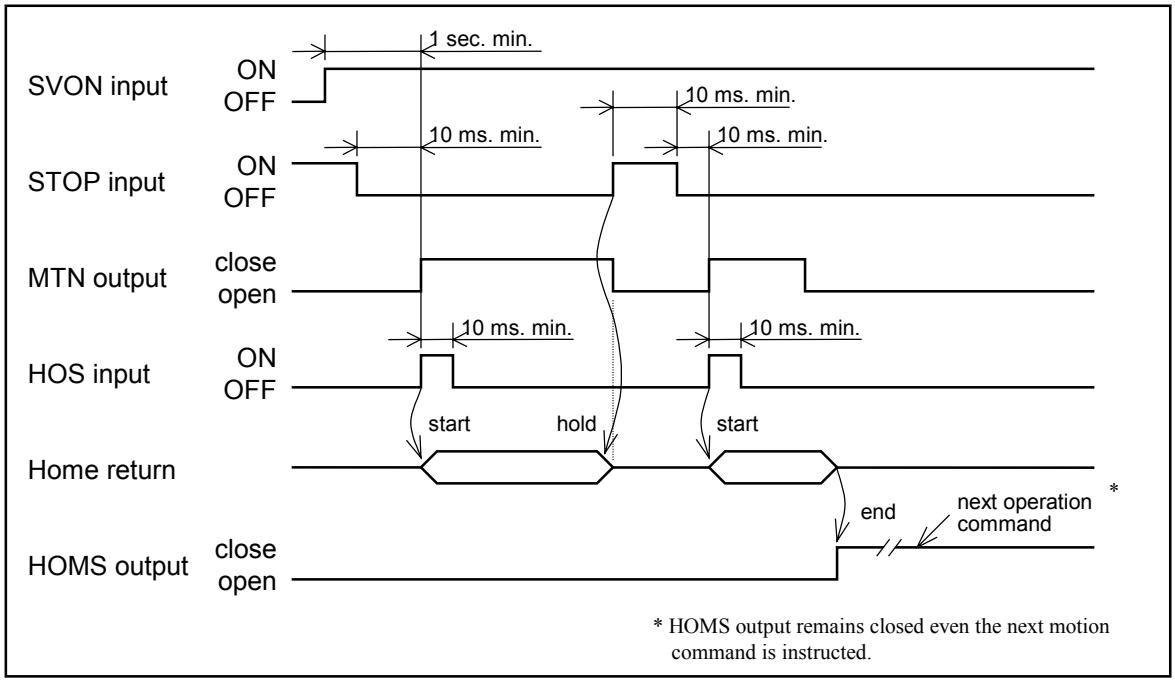
4) HOMS output will be closed at the end of Home return.

Note: (1) HOS input is detected at the rising edge of signal (effective at the rising point of off to on.). You cannot start Home return operation if HOS input is being on.
(2) HOMS output is not Home proximity signal. It is the signal for the end of Home return. Thereafter Home return has completed, it will be automatically closed after the initialization, even the power is turned off and turned on again.

◆ Timing of control input / output of Home return

- Observe input / output signal timing as shown in Figure 17-7.

Figure 17-7



17.4.4. Programmed Operation

◆ Input / Output for programmed operation

Table 17-4

<table>
<tr><th rowspan="2">Signal</th><th colspan="2">Pin No.</th><th colspan="2" rowspan="2">Description</th></tr>
<tr><th>CN3</th><th>P1-EXT.I/O</th></tr>
<tr><td>SVON</td><td>8</td><td>–</td><td>Input: Servo on</td><td>On : Turn Servo on. Off : Turn Servo off.</td></tr>
<tr><td>RUN</td><td>7</td><td>–</td><td>Input: Start programmed operation</td><td>Starts a programmed operation when rising edge signal turns on.</td></tr>
<tr><td>PROG0 to PROG6</td><td>–</td><td>8 to 10, 25 to 28</td><td>Input: Program No. selection</td><td>Specify a program number to be operated by PROG 0 to 6 inputs in binary numbers.
0 : OFF 1 : ON
<table><tr><th rowspan="2">Input</th><th colspan="5">Program No.</th></tr><tr><th>0</th><th>1</th><th>2</th><th>· · ·</th><th>127</th></tr><tr><td>PROG6</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>PROG5</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>PROG4</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>PROG3</td><td>0</td><td>0</td><td>0</td><td>· · ·</td><td>1</td></tr><tr><td>PROG2</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr><tr><td>PROG1</td><td>0</td><td>0</td><td>1</td><td></td><td>1</td></tr><tr><td>PROG0</td><td>0</td><td>1</td><td>0</td><td></td><td>1</td></tr></table></td></tr>
<tr><td>MTN</td><td>34</td><td>–</td><td>Output: Operation underway</td><td>Closes during operation. Opens when an operation ends.</td></tr>
<tr><td>CSTP</td><td>24</td><td>–</td><td>Input: Cycle stop</td><td>Stops a cycle when it is on during a sequential programmed operation.
★ Execute a programmed step operation if the command for a programmed operation is inputted while this signal is on.</td></tr>
<tr><td>HOLD</td><td>5</td><td>–</td><td>Input: Hold</td><td>It decelerates then stops a motion when it is on during programmed operation.
★ If HOLD signal remains on, start command of a programmed operation is not valid.</td></tr>
<tr><td>STOP</td><td>26</td><td>–</td><td>Input: Forced stop</td><td>It decelerates then stops a motion when it is on during programmed operation.
★ If STOP input remains on, start command of a programmed operation is invalid.</td></tr>
<tr><td>CSTPA</td><td>32</td><td>–</td><td>Output: Indicates cycle stop is on.</td><td>It is closed during the state of cycle stop by CSTP input.</td></tr>
<tr><td>HOLDA</td><td>13</td><td>–</td><td>Output: Indication of hold</td><td>It is closed during the state of hold by HOLD input.</td></tr>
</table>

★ Refer to "17.4.5. Stopping Programmed Operation" for details.

◆ Procedures of programmed operation

1) Turn "servo on" referring to "17.4.2. Servo on."

2) Execute Home return referring to "17.4.3. Home Return."

3) Make sure that none of the inputs for termination of operation, such as CSTP, STOP and HLD, is on and MTN output is opened. You cannot start a programmed operation while STOP or HLD inputs is on.
If MTN input is closed (= Programmed operation is under way.), you cannot start a new programmed operation.

→ Wait till the programmed operation ends or forcibly terminate the programmed operation by inputting STOP signal. (Deactivate STOP signal after completion of the programmed operation.)

4) Specify a program number by inputs of program number selection.

5) Turn RUN input on. A programmed operation starts and MTN output is closed.

6) MTN output is opened when a programmed operation completes.

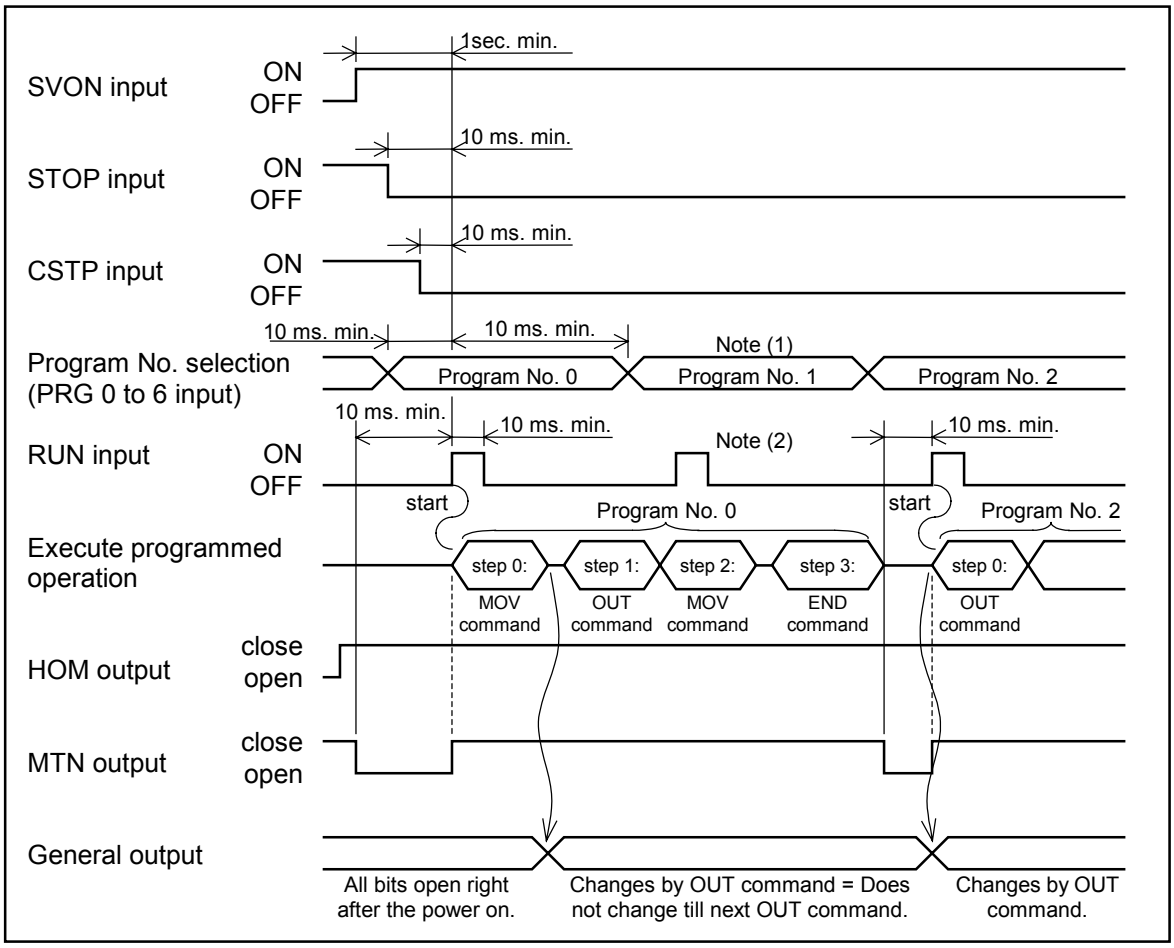
Note: (1) Once a programmed operation is started (= MTN output remains closing), the program selection input is not valid. If you want to change the program number, wait till the programmed operation completes or forcibly stop the programmed operation by inputting STOP signal.

(2) RUN input is detected at its rising edge signal (effective at the point of off to on). If RUN input is being on, you cannot start a programmed operation.

◆ Timing of control input / output of programmed operation

- Observe signal timing as shown in Figure 17-8 below.

Figure 17-8



Note: (1) Once a programmed operation starts (= MTN output is closed), program number selection is invalid until the end of the operation. It can be used as general input.

(2) Once a programmed operation starts (= MTN output is closed), RUN input is invalid until the end of the operation.

17.4.5. Stopping Programmed Operation

- The system provides following functions to terminate programmed operation in the middle of operation in external control mode.
  1) Emergency stop
  2) Forced stop
  3) Cycle stop in the middle of sequential operation
  4) Hold

◆ Emergency stop procedure : EMST input off

- Terminate a programmed operation immediately. The motor is in the "servo lock" state.

  * For function of the emergency stop, see "11.3.1. Emergency Stop."

◆ Forced stop procedure : STOP input on

- Cease a programmed operation instantly (when the robot is in motion, the motor decelerates and stops) and the system returns to the top of the program and waits.
- The motor is in "servo lock" state.
- After STOP input is off, RUN input resumes the programmed operation from the top of the program.
- If STOP input remains on, RUN input is invalid.

Note: The execution of forced stop resets data register and internal flag data excluding the following data.

◊ Data of point register : (P0000 to P3999)

◊ General output data by OUT command

◆ Cycle stop in the middle of sequential operation : Procedure : CSTP input on in the middle of sequential operation

- The system stops after completing the step being executed.
- CSTPA output is closed during the cycle stop.
- The motor is in servo lock state.
- RUN input after deactivation of CSTP input resumes the sequential operation from the next step of the program.
- RUN input, keeping CSTP input on, starts programmed step operation from the next step.

Note: (1) During cycle stop, detection of an alarm, inputting STOP input or deactivation of EMST input will cancel the cycle stop state. (= MTN output is opened because the program is ceased.) This disables to resume the programmed operation. If you try to resume the programmed operation in this state, it starts from the top of the program. (after releasing emergency stop when the system is in the emergency stop state.)

(2) Cycle stop is not accepted during a continue path motion. It is valid after the continue path has completed.

◆ Procedure for hold : HOLD input on

- This is to cancel a motion command. It decelerates and stops the motion. HOLDA output is closed. The system stands by at the position where it stopped.
- The motor is in servo lock state.
- In the hold state (when both outputs, HOLDA and MTN, are closed), input of RUN after deactivate HOLD input resumes the remaining motion from the stopped position, then continues the programmed operation.
- RUN input is invalid while HOLD input is on.
- If the cycle stop is on in the hold state, it resumes remaining motion of the step then stops the operation.

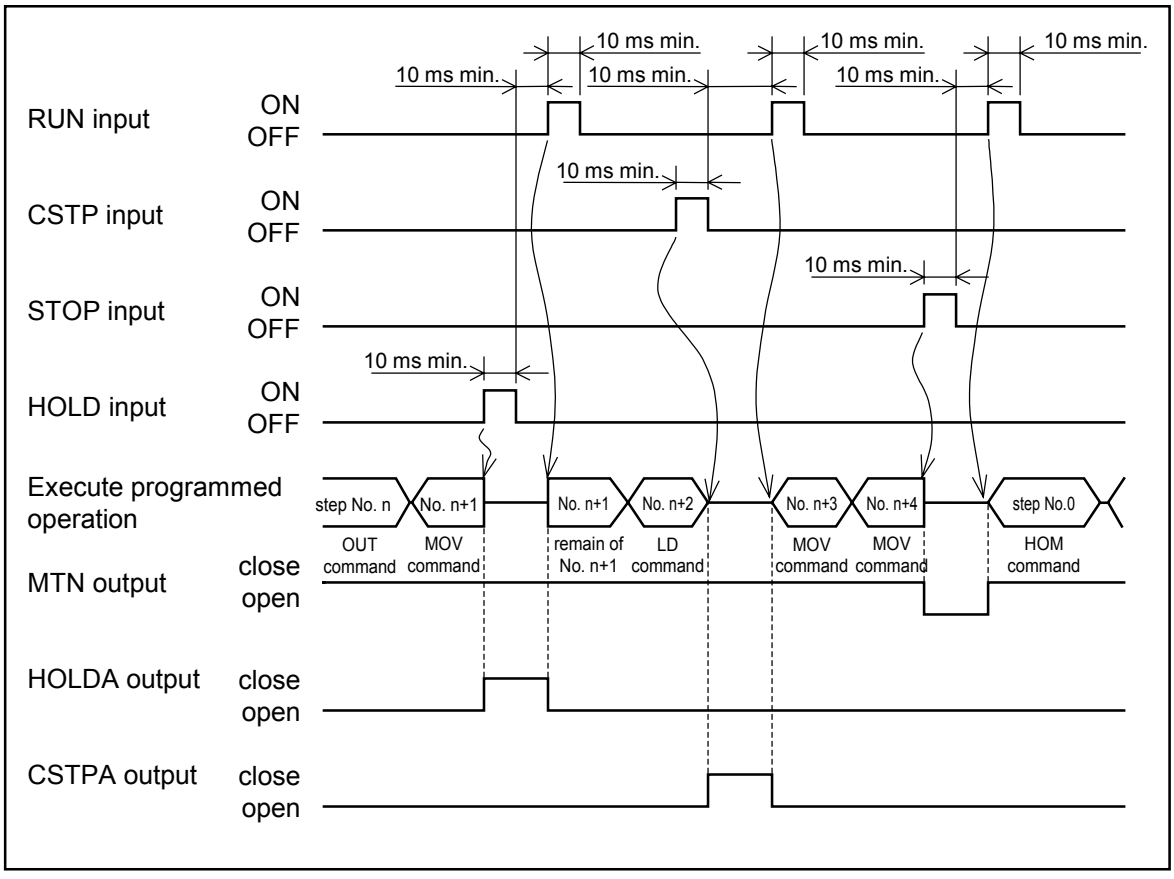
Note: (1) During hold state, a detection of an alarm, activating STOP input or deactivating EMST input will cancel the state (= MTN output is opened as the program is ceased.). This disables to resume the programmed operation. If you try to resume the programmed operation in this state, it starts from the top of the program. (after releasing emergency stop when the system is in the emergency stop state.)

(2) Input of HOLD command in the middle of a continuous path motion terminates the motion. The remain of the motion will be treated as an ordinal motion command, not as an continuous path motion.

(3) In the state of hold which is activated in the middle of execution of move command, deactivation of SVON input makes 'servo off' state, then an alarm will be given. (program error)

◆ Signal timing to terminate sequential programmed operation

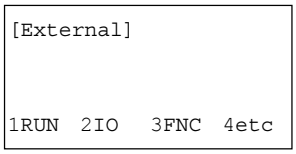
Figure 17-9



17.4.6. Operating Teaching Box

- Several operation status and data can be monitored using the teaching box in external control mode.
  1) Monitor of operation status
  2) Input / output monitor
  3) Monitor of software version and history of alarm

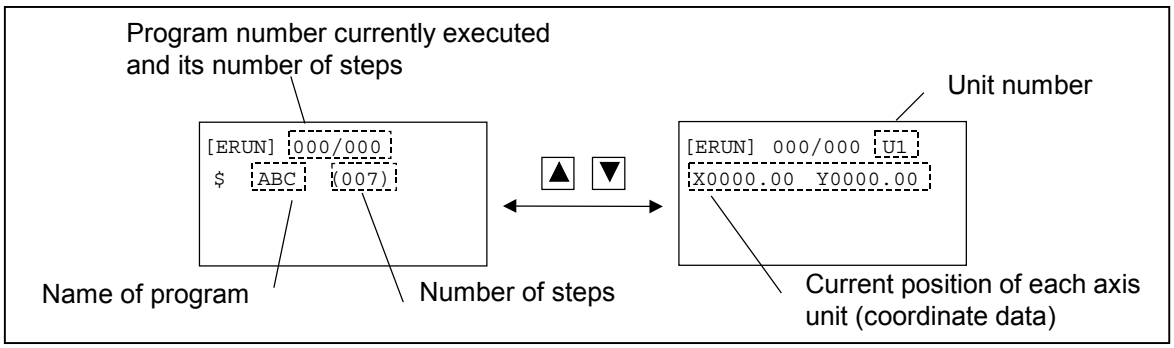
Figure 17-10: Screen of external control mode [External]



17.4.6.1. Monitor of Operation Status

- A screen to select monitor function appears when F1 key is pressed in "External" screen. Use ▲ ▼ keys to switch monitors of program name, number of program steps and current position. Then, ◀ ▶ keys enable to select the unit number when the screen shows current position in case of multi-axis combination.

Figure 17-11: Operation monitor screen

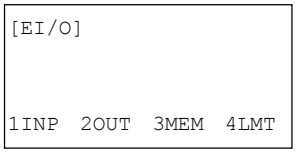


17.4.6.2. I / O Monitor

- A screen to monitor the status of input / output signal appears as shown below when F2 key (IO) is pressed in the [External] screen. Use F1 ~ F4 keys to select a monitor screen.

- Items that can be monitored.
  INP : Input port
  OUT : Output port
  MEM : Imaginary I/O port
  LMT : Travel limit port

Figure 17-12: I/O monitor selection screen

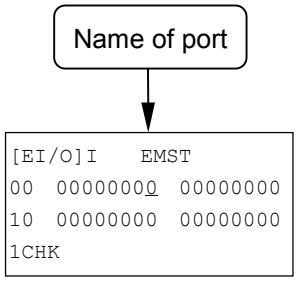


1 Monitoring input port

1) Monitor input port.

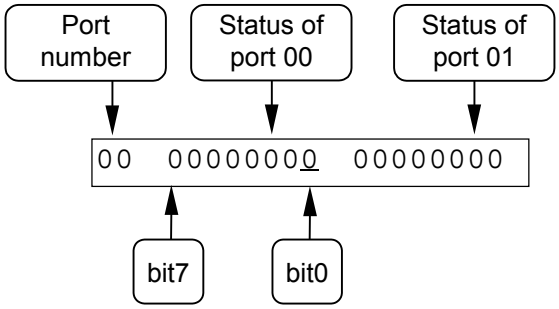
- Monitor screen for the input port appears when F1 (INP) key is pressed on the selecting screen of I / O monitoring.
- I is indicated on the side of [EI/O] in this screen.
- Current state of four input ports is reported on 2nd and 3rd lines. Use ◀ ▶ and ▲ ▼ keys to move the cursor.
- Name of input port, of which bit is indicated by the cursor, is displayed on 1st line.
- If more than four input ports were to be monitored, use ▲ and ▼ keys to scroll the screen.

Figure 17-13: Monitor screen of input port



- Indication of the 2nd and 3rd lines are, from left to right, port number, status of the port and status of port of the port number +1.
- Port number is indicated by unit of 8 bit, as left side is bit 7 and right side is bit 0.
- Status of the port is indicated by 0 as off and 1 as on regardless the port logic (normally open or close).

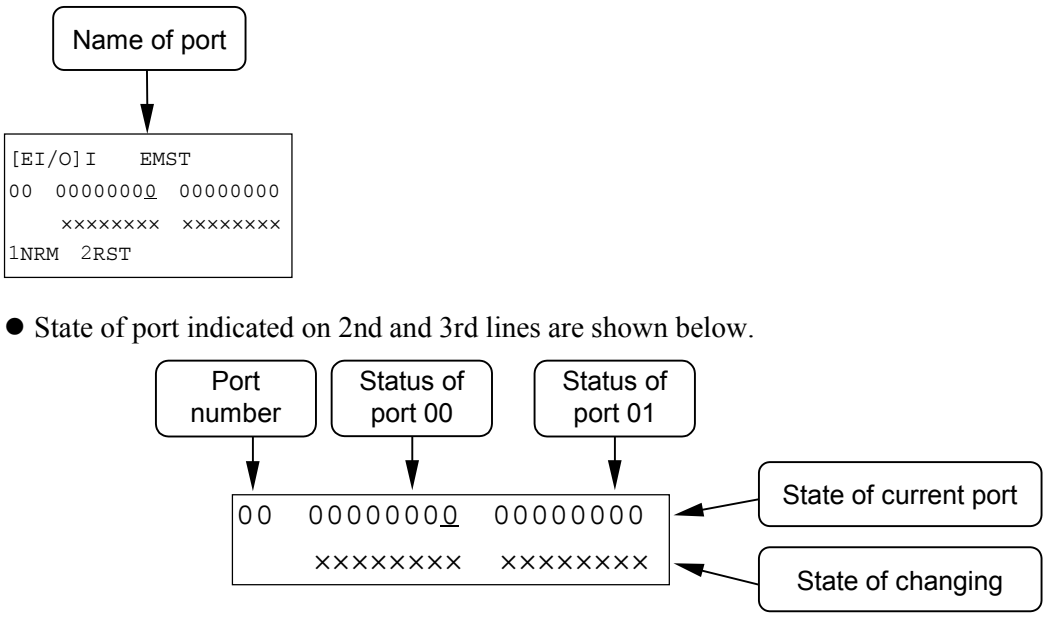
Figure 17-14: Monitor example of input port



2) Monitor status of input port.

- Setting screen of the input port appears when F1 (CHK) key is pressed in the monitor screen of input port status.
- The screen displays two kinds of state (current and changing state) on the 2nd and the 3rd lines. ◀ and ▶ keys move the cursor.
- Port name of which bit is indicated by the cursor is displayed on the 1st line.
- If more than two ports were to be monitored, use ▲ and ▼ keys to scroll the indications.

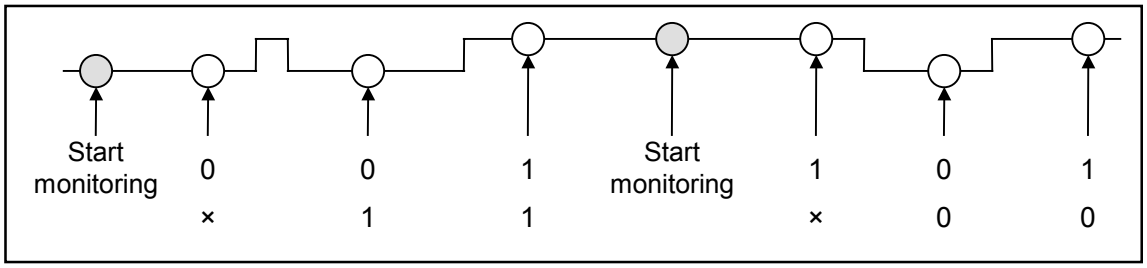
Figure 17-15: Screen to monitor input port state



- State of port indicated on 2nd and 3rd lines are shown below.

- Changing state of port on the 3rd line shows the changes on a port after RDY output is on or F2 (RST) key is pressed.
- Each bit corresponds one to one to the bit on the 2nd line.
- When status of a port changes 0 to 1 after start of monitoring, x changes to 1, while a port status changes 1 to 0 after start of monitoring, x changes to 0. A bit of which status is changed keeps its indication until F2 (RST) key is pressed.
  ("Current port status" on 2nd line changes in real time basis.)

Figure 17-16: Example of changes of a port.

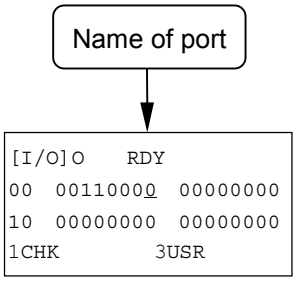


- The screen returns for monitoring input port by pressing F1 key (NRM) in the monitor screen of input port state.

2 Monitoring output port

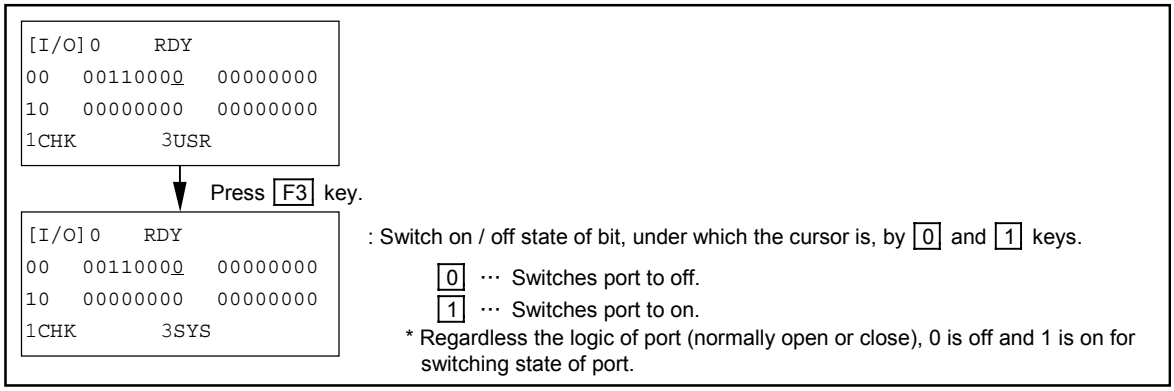
- Monitor screen for output port appears when F2 (OUT) key is pressed on the selecting screen of I / O monitoring.
- O is indicated on the side of [EI/O] in this screen.
- Current state of four output ports is reported on the 2nd and 3rd lines. Use ◀ ▶ and ▲ ▼ keys to move the cursor.
- Name of output port, of which bit is indicated by the cursor, is displayed on the 1st line.
- If more than four output ports were to be monitored, use ▲ and ▼ keys to scroll the screen.

Figure 17-17: Monitor screen of output port



- Functions and structure of indication are the same as that of input port. However, pressing F3 key (USR) makes possible to switch temporarily on / off state of a bit, under which the cursor is, by 0 and 1 keys.

Figure 17-18



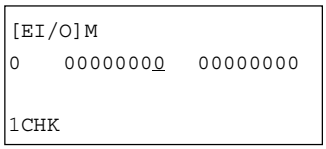
- Setting is effective till pressing F3 key (SYS) again or MODE key. When the setting is canceled, it returns to the former state.

3 Monitoring imaginary I/O port

1) Monitor imaginary input/output port.

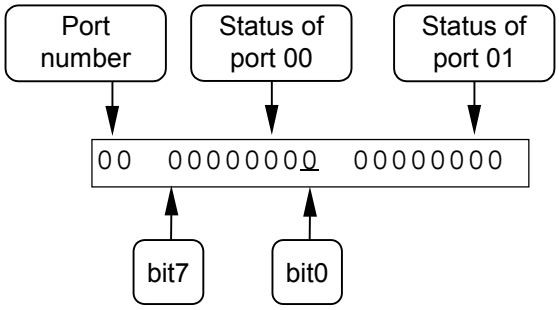
- Monitor screen for imaginary input and output port appears when F3 key (MEM) is pressed in the selecting screen of I/O monitor.
- M is indicated on the side of [EI/O] in the screen.
- Current state of two imaginary I/O port is displayed on the 2nd and 3rd lines. Use ◀ and ▶ keys to move the cursor.

Figure 17-19: Monitor screen of imaginary I/O port



- Indications of 2nd line are, from left to right, port number, status of the port and status of port of the port number +1.
- Status of the port is indicated by unit of 8 bit, as left side is 7 and right side is bit 0.
- Status of the port is indicated by 0 as off and 1 as on regardless the port logic (normally open or close).

Figure 17-20: Monitor examples of imaginary I/O port



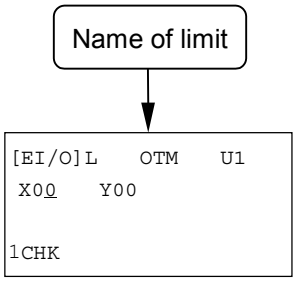
2) Monitor of imaginary I/O port status.

- Monitor screen of imaginary I/O port status appears when F1 key is pressed in the monitor screen of imaginary I/O port.
- The screen displays two kinds of state, current and changing status, on the 2nd and the 3rd lines. Use ◀ and ▶ keys to move the cursor.
- Refer to "2) Monitor status of input port. in 1 Monitoring input port" for the way to monitor the status of the port.

4 Monitoring hardware travel limit

- This is to report the state of hardware travel limit (F3 alarm). Refer to "13.4.18. Travel Limit (Mechanical lock)."
- The monitor screen appears when F4 key (LMT) is pressed in the I/O monitor screen.
- L is indicated on the side of [EI/O] in the screen.
- The state of hardware travel limit of specified motion unit is indicated on the 2nd line of the screen. Use ◀ and ▶ keys to move the cursor. The name of the travel limit, of which bit is indicated by the cursor, is displayed on the 1st line.
- Use ▲ and ▼ keys to move the cursor to scroll the screen if the motion units are more than two for a multi-axis combination.
- The functions and contents of indications are the same as the input port monitoring.

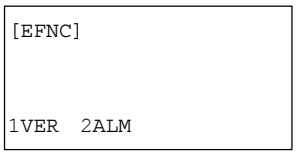
Figure 17-21: Indication of hardware travel limit state (Two axes combination)



17.4.6.3. Monitor of Software Version and Alarm History

- Monitor screen of software version and alarm history appears as shown below when F3 (FNC) key is pressed in [External] screen. Select a required monitor using F1 and F2 keys in the screen of function selection.
- It can monitor software version (VER) and alarm history (ALM).

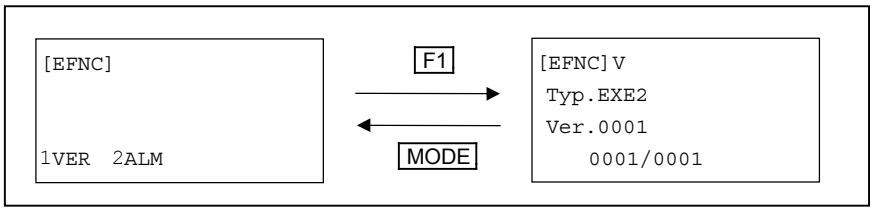
Figure 17-22: Selecting function screen



1 Monitor of software version

- Screen shows the software version by pressing F1 (VER) key in the selecting function screen.

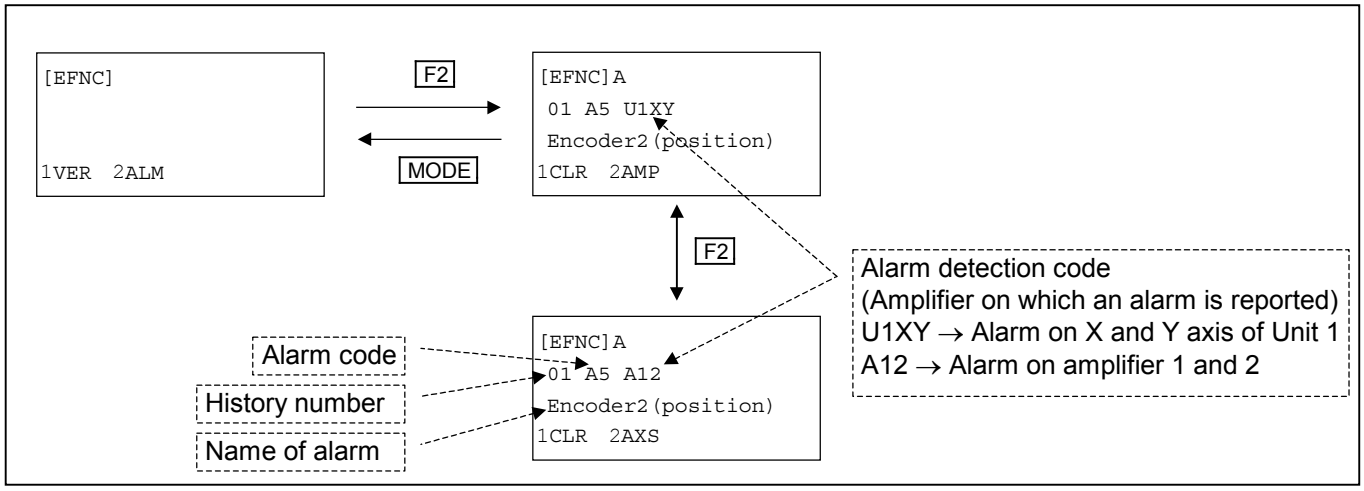
Figure 17-23: Display of software version screen (Two axes combination)



2 Monitor of alarm history

- Alarm history is displayed by pressing F2 (ALM) key in the selecting function screen.
- Information on alarm histories are:
  On the 2nd line: history number, alarm code and alarm detection code
  On the 3rd line : name of alarm.
  Newer the alarm, smaller the number. Refer to "Table 13-49 Contents of alarm history" in "13.7.2. Indication of Alarm History."
- Alarm history is displayed by pressing ▼ key.
- Use F2 key to switch indication of an motion axis or a amplifier number in the alarm detection code on which an alarm is reported.

Figure 17-24: Display of alarm history screen



- F1 (CLR) key in the monitor screen clears alarm history.

17.5. Operational Function

17.5.1. Home Return

17.5.1.1. Function of Home Return

- Table 17-5 shows functions of Home return.

Table 17-5: List of Home return function

Item	Refer to:	Description
Direction to search mechanical stopper position	9.3.2. Parameters for Home Return Operation	• Specifies moving direction to search position of mechanical stoppers. (motor or opposite side to the motor) The shipping set is motor side.
Home position offset	9.3.2. Parameters for Home Return Operation	• If "Home position offset" has been set, the Home return operation completes after moving "off-set" distance from the defined Home position. • The factory set is 0 (zero).
Stan-by position	9.3.2. Parameters for Home Return Operation	• If stand-by position is set, Home return completes after moving to the set position from defined Home position.
Coordinate direction	9.4.1.1. Parameter for Servo	• Specify the + (plus) and - (minus) direction of the coordinate. • Factory set is + (plus) for opposite to motor side and - (minus) for motor side.
Order to search position of mechanical stopper *	9.3.2. Parameters for Home Return Operation	• You may set Home return in two ways, to perform it one by one axis, or all axes simultaneously. • You may set precedence of axes for "one by one" Home return. Shipping set is to perform "all axes simultaneously."
Speed Acceleration/ deceleration Mechanical stopper searching speed	9.3.2. Parameters for Home Return Operation	• Specify the speed, acceleration and deceleration of Home return. • Factory set Speed : 20 mm / sec Acceleration / Deceleration : 0.5 m /s^2 Mechanical stopper search speed : 1 mm / sec

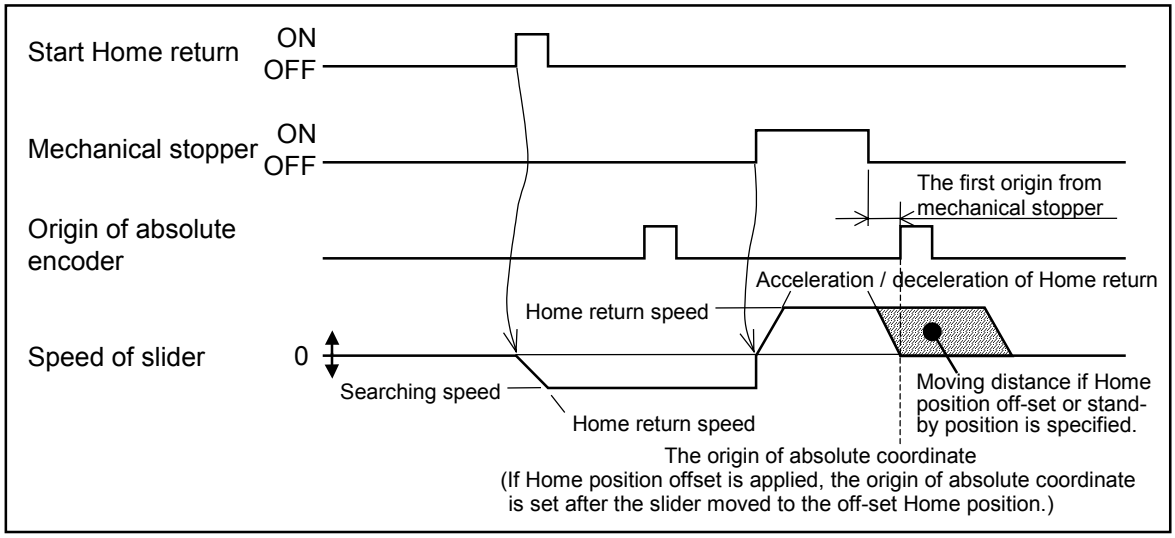
* Not applicable for a sigle axis system.

17.5.1.2. Outline of Home Return

- The slider of a module main unit starts to the motor side* for Home return and reverses its motion when it hits the mechanical stopper. Then it stops at the first origin position of absolute encoder and sets Home position.

  * It moves to opposite side of motor when the setting is made to reversed direction or the motor mounting parameter (direct or indirect mount) is not compatible to the module main unit.

Figure 17-25: Home return



17.5.1.3. Signal Timing of Home Return (Multi-axis combination only)

- Motion of individual axis unit is described in "17.5.1.2. Outline of Home Return." The following describes Home return in a multi-axis combination.

◆ Example: Home return sequence in Z axis unit → Y axis unit → X axis unit (Three axes combination)

1) Z axis unit starts Home return and stops when it hits the mechanical stopper.

2) Y axis unit starts Home return and stops when it hits the mechanical stopper.

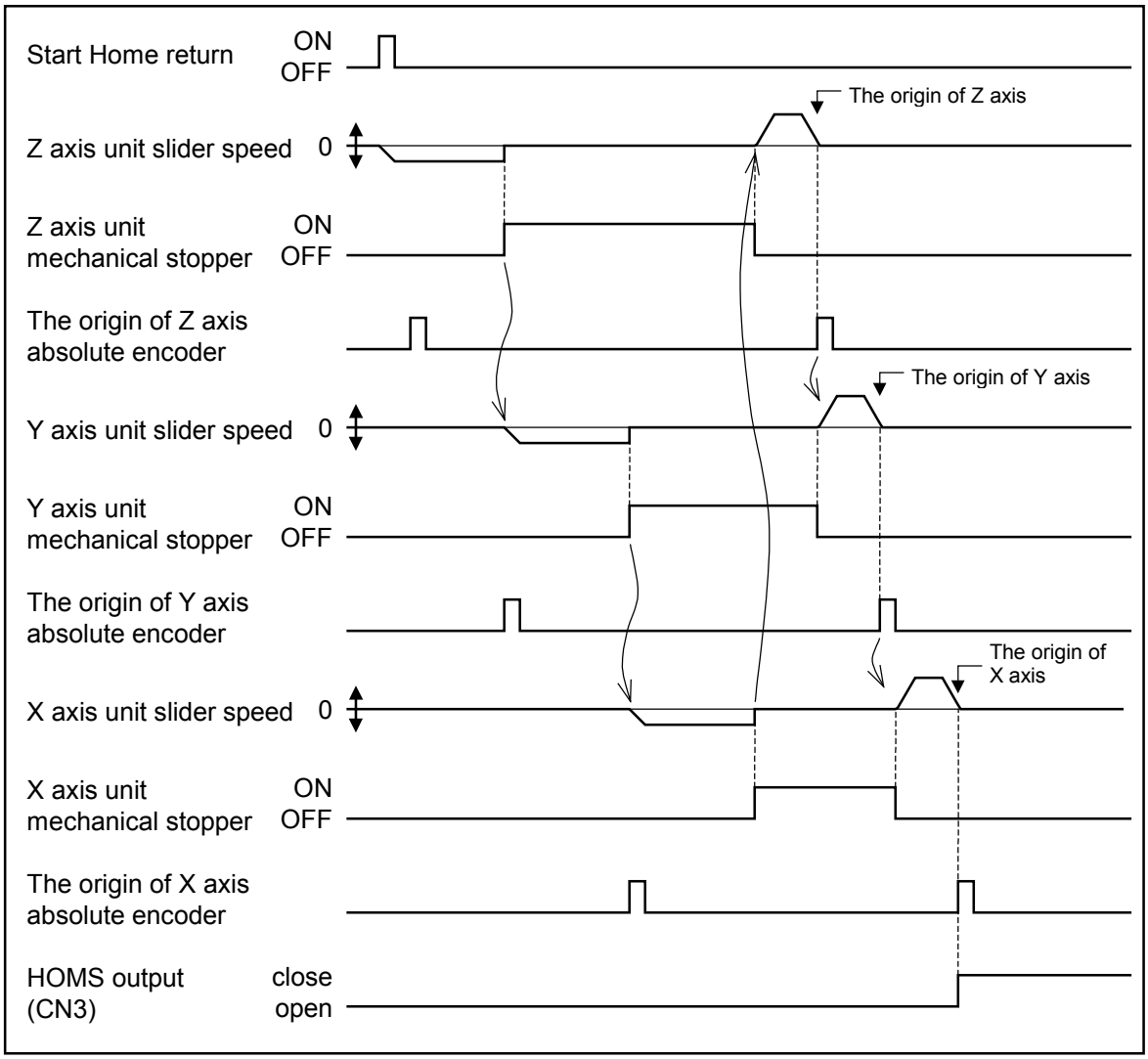
3) X axis unit starts Home return and stops when it hits the mechanical stopper. All axis units stop at the position of its mechanical stopper. Then Z axis unit starts to reverse its motion to the origin of Z axis absolute encoder.

4) Z axis unit stops at the origin of absolute encoder and defines there as the Home position.

5) Then Y axis unit does the same and X axis unit follows.

6) The Home return operation is completed when the Home positions of all axis units are specified and HOMS output (CN3) is closed.

Figure 17-26: Example: Sequential Home return in order of Z axis unit → Y axis unit → X axis unit



17.5.2. Resume Programmed Operation

- You can resume a programmed operation only when it is terminated by a cycle stop, even the power was once turned off and recovered.
- It requires to change internal parameter setting as the resumption of a programmed operation, which is terminated by power shutoff, is not set to the factory set.

Caution : When resuming programmed operation after stopping due to power shutoff is set, do not turn off the power during storing the data of operation state to flash memory. Be sure to check that DATWT output opens before turn off the power as it is closed during storing the data. Otherwise, a memory error alarm arises.

1 Parameter setting to resume a programmed operation

- Table 17-6 shows internal parameter setting for resuming programmed operation once interrupted by power shutoff.

Table 17-6

Parameter	Factory set	Set to resume programmed operation	Chapter to be referred
CSTP data save	NOP	SAVE	9.7. Parameters for PMD setting

2 Input / Output for resuming programmed operation

Table 17-7

Signal	Pin No.	Function	
SVON	8	Input: Servo on	ON: Servo is on. OFF: Servo is off.
RUN	7	Input: Start programmed operation	Rising signal from off to on starts direct operation.
RSTA	6	Input: Resume programmed operation	Programmed operation may be resumed when this input is set to on at the moment of RUN signal input.
CSTP	24	Input: Cycle stop	Turn on the input during programmed operation to make cycle stop.
DATWT	12	Output: Saving data	It is closed when saving program and data of parameter to internal flash memory. If the power is turned off during this moment, you lose the data and memory error alarm is given. Programmed operation and Home return cannot be performed during this moment.
RSTAE	31	Output: Ready to resume	This output closes when a programmed operation that has been interrupted is ready to resume.

17.5.2.1. Procedure to Resume Programmed Operation in External Control Mode

1) Stop programmed operation by turning CSTP input on. Turn off SVON input and turn off the power after confirming that output of DATWT is open.

2) Recover the power then put on SVON signal. Confirm that RATAE output is closed. Turn RATAE signal on, then restart the program by RUN signal.

Note: As RUN signal is detected in the rising current from off to on, the program cannot start if RUN has been on from the beginning.

- If RUN signal is on without turning on RSTA signal, the program starts its operation from the top, same as a normal operation. In this case, data registers D000 to D099 will be initialized, while data registers D100 to D199 will retain their state. If RUN signal is on after turning on RSTA signal, all data registers D000 to D199 will retain their state before the power was off.

- If RSTA command* is set to the top of an interrupted program, its programmed operation resumes after initializing program for recovery,* which is specified by RATA command,* is executed.

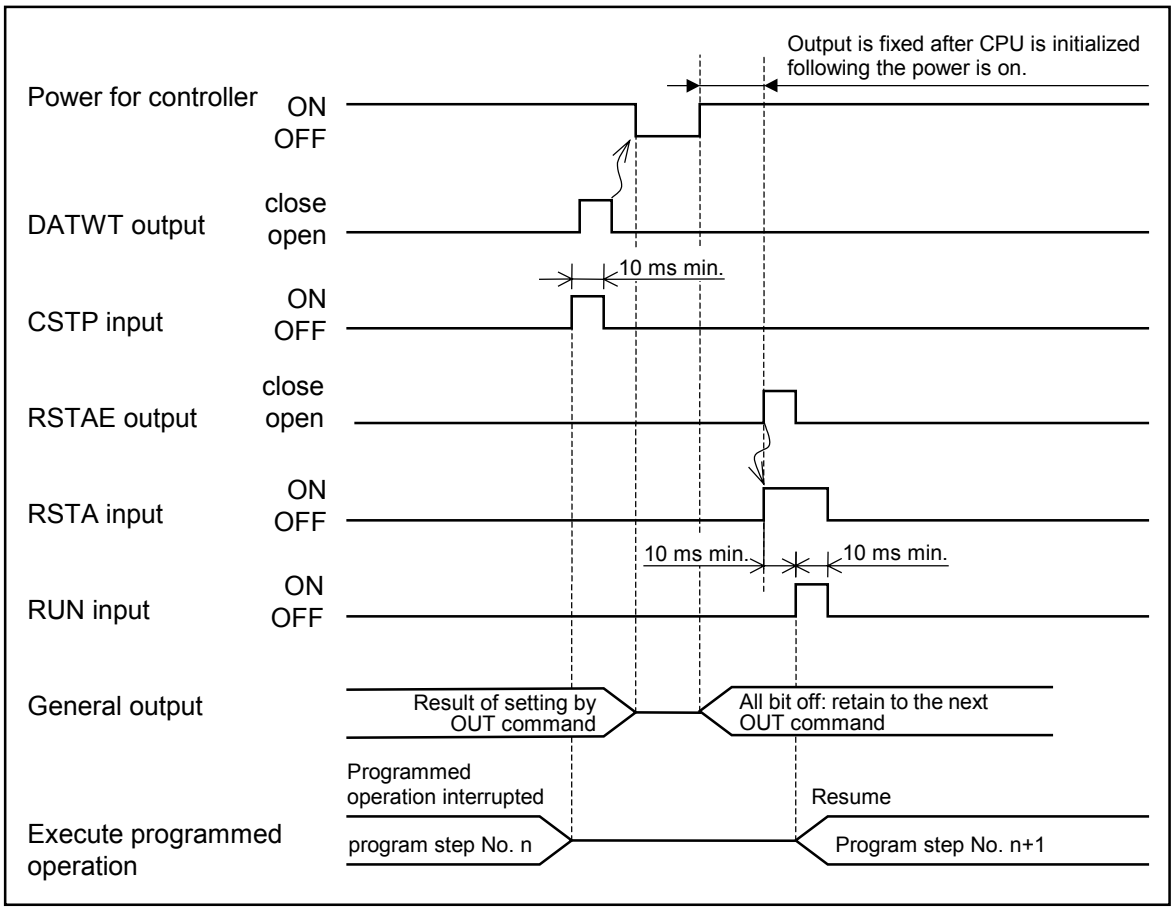
- If the position of starting operation and the state of general input are required to go back to the condition when the program is interrupted, it can be done by an initializing program for recovery.*

* Refer to "15.2.7. Description of Program Command" for RSTA command and initializing program for recovery.

1 Timing of control Input / Output of resuming programmed operation ----- 1

RSTA command (initializing program for recovery) is not set.

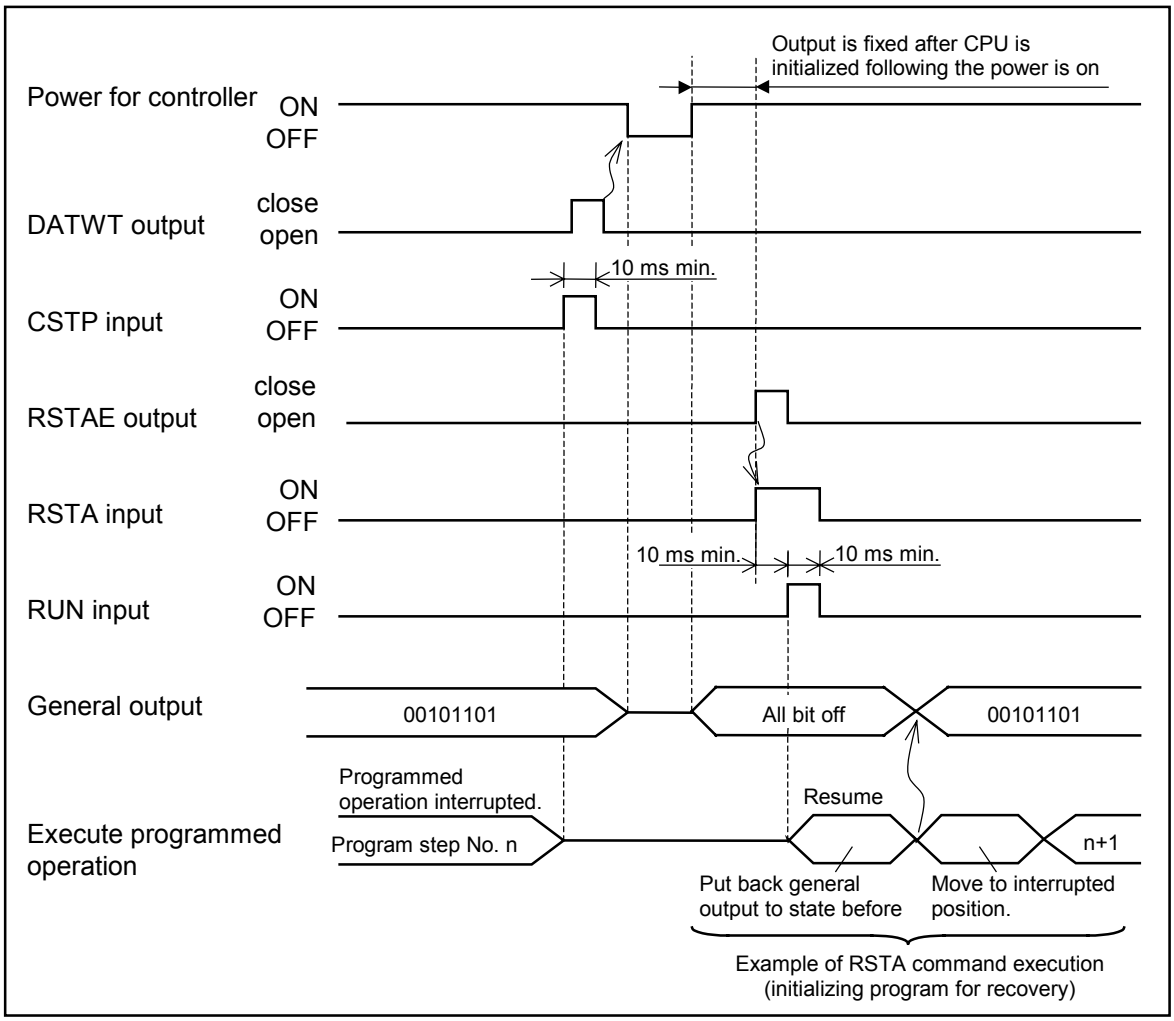
Figure 17-27



2 Timing of control Input / Output of resuming programmed operation ------- 2

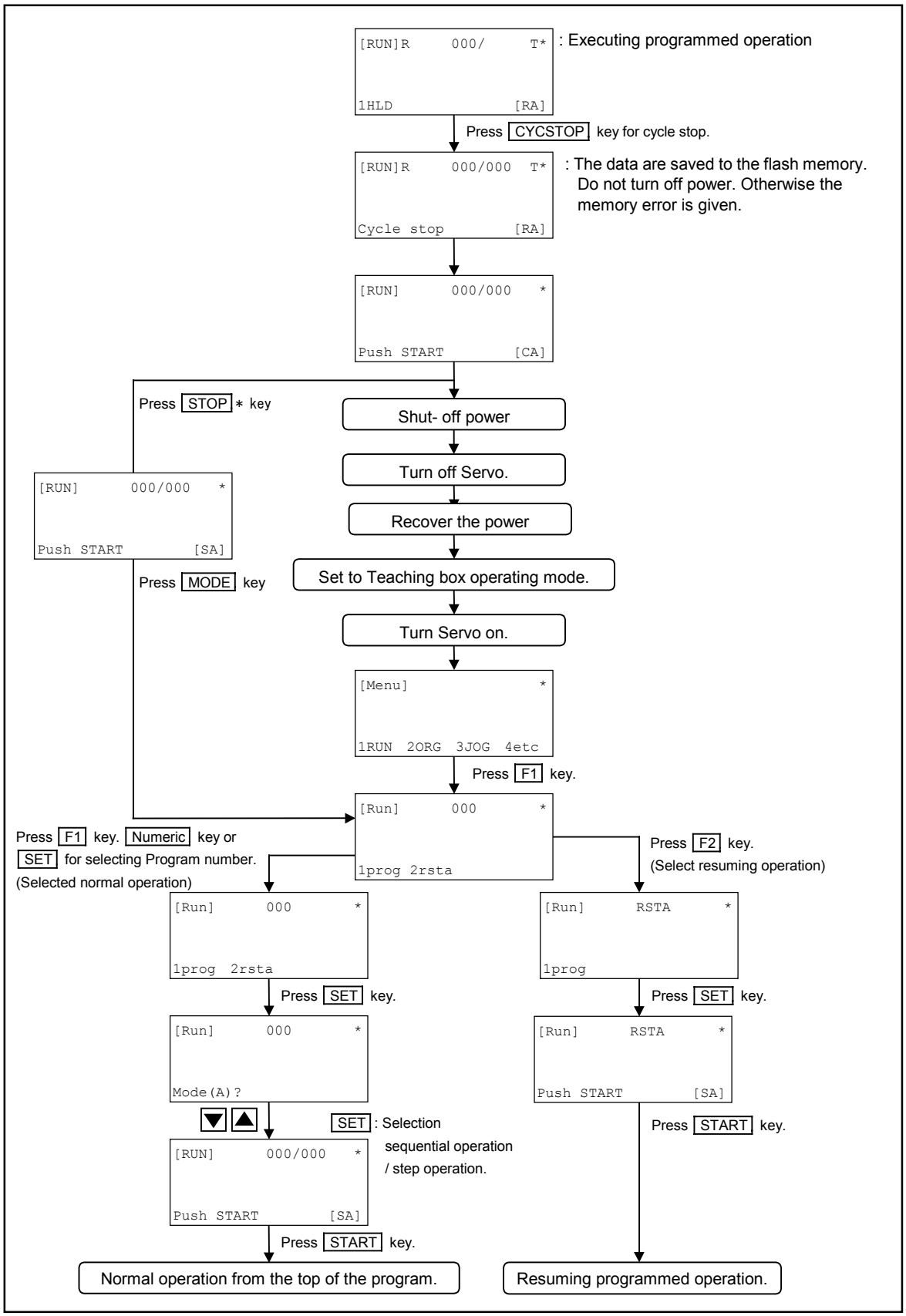
RSTA command (initializing program for recovery) is set.

Figure 17-28



17.5.2.2. Procedure to Resume Programmed Operation in Teaching Box Control Mode

Figure 17-29



* It is possible to resume a programmed operation when the program is interrupted by MODE key without turning off the power after the end of cycle stop.

17.5.3. Output of In-position

- FIN signal may be set to P1-EXT. IO connector. (Refer to "9.9.1.Parameter List.")

  Note: The factory set does not set FIN signal to PI-EXT-IO connector. Also FIN signal cannot be outputted as the parameter to set signal for completion of positioning (Fin control) are set to OFF.
  Turn Fin control ON in "9.3.1. Parameters for Programmed Operation" and set FIN signal referring to "9.9. Parameters for Output Signal Format."

- As an in-position signal format, you may select either FIN or COIN format.

Table 17-8

Format	Function	Way of selection	Factory set
FIN format	One-shot output signal which closes for a specified time after end of positioning.	Specify Fin out time* numerically. (Set other than OFF.)	✓ (0.1)
COIN format	Output format that opens during motion and closes after end of positioning.	Set Fin out time* to OFF.	–

* Refer to "9.3.1. Parameters for Programmed Operation."

Note: (1) FIN output does not change by a command other than a motion command. (such as Jump and General output command, etc.)

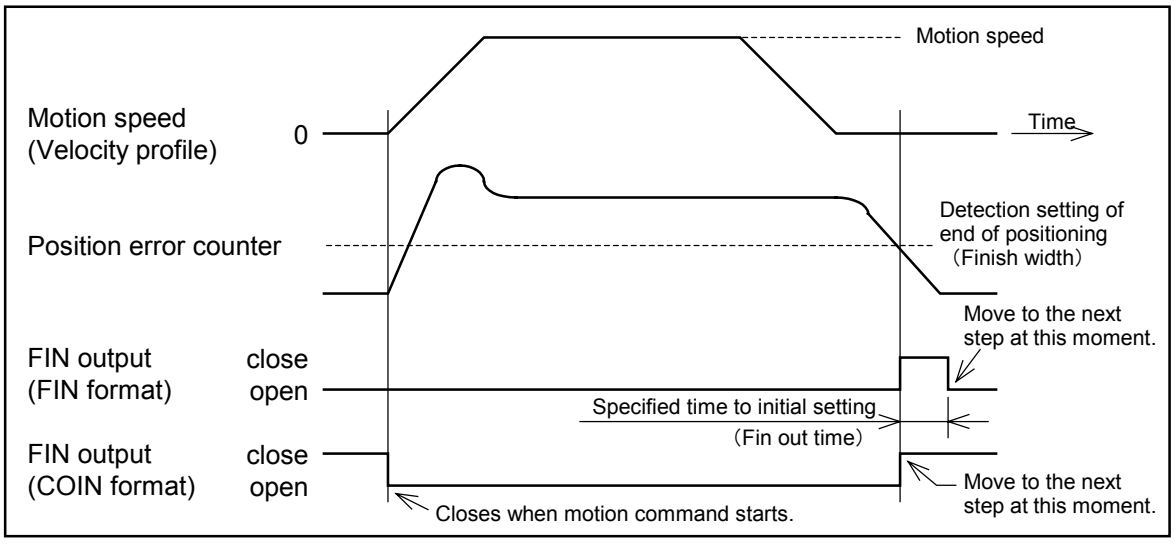
FIN format ---------- Remains open
COIN format---------- Remains close

(2) End of positioning output is on after completion of all motion commands in a continuous path operation. FIN output does not change in every end of motion command in continuous path operation. (Multi-axis combination)

- FIN output closes for a specified time when the motion command is completed and the error of position error counter reaches less than a specified value. Proceed to the next step after end of FIN output.

- Specify detecting width of end of positioning (Finish width) and a closing time of FIN output in initial setting referring to "9.3.1. Parameters for Programmed Operation."

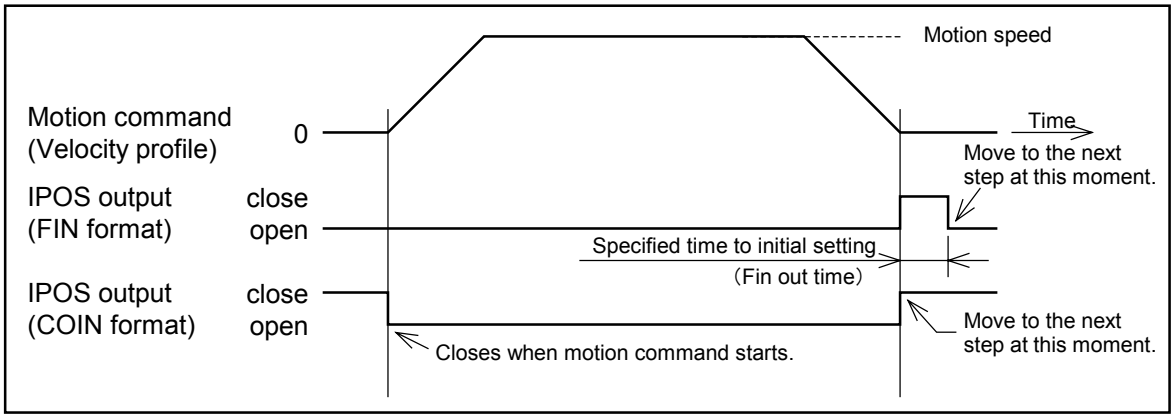
Figure 17-30



◆ In-position output in a special setting

- FIN output follows the timing of motion command regardless residual error of position error counter when an In-position signal detection (Finish width) is set to OFF.

Figure 17-31



17.5.4. Pulse Train Input (Single axis system only)

17.5.4.1. Function of Pulse Train Input

- The function of pulse train input is described in Table 17-9 below.

Table 17-9: Function list of pulse train input

Item	Section to be referred for setting	Description
Resolution of pulse train input	"9.3.5. Parameters for Pulse Train Input"	• You may set the motion resolution for one pulse in pulse train input operation. Factory set : 0.010 mm / pulse

17.5.4.2. Description of Operation

- The operation of pulse train input can be available when the following conditions are met.
  1) In the external operation mode TBXM output = open
  2) Controller is ready. RDY output = close
  3) Not in motion MTN output = open
  4) No in remote control mode EREM input = off
  5) Servo is on. SVON = on

- The moving distance for a pulse may be selected from three styles below by setting the pulse train input resolution (Pulse resolution).
  1) 0.01 ~ 0.10 mm
  2) $\left(\frac{\text{Ball screw lead}}{\text{Encoder resolution}} \times \text{speed reduction ratio} \right)$ mm
  3) Pulse train invalid (Disregard input pulse.)

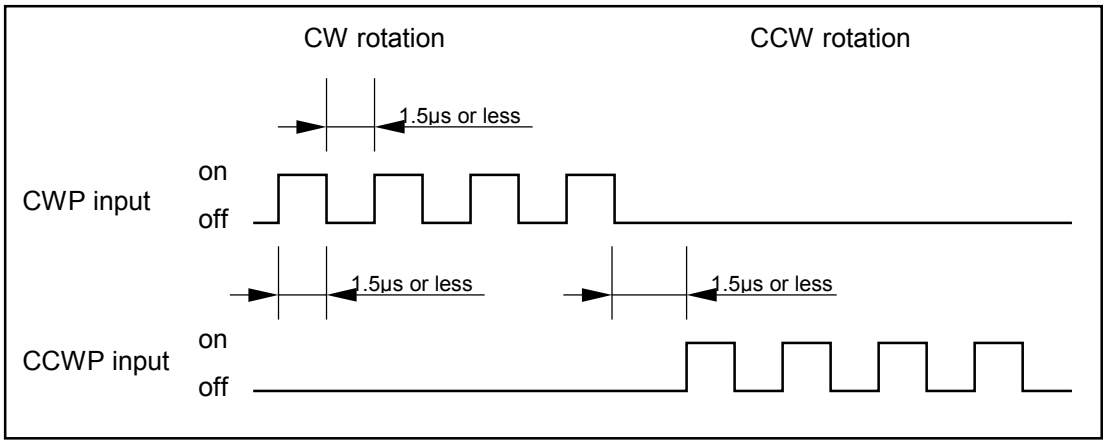
- Programmed operation and Home return are possible in the pulse train input operation. However, do not use the start commands (RUN and HOS) in the pulse train input operation.

- When turning on the servo of a main unit which is incorporated motor brake, there may be approximately 100 ms time lag to release a magnetic brake of the motor. Do not input the pulses at that moment.

17.5.4.3. Input Timing

Caution : The timing specified in Figure 17-32 is to receive the pule train input. There is another restriction such as the maximum speed. Adjust a frequency of pulse train for the allowable speed of a main unit.

Figure 17-32



(Blank Page)

18. Remote Control Operation

- You can control EXEA controller directly by commands from personal computer via RS-232C interface. This operation mode is called "Remote Control."

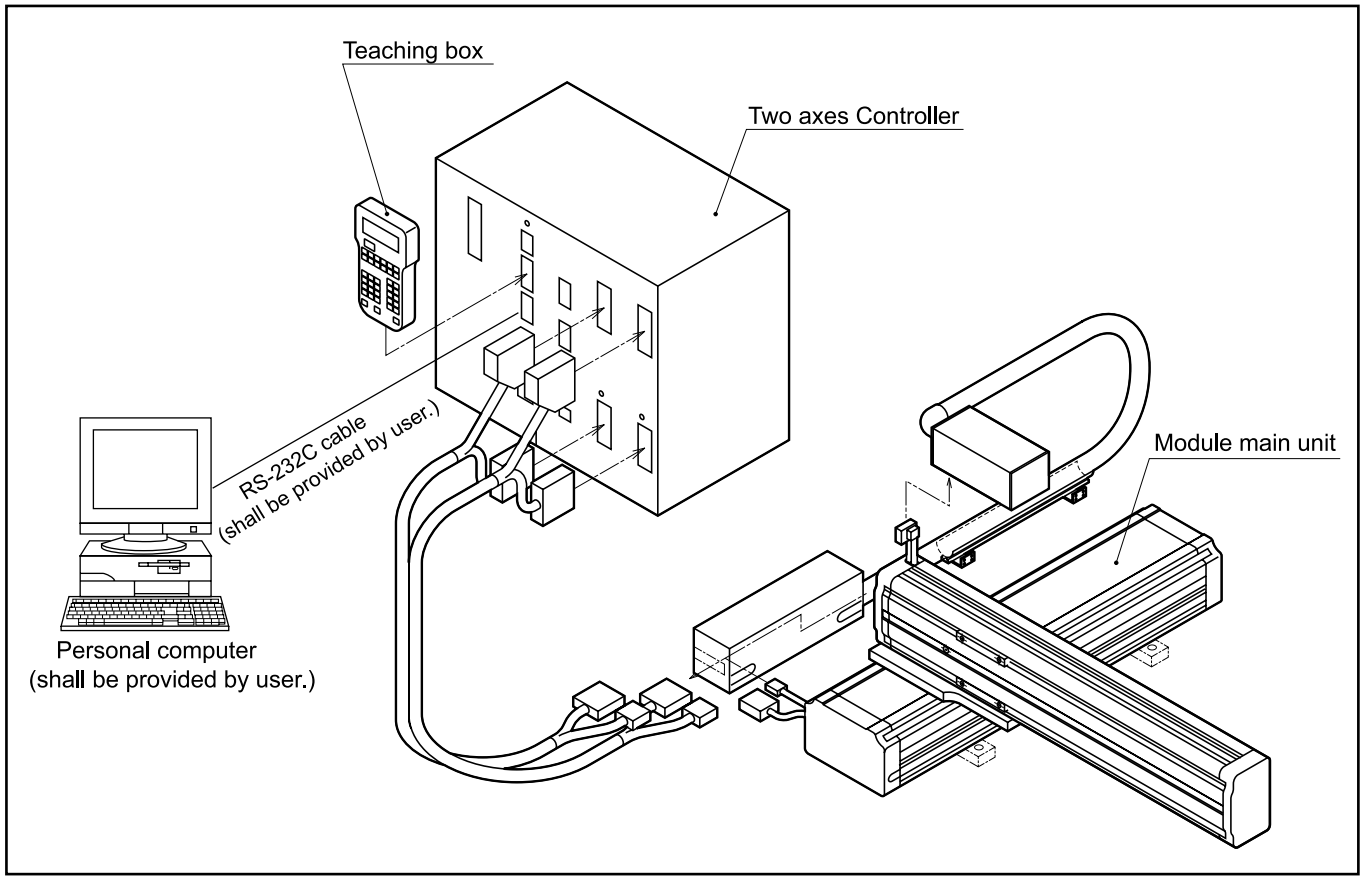
- Inputting the commands, which are provided for the remote control, to EXEA controller enables the following operations.

  1) Sequentially execute move commands (motion command) and output commands from the personal computer. In this case, the operation programs are stored in the personal computer and EXEA controller performs simply the positioning operation and outputs information to the general output port.

  2) Execute the operation programs stored in EXEA controller, that are programmed referring to "15. Programming," by start and execution commands from the personal computer.

  3) The EXEA controller reports its state (alarm conditions, current position etc.) to the personal computer.

Figure 18-1



- Use the CN2 connector dedicated to RS-232C interface.

* The user shall provide an adapter to your own personal computer for RS-232C cable in accordance with its specification.

18.1. Interface Specification

Table 18-1

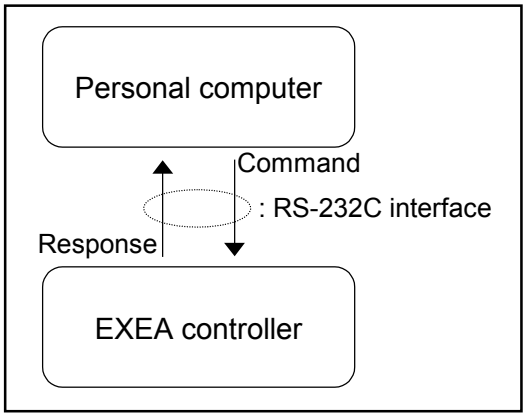
Transmission	Full duplex
Communication speed	9600 bps
Data length	8 bit
Stop bit	2 bit
Parity check	None
Communication control	X parameter ---------- none Control signal (RTS, CTS)------ available

18.2. Outline of Remote Control

18.2.1. Outline

- When the personal computer inputs a command to EXEA controller in the remote control mode, it returns some signals in accordance with the commands to the personal computer.

Figure 18-2



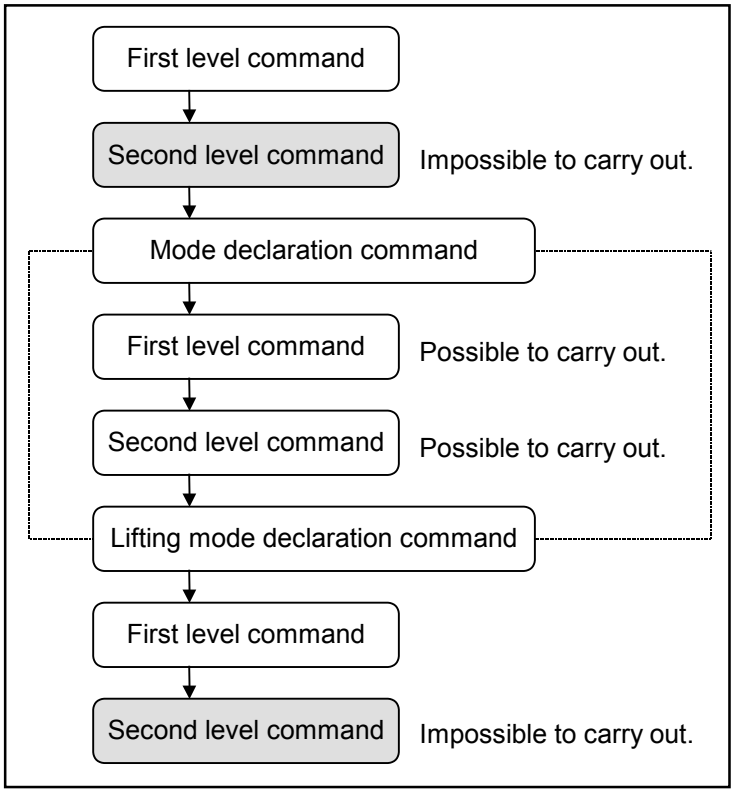
- Commands and responses are written in ASCII coded characters. For details of command and response, refer to "18.4.2. Description of Command."

- Put **CR** (carriage return) code to the end of commands. **CR** code concludes the command.

- The commands in remote control are classified into three categories as shown below. (Refer to Figure 18-3.)

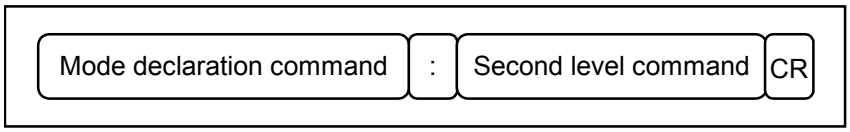
  First level command : A command itself has the meaning and is always executable such as instructing "Servo on" or "Emergency stop."
  Mode declaration command : This is to declare to use second level command.
  Second level command : This command can be executed when it is specified the command mode by a mode declaration command.

Figure 18-3



- A normal second level command is carried out after specifying command mode. However, you can combine the two commands, mode declaration and second level command, into one string of command characters and execute them. (Refer to Figure 18-4.)

Figure 18-4



- Each second level command belongs to a command mode. It cannot be used under different command mode. Be careful that, in some mode declaration commands, there may be the same name second level commands with different function. To complete the current command mode, it is necessary to execute the command to remove declaration (QUIT command) or perform other mode declaration command.

◆ Editing string of command characters

- BS code (08h) can delete letters in a character string of command. A letter in front of the BS code is an object to be deleted. However, if the BS code is in the top of a character string, it is meaningless.
  In case of character train shown below, the final character string of command will be "ABCEF."

A	B	C	D	BS	E	F	CR	→	A	B	C	E	F	CR
			↑	↑										
			Delete											

◆ Output of prompt

Command----- | S | V | O | N | CR |

- In remote control communication, the standard prompt "∗" CR is always returned for a command. However, there may be a character string of response in front of the prompt.
  (Example 1)

  Command--- | S | V | O | N | CR |

  Response --- | ∗ | CR |

  (Example 2)

  Command--- | P | O | S | CR |

  Response --- | U | 1 | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 | 0 | CR |

  | ∗ | CR | (In case of two axes combination.)

- If the character string of command cannot be carried out due to an error, the controller returns "error message" followed by the standard prompt. Four digits of numerals following ? are an error code. (Refer to ERR command.)
  (Example)

  Command--- | A | B | C | D | CR |

  Response --- | ? | 0 | 2 | 0 | 1 | CR | ← Error code output

  | ∗ | CR |

- When a command cannot be executed due to the controller problems, even a character string of command is normal, or the execution of command is interrupted somehow, an alarm output followed by the standard prompt is sent back from EXEA controller. Two digits of letters behind ! are an alarm code. To check the details of the alarm, use ALM command.
  (Example)

  Command--- | A | B | C | D | CR |

  Response --- | ! | F | 5 | CR | ← Alarm code output

  | ∗ | CR |

- If a second level command is effective by the execution of mode declaration command, a mode name is put infront of the standard prompt.
  (Example)

  Command--- | A | B | C | CR |

  Response --- | A | B | C | > | ∗ | CR |

18.2.2. Caution for Remote Control

◆ CN3: Control Input / Output

- When getting into the remote control mode, activate EREM input of CN3 control input. In the remote control operation, all CN3 control input excluding EMST input are invalid. The control outputs are activated in the same manner as in the external control mode.

◆ Servo ON / OFF

- The motor must be in "servo on" state to execute a move command or perform a move command in programmed operation by RUN input. Make the motor in servo on by inputting SVON command, then input a move command.

  Note : If a move command is inputted when the servo is off, the EXEA controller returns abnormal message and gets into the alarm state.

◆ Error response

- Input of a character unspecified in the command or a command which is restricted to enter will have "error response" from EXEA controller.

  → Returns a character string start with [?].

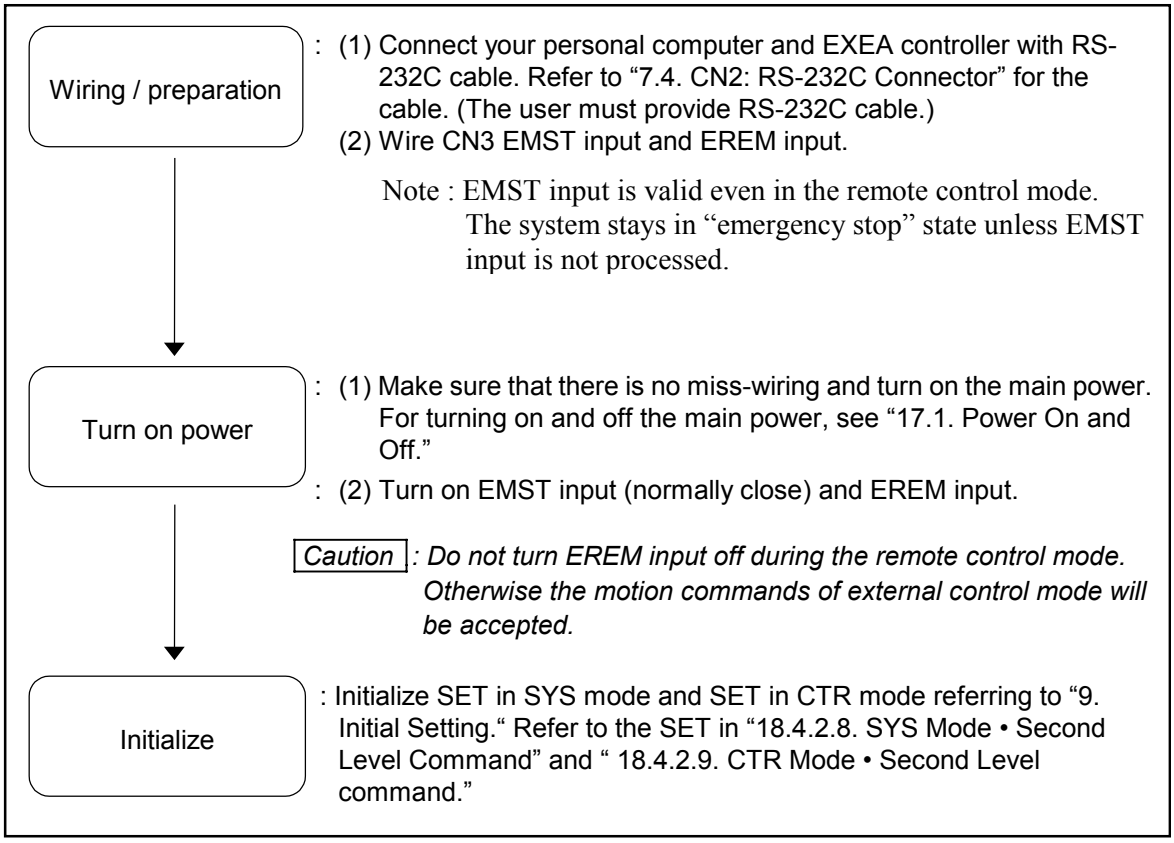
- When an alarm is detected in EXEA controller in the middle of operation, or when a command to lead an alarm is inputted, EXEA controller returns "error response."

  → Returns a character string start with [!].

18.3. Startup of Remote Control

18.3.1. Startup Procedure

Figure 18-5.



18.4. Operation Command of Remote Control

18.4.1. Command List

Table 18-2 (1/2)

Classification	Command		Description
Operational Command (First level command)	SVON		Activates the motor servo. Cannot carry out a motion command if the servo is off.
	SVOF		Deactivates the motor servo.
	EMST ⋆		Emergency stop
	ACLR		Clears alarms currently issued.
	VER ⋆		Displays the system reference number.
	ERR ⋆		Reports an error information in the communication.
	ALM ⋆		Reports alarms currently issued.
	POS ⋆		Indicates current position.
	IOP ⋆		Indicates status of input and output ports.
	UNT ⋆*		Indicates the number of operating units.
Withdrawal mode declaration	QUIT *		Cancels the mode declaration.
Mode Declaration Command and Second Level Command	MOT		Gets in the remote control operation mode.
	MOT mode	SPD	Sets motion speed and acceleration / deceleration.
		TYP	Sets the format of motion.
		NOF	Specifies offset value of position register.
		PBS	Specifies the starting point.
		ESCZ *	Specifies the prohibited area for motion of Z axis.
		SRV	Switches servo on / off.
		HOM	Starts the Home return.
		MOV	Starts the linear interpolation. **
		MOVM	Starts the continuous linear interpolation. ** (multi-point)
		ARC *	Starts the arc interpolation (defined by three points).
		CIR *	Starts the circular interpolation (defined by three points).
		MSTP	Stops operation.
		MSPD	Changes speed and acceleration.
		MSTS	Reads out the state of motion.
		PALI	Initializes pallet data.
		PALM *	Starts palletizing.
		PALN *	Specifies palletizing position number.
		QSTS *	Reads out the status of palletizing.
		LDS	Reads out the system state.
		LD	Substitutes data.
		CAL	Calculates data.
		TCH	Substitutes current coordinate data for specified variable.
		OUT	Controls output port.
		INP	Reads out state of input port.
		LCAL	Executes logical operation of data.
		DAT ⋆	Indicates state of variables.

⋆ Effective command even the EREM input is off.
* Not applicable for a single axis system.
** In a single axis system, it will be a linear motion.

Table 18-2 (2/2)

Classification	Command		Description
Mode Declaration command and Second Level Command	RUN		Sets to programmed operation mode.
	RUN mode	LST ⋆	Indicates status of program registration.
		STA	Starts programmed operation.
		STP	Stops programmed operation.
		CSTP	Stops a cycle of programmed operation.
		HLD	Holds programmed operation.
		STS	Indicates status of operation.
	TCH		Gets in setting mode of coordinate data.
	TCH mode	LST ⋆	Indicates coordinate data.
		SET	Register coordinate data.
		CLR	Initializes coordinate data.
		CPY	Copies coordinate data.
		SAV	Stores edited data.
		LOD	Loads stored coordinate data.
	EDT		Gets in program editing mode.
	EDT mode	LST ⋆	Indicates program data.
		SET	Register program data.
		CLR	Initializes program data.
		DEL	Deletes program data.
		INS	Inserts program data.
		CPY	Copies program data.
		SAV	Stores program data.
		LOD	Loads stored program data.
	PAL *		Gets into palletizing data editing mode.
	PAL * mode	LST ⋆	Indicates palletizing data.
		SET	Registers palletizing data.
		CLR	Initializes palletizing data.
		CPY	Copies palletizing data.
		TCH	Develops palletizing data to point register.
		SAV	Stores edited palletizing data.
		LOD	Loads stored palletizing data.
	SYS		Gets into parameter editing mode.
	SYS mode	LST ⋆	Indicates parameter data.
		SET	Registers parameter data.
		CLR	Initializes parameter data.
		SAV	Stores edited parameter data.
		LOD	Loads stored parameter data.
	CTR		Gets into controller setting mode.
	CTR mode	LST ⋆	Indicates set values of controller.
		SET	Register controller setting values.
		CLR	Initializes controller setting value.
		MDL ⋆	Reports module name list.
		TYL ⋆	Reports module combination list.
		PWL ⋆	Reports list of power amplifier type.
		SAV	Stores edited data.
		LOD	Loads stored data.
	FNC		Gets into control mode of special function.
	FNC mode	INI	Initializes controller.

⋆ Effective command even EREM input is off.
* Not applicable for a single axis system.

18.4.2. Description of Command

- This section is to describe the function of command and its communicating data.
- [Multi-axis] is indicated on the right side of a command name that is only applicable to the multi-axis combination. A command with no indication is common to the multi-axis combination and the single axis system.
- Function of command of which name is the same as program command has basically the same function as well.
- A command mode, to which a second level command belongs, is indicated on the left side of its name.

18.4.2.1. First Level Command

SVON: Servo on

- This command makes the system ready for "Servo on." However, when the SRV command (programmed operation or remote control command) has set to servo-off, the system cannot get in servo on.

Command-----	S	V	O	N	CR
Response -----	*	CR			

- When switching servo state through remote communication is prohibited (CN3: EREM input is off to prohibit remote control), input of SVON command will lead to an error state.

Command-----	S	V	O	N	CR	
Response -----	?	0	2	0	2	CR
	*	CR				

SVOF: Servo off

- Makes the system in "servo off" state. (prohibitive state of servo on)

Command-----	S	V	O	F	CR
Response -----	*	CR			

- When switching servo state through remote communication is prohibited (CN3: EREM input is off to prohibit remote control.), input of SVOF command will lead to an error state.

Command-----	S	V	O	F	CR	
Response -----	?	0	2	0	2	CR
	*	CR				

EMST: Emergency stop

- Puts the system into the emergency stop state.

Command-----	E	M	S	T	CR
Response -----	*	CR			

ACLR : Alarm clear

- Clears the program error alarms which are activated in EXEA controller.

  Command----- | A | C | L | R | CR |

  Response ----- | * | CR |

- To clear all alarms in EXEA controller, add "ALL" following to ACLR command. (There are some commands that cannot be cleared by this command.)

  Command----- | A | C | L | R | | A | L | L | CR |

  Response ----- | * | CR |

VER: Report reference number

- Reports the system reference number.

  Command----- | V | E | R | CR |

  Response ----- | T | y | p | . | E | X | E | 3 | CR | (In case of 3 axes controller)

  | V | e | r | . | 0 | 0 | 0 | 1 | . | 0 | CR |

  | | | | | 0 | 0 | 0 | 1 | . | 0 | / | 0 | 0 | 0 | 1 | . | 0 | CR |

  | * | CR |

ERR: Read out error state

- Reports the current error state in the remote communication.

  Note: The error codes reported by this command are listed in Table 18-3. Be aware that an error is not an alarm arisen in the EXEA controller, but an error in the remote communication.

  Command----- | E | R | R | CR |

  Response ----- | 0 | 0 | 0 | 0 | CR | ← Indication of an error code.

  | * | CR |

- Adding "ALL" to the end of the command will make to report the history of error state (eight maximum). It reports error code in reverse order of occurence. Each error report is accompanied by its history code.

  Command----- | E | R | R | | A | L | L | CR |

  Response ----- | 1 | | 0 | 0 | 0 | 0 | CR | ← Indication of an error code.

  ↓

  | 8 | | 0 | 0 | 0 | 0 | CR | ← Indication of an error code.

  | * | CR |

- Adding a numeral from 1 to 8 as an error code to the command is to report the history of specified error code. The numeral of 1 is to report the code of the latest error.

  Command----- | E | R | R | | 3 | CR |

  Response ----- | 3 | | 0 | 0 | 0 | 0 | CR | ← Indication of an error code.

  | * | CR |

- Adding "DAT" to the command is to indicate total number of errors in the error history.

Command-----	E	R	R		D	A	T	CR
Response -----	3	CR						
	*	CR						

- Table 18-3 is the list of error code.

Table 18-3

0000	No error
0100	Alarm in the controller
02xx	Command input error (Listed below, 0201 ~ 020A)
01	No command
02	Execution impossible: No.1. (Syntax is correct but the command is banned.)
03	Syntax error
04	Data range error
05	Too long string of command character
06	Execution impossible: No. 2. (Cannot start as the command is currently in execution.)
07	Execution impossible: No.3. (Cannot stop as the system has stopped already.)
08	Execution impossible: No. 4. (Cannot start as the buffer for starting is full.)
09	Execution impossible: No. 5. (Does not correspond to inputted data format.)
0A	Cannot process program code due to full capacity of the memory.
03xx	Transmitting data error (Listed below, 0301 ~ 0309)
01	Check sum error
02	Wrong character in record
03	Existence of irregular record
04	Record data error
05	Reserved
06	Reserved
07	Reserved
08	Reserved
09	Error on specified address

ALM: Read out alarm state

- This is to report current state of alarms in EXEA controller. It reads out all alarms currently arisen. The reports indicate in order of an alarm code, an alarm description and a history code.

Command----- | A | L | M | CR |

Response ----- | F | 4 | : | E | m | s | t | | ~ | | , | 0 | 0 | 0 | 0 | CR |

↓

| F | 5 | : | P | r | g | (| C | ~ |) | , | 0 | 0 | 0 | 0 | CR |

| * | CR |

- Adding "ALL" to the command is to report the history of alarm state in the past. (31 alarms maximum) It indicates in reverse order of occurence in turn. Each report of the alarm is accompanied by a history code. If there is no alarm history, only the number is indicated.

Command----- | A | L | M | | A | L | L | CR |

Response ----- | 0 | 1 | | F | 4 | : | E | m | s | t | | ~ | | , | 0 | 0 | 0 | 0 | CR |

| 0 | 2 | | F | 5 | : | P | r | g | (| C | ~ |) | , | 0 | 0 | 0 | 0 | CR |

| 0 | 3 | CR |

↓

| 3 | 1 | CR |

| * | CR |

- Adding a numeral from 1 to 31 as the history code to the command is to indicate the alarm code of specified history code. Adding a numeral of 1 is to report the latest error state.

Command----- | A | L | M | | 0 | 2 | CR |

Response ----- | 0 | 2 | | F | 5 | : | P | r | g | (| C | ~ |) | , | 0 | 0 | 0 | 0 | CR |

| * | CR |

- Adding DAT to the command is to indicate the number of data that can record the alarm history. It can record actually 31 data, even 32 is on the indication.

Command----- | A | L | M | | D | A | T | CR |

Response ----- | 3 | 2 | CR |

| * | CR |

- The alarm history is renewed every time an alarm arise in the controller. You cannot clear the alarm history in the external control mode. Use the teaching box to clear the history. (Refer to "13.7.2. Indication of Alarm History.")

POS: Read out position data

- This is to report position data of a unit specified by UNT command. The unit number is indicated on the top of report, and followed by the coordinate data of each axis.

Command-----	P	O	S	CR																
Response -----	U	1		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0	0	CR
	*	CR																		

- For position data of a particular unit, specify its unit number to the command.

Command-----	P	O	S		U	2	CR													
Response -----	U	2		X	0	0	0	0	.	0	0		Z	0	0	0	0	.	0	0	CR
	*	CR																		

- Adding "ALL" to the command is to report position data of all units connected to the controller. (multi-axis only)

Command-----	P	O	S		A	L	L	CR												
Response -----	U	1		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0	0	CR
	U	2		X	0	0	0	0	.	0	0		Z	0	0	0	0	.	0	0	CR
	U	3		Y	0	0	0	0	.	0	0		Z	0	0	0	0	.	0	0	CR
	*	CR																		

IOP: Report input / output state

- This is to report current state of all input / output ports. It reports name of a port and a port pattern in 8 bit.

Command-----	I	O	P	CR										
Response -----	I	P	0	0		0	0	0	0	0	0	0	0	CR
	I	P	1	0		0	0	0	0	0	0	0	0	CR
								↓						
	O	P	1	1		0	0	0	0	0	0	0	0	CR
	*	CR												

- For state of a particular port, specify a port number to the command.

Command-----	I	O	P		I	P	1	0	CR					
Response -----	I	P	1	0		0	0	0	0	0	0	0	0	CR
	*	CR												

- For controlling state of input and output ports, use OUT and INP command in MOT mode.

UNT: Specify motion unit number *[Multi-axis]*

- This is to specify a motion unit number.

  Command-----| U | N | T | | U | 2 | CR |

  Response -----| * | CR |

- If the only UNT command is inputted, it outputs a unit number currently selected.

  Command-----| U | N | T | CR |

  Response -----| U | 2 | CR |

  | * | CR |

- If none of unit is specified by the command U1 is valid as the initial setting.

18.4.2.2. Mode Declaration Command

QUIT: Withdrawal of mode declaration command

- Withdraw a current declaration of the command mode. This command cannot be carried out while executing the remote control operation or programmed operation. Declaration of the mode will be withdrawn after execution of these operations.

Command (1) -	A	B	C	CR	← Declared command mode		
Response (1)--	A	B	C	>	*		CR
Command (2) -	Q	U	I	T	CR	← Withdrawal of declaration	
Response (2)--	*	CR					

MOT: Remote control operation mode

- This is to specify an operation mode that controls directly the operations such as Home return or a linear interpolation. Switching mode by this command is invalid during execution of the remote control operation in any manner.

Command-----	M	O	T	CR		
Response -----	M	O	T	>	*	CR

RUN: Programmed operation mode

- This is to specify a mode to start a programmed operation stored in the EXEA controller memory.

Command-----	R	U	N	CR		
Response -----	R	U	N	>	*	CR

- This command is invalid while the other programmed operation is executed.

TCH: Teaching mode

- This is to specify a mode to edit coordinate data.

Command-----	T	C	H	CR		
Response -----	T	C	H	>	*	CR

EDT: Program editing mode

- This is to specify a mode to edit program.

Command-----	E	D	T	CR		
Response -----	E	D	T	>	*	CR

PAL: Palletizing editing mode *[Multi-axis]*

- Specifies a mode to edit palletizing.

Command----- | P | A | L | CR |

Response ----- | P | A | L | > | * | CR |

SYS: Parameter editing mode

- This is to specify a mode to initialize parameters (editing parameters).

Command----- | S | Y | S | CR |

Response ----- | S | Y | S | > | * | CR |

CTR: Controller setting mode

- Specifies a mode to set function of the controller.

Command----- | C | T | R | CR |

Response ----- | C | T | R | > | * | CR |

FNC: Special function operating mode

- This is to specify a mode to operate special function of the controller.

Command----- | F | N | C | CR |

Response ----- | F | N | C | > | * | CR |

18.4.2.3. MOT Mode • Second Level Command

- Refer to "15.2.7.3. Program Command" for details of each commands.

MOT mode ***SPD: Set motion speed and acceleration / deceleration***

- Specifies the interpolation speed, acceleration and deceleration for a motion unit currently specified by UNT command. Unit number can be omitted when its speed, acceleration and deceleration are unchanged. Execution of the command, which is attempted during the specified unit is in motion, is not valid until its next motion command.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	S	P	D		S	1	0	0		A	1	0		B	1	0	CR
Response -----	M	O	T	>	*	CR											

- Add a particular unit number to the command when changing its speed.
  (multi-axis combination only)

Command-----	S	P	D		U	2		S	1	0	0		A	1	0	CR
Response -----	M	O	T	>	*	CR										

MOT mode ***TYP: Set motion format***

- This is to specify the motion format of a motion unit specified by UNT command. When the command is input during an objective unit is in motion, the command is invalid till it is ready for the next start.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	T	Y	P		&	A	T	F	P	E	L	CR
Response -----	M	O	T	>	*	CR						

- When changing the motion format of a particular unit, add its unit number to the command.
  (multi-axis combination only)

Command-----	T	Y	P		U	2		&	A	T	F	P	E	L	CR
Response -----	M	O	T	>	*	CR									

MOT mode ***NOF: Set off-set value of point register***

- This is to specify off-set number of point register of a motion unit currently selected by UNT command. When the command is inputted during the unit is in motion, it is invalid until the unit starts the next motion.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	N	O	F		#	1	0	0	CR
Response -----	M	O	T	>	*	CR			

- When changing off-set value of a particular unit, add its unit number to the command.
  (multi-axis combination only)

Command-----	N	O	F		U	2		#	1	0	0	CR
Response -----	M	O	T	>	*	CR						

MOT mode PBS: Setting working reference point

- Specifies the working reference point of a currently specified motion unit by UNT. When the objective unit is in motion, the command is invalid until it starts the next operation.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command 1 -- | P | B | S | | P | 0 | 0 | 0 | 0 | CR | ← Specify point register

Command 2 -- | P | B | S | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 | 0 |

| CR | ← Specify coordinate data directly

Response ----- | M | O | T | > | * | CR |

- When changing the setting of the working reference point of a particular unit, add its unit number to the command. (multi-axis combination only)

Command----- | P | B | S | | U | 2 | | P | 0 | 0 | 0 | 0 | CR |

Response ----- | M | O | T | > | * | CR |

MOT mode ESCZ: Set off-limits area of Z axis [Multi-axis]

- This is to specify upper and lower off-limits boundary and turnout position of Z axis of a motion unit currently specified by UNT command. When the command is input during the unit is in motion, it is invalid until the unit starts the next motion.

Command (1) - | E | S | C | Z | | P | X | 0 | 0 | 0 | 0 | | P | Y | 0 | 0 | 0 | 0 | | P | Z |

| 0 | 0 | 0 | 0 | CR | ← Specify point register.

* This is an example of setting upper and lower off-limits boundary and turnout position of point number 0000 by X, Y and Z axes coordinates.

Command (2) - | E | S | C | Z | | 0 | 0 | 0 | 0 | . | 0 | 0 | | 0 | 1 | 0 | 0 | . | 0 | 0 | |

| 0 | 1 | 2 | 5 | . | 0 | 0 | CR | ← Specify coordinates data directly.

* This is an example of setting off-limits area in order of upper, lower boundary and turnout position by coordinate data directly.

Response ------ | M | O | T | > | * | CR |

- When specifying off-limits area of a particular unit, add its unit number to the command.

Command----- | E | S | C | Z | | U | 2 | | 0 | 0 | 0 | 0 | . | 0 | 0 | | 0 | 1 | 0 | 0 | . |

| 0 | 0 | | 0 | 1 | 2 | 5 | . | 0 | 0 | CR | ← Specify coordinates data directly.

Response ----- | M | O | T | > | * | CR |

MOT mode SRV: Switch servo on / off

- Switches servo on and off of the designated axis of motion unit currently specified by UNT command.
  However, to make it works, the controller must be in the "permitting servo on" state by SVON command.
  The command works only in MOT mode.

Command-----	S	R	V		Designate axis		CR
Response -----	M	O	T	>	*		CR

* Designate axis ---- Specify servo on (1) or servo off (0) of each axis unit such as "X1 Y1 Z1" or "X0 Y0 Z0."
Default axis unit does not change its state of servo.

Command-----	S	R	V		X	1	CR
Response -----	M	O	T	>	*		CR

- To specify a particular motion unit, add its number to the command.

Command-----	S	R	V		U	1		Designated axis	CR
Response -----	M	O	T	>	*				CR

← Change state of servo of unit 1.

- Add ON or OFF to the command to switch the servo state of all units that are connected to the controller.

Command-----	S	R	V		O	N	CR		← Servo on all units
	S	R	V		O	F	F	CR	← Servo off all units
Response -----	M	O	T	>	*		CR		

MOT mode ***HOM: Start home return***

- Starts Home return of a motion unit currently specified by UNT command. When the system enables to start Home return normally, the prompt changes from "*" to "#." The prompt changes to "*" again when Home return is completed.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	H	O	M	CR			
Response -----	M	O	T	>	#	CR	← Executing home return.
	M	O	T	>	*	CR	← Home return complete.

- To start Home return of a particular motion unit, add its unit number to the command. (multi-axis combination only)

Command-----	H	O	M		U	2	CR
Response -----	M	O	T	>	#	CR	

- Add a particular axis number to the command to start its Home return. You may omit unit number. (multi-axis combination only)

Command-----	H	O	M		U	2		X	CR
Response -----	M	O	T	>	#	CR			

- Add "ALL" to the command to start Home return of all units that are connected to the controller. You can not specify a particular axis to this command. (multi-axis combination only)

Command-----	H	O	M		A	L	L	CR	
Response -----	M	O	T	>	#	CR			

MOT mode MOV: Start linear interpolation * *(In a single axis system this is regarded as linear motion.)*

- Starts linear interpolation of motion unit currently specified by UNT command. When the system enables to start the linear interpolation normally, the prompt changes from "*" to "**#.**" If the linear interpolation is completed, the prompt returns to "*" again.

Command-----	M	O	V		P	0	0	0	1	CR
Response -----	M	O	T	>	#	CR	← Executing linear interpolation.			
	M	O	T	>	*	CR	← Complete linear interpolation.			

- To start linear interpolation of a particular motion unit, add its unit number to the command.

Command-----	M	O	V		U	2		P	0	0	0	1	CR
Response -----	M	O	T	>	#	CR							

- It is possible to specify coordinate values of each point directly instead of inputting the point register. Input the data for operating axis only.

Command-----	M	O	V		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0	0
	CR																			
Response -----	M	O	T	>	#	CR														

- It is possible to specify speed, acceleration, deceleration and motion format. However, these settings are valid for this command only, but not for other operation.

Command-----	M	O	V		P	0	0	0	1		S	6	0	0	CR
Response -----	M	O	T	>	#	CR									

MOT mode MOVM: Start continuous linear interpolation * (multi-point)

**** This is regarded as continuous linear motion (multi-point) in a single axis system.***

- Starts continuous linear interpolation of motion unit currently specified by UNT command. When the continuous linear interpolation starts normally, the prompt changes from "*" to "**#.**" When the continuous linear interpolation is completed, the prompt returns to "*" again.

Command-----	M	O	V	M	P	0	0	0	1		P	0	0	1	1	CR
Response -----	M	O	T	>	#	CR	← Executing continuous linear interpolation.									
	M	O	T	>	*	CR	← Complete continuous linear interpolation.									

- To start continuous linear interpolation of a particular motion unit, add its unit number to the command. (multi-axis combination only)

Command-----	M	O	V	M		U	2		P	0	0	0	1		P	0	0	1	1	CR
Response -----	M	O	T	>	#	CR														

- It is possible to specify speed, acceleration, deceleration and motion format. However, these settings are valid for this command only, but not for other operation.

Command-----	M	O	V	M		P	0	0	0	1		P	0	0	1	1		S	6	0
	0	CR																		
Response -----	M	O	T	>	#	CR														

MOT mode ARC: Start circular arc interpolation [Multi-axis]

- Starts circular arc interpolation of a motion unit currently specified by UNT command. When the arc interpolation starts normally, the prompt changes from "∗" to "**#**." As the arc interpolation is completed, the prompt returns to "∗" again.

```
Command-----  A R C   P 0 0 0 1   P 0 0 0 2   P 0 0 0 3
              CR
Response -----  M O T > # CR  ← Executing arc interpolation.
                M O T > ∗ CR  ← Complete of arc interpolation.
```

- To start an arc interpolation of a particular motion unit, add its unit number to the command.

```
Command-----  A R C   U 2   P 0 0 0 1   P 0 0 0 2   P 0
              0 0 3 CR
Response -----  M O T > # CR
```

- It is possible to specify speed, acceleration, deceleration and motion format. However, these settings are valid for this command only, not valid for other operation.

```
Command-----  A R C   P 0 0 0 1   P 0 0 0 2   P 0 0 0 3
                S 6 0 0 CR
Response -----  M O T > # CR
```

MOT mode CIR: Start circular interpolation [Multi-axis]

- Starts a circular interpolation of motion unit currently specified by UNT command. When the circular motion starts normally, the prompt changes from "∗" to "**#**." As the circular motion is completed, the prompt returns to "∗" again.

```
Command-----  C I R   P 0 0 0 1   P 0 0 0 2   P 0 0 0 3
              CR
Response -----  M O T > # CR  ← Executing circular motion.
                M O T > ∗ CR  ← Complete of circular motion.
```

- To start a circular motion of a particular motion unit, add its unit number to the command.

```
Command-----  C I R   U 2   P 0 0 0 1   P 0 0 0 2   P 0
              0 0 3 CR
Response -----  M O T > # CR
```

- It is possible to specify speed, acceleration, deceleration and motion format. However, these settings are valid for this command only, but not for other operation.

```
Command-----  C I R   P 0 0 0 1   P 0 0 0 2   P 0 0 0 3
                S 6 0 0 CR
Response -----  M O T > # CR
```

MOT mode MSTP: Motion stop

- Decelerates and stops motion of motion unit currently specified by UNT command. The prompt "#" is outputted when decelerating and returns to prompt "*" after the unit stops. This command is meaningless when the motion unit is not moving. (It does not lead to an alarm.)
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	M	S	T	P	CR		
Response -----	M	O	T	>	#	CR	← Decelerating the motion unit.
	M	O	T	>	*	CR	← Stopped.

- To stop motion of a particular motion unit, add its unit number to the command.
  (multi-axis combination only)

Command-----	M	S	T	P		U	2	CR
Response -----	M	O	T	>	#	CR		

- When stopping a particular axis unit, add its axis name. The unit number can be a default.
  (multi-axis combination only)

Command-----	M	S	T	P		U	2		X	CR
Response -----	M	O	T	>	#	CR				

- Add "ALL" to the command to stop all motion units connected to the controller. In this case, you cannot specify an axis. (multi-axis combination only)

Command-----	M	S	T	P		A	L	L	CR
Response -----	M	O	T	>	#	CR			

MOT mode MSPD: Reserved (cannot use)

MOT mode MSTS: Read out motion state

- This is to substitute the motion state of a motion unit currently specified by UNT command to a data register. (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command-----	M	S	T	S		D	0	0	0	CR
Response -----	M	O	T	>	*	CR				

- When substituting the motion state of specified unit to a data register, specify an objective unit name as shown below. (multi-axis combination only)

Command-----	M	S	T	S		D	0	0	0		=		U	2	CR
Response -----	M	O	T	>	*	CR									

- Motion state number to be substituted has meaning as shown below.

  0 ----------Stopping

  1 ----------In motion

MOT mode PALI: Initialize palletizing operation [Multi-axis]

- This is to correlate the palletizing operation number (QN00) and the pallet data number to enable a palletizing operation of a motion unit currently specified by UNT command.

Command-----	P	A	L	I		Q	N	0	0		#	0	0	CR
Response -----	M	O	T	>	*	CR								

- You may specify a name of pallet data instead of a palletizing operation number.

Command-----	P	A	L	I		Q	N	0	0		$	P	A	L	N	A	M	E	CR
Response -----	M	O	T	>	*	CR													

- For execution to initialize palletizing operation of a particular motion unit, specify its unit number to the command.

Command-----	P	A	L	I		U	2		Q	N	0	0		#	0	0	CR
Response -----	M	O	T	>	*	CR											

- It is possible to specify a motion format. However, only &A and &I are valid. These are effective to this command only, but not for other motions.

Command-----	P	A	L	I		U	2		Q	N	0	0		#	0	0		&	A	CR
Response -----	M	O	T	>	#	CR														

MOT mode PALM: Palletizing operation: positioning [Multi-axis]

- This is to start the motion of designated palletizing operation number to a pallet position. When operation starts normally, the prompt changes from "*" to "**#**." As the operation is completed, the prompt returns to "*" again and reset the palletizing operation counter.

Command-----	P	A	L	M		Q	N	0	0	CR
Response -----	M	O	T	>	#	CR	← Palletizing in motion.			
	M	O	T	>	*	CR	← Palletizing completed.			

- Specify a pallet position number with QP××× when moving to a particular pallet position.

Command-----	P	A	L	M		Q	N	0	0		Q	P	0	0	0	CR
Response -----	M	O	T	>	#	CR										

- It is possible to specify speed, acceleration / deceleration and motion format. However, those settings are effective to this command only, but not for other operation.

Command-----	P	A	L	M		Q	N	0	0		S	6	0	0	CR
Response -----	M	O	T	>	#	CR									

MOT mode PALN: Palletizing operation; change positioning number [Multi-axis]

- Changes the setting of palletizing positioning counter of a specified palletizing operation number. Normally a palletizing sequential operation is performed by PALM command. This PALN command is to change the sequential order of the operations. When the command is inputted while the palletizing operation of a designated palletizing operation number is executed, it is not effective until the next PALM command is on.

Command-----	P	A	L	N		Q	N	0	0		#	2	2	CR
Response -----	M	O	T	>	*	CR								

MOT mode QSTS: Palletizing operation: read out the palletizing state [Multi-axis]

- This command is to substitute a state of palletizing (number of palletizing points) of a specified palletizing operation number to a data register.

Command-----	Q	S	T	S		D	0	0	0		=		Q	N	0	0		Q	P	M	CR
Response -----	M	O	T	>	*	CR															

- Substitute a state of palletizing (point number of the next palletizing) of a specified palletizing operation to a data register.

Command-----	Q	S	T	S		D	0	0	0		=		Q	N	0	0		Q	P	C	CR
Response -----	M	O	T	>	*	CR															

* The output of point number is normally in the range of 0 (zero) to (QPM - 1). However the output is " -1 (minus)" if all palletizing operation have completed.

- Substitute coordinates data of palletizing point of a specified palletizing operation number to a point register.

Command-----	Q	S	T	S		P	0	0	0	0		=		Q	N	0	0		Q	P	0
	0	0	CR																		
Response -----	M	O	T	>	*	CR															

MOT mode LDS: Read out, system setting state

- This is to output the state of system setting (offset value of point number) of a unit currently specified by UNT command then, substitute it for a designated variables (data register or point register). As the variable is shared with the programmed operation, you must be very careful when execute the command simultaneously with the programmed operation.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command 1 --	L	D	S		D	0	0	0		=		N	O	F	CR	
Command 2 --	L	D	S		P	0	0	0	0		=		P	B	S	CR
Response -----	M	O	T	>	*	CR										

- To read out system setting state of a particular unit, specify its unit number to the command. (multi-axis combination only)

Command-----	L	D	S		D	0	0	0		=		U	1		N	O	F	CR
Response -----	M	O	T	>	*	CR												

- Changing a code of the command enables to read out the state of motion setting.

  1) Data to be substituted to a specified data register.

N	O	F	← Offset value of point number
S	P	D	← Speed (numerical number) setting
S	P	D	R ← Setting of motion speed (percentage reading)
A	C	C	← Acceleration (numerical number) setting
A	C	C	R ← Acceleration (percentage reading) setting
D	A	C	← Deceleration (numeric reading) setting
D	A	C	R ← Deceleration (percentage reading)

  2) Data to be substituted to a specified point register.

P	B	S	← Setting of working reference point

  3) Data to be substituted to a specified axis in a point register.
  (multi-axis combination only)

E	S	C	Z		U	P	R	← Off-limits area of Z axis (upper boundary)
E	S	C	Z		L	W	R	← Off-limits area of Z axis (lower boundary)
E	S	C	Z		P	O	S	← Off-limits area of Z axis (turnout position)

MOT mode LD: Substitute data

- This is to substitute numerical value or contents of point register for a designated variable. As the variable is shared with programmed operation, you must be very cautious to execute this command simultaneously with programmed operation.

Command 1 --	L	D		D	0	0	0		=		D	0	0	0	CR							
Command 2 --	L	D		P	0	0	0	0		=		X	0	0	0	0	.	0	0		Y	
	0	0	0	0	.	0	0	CR														
Response -----	M	O	T	>	*	CR																

MOT mode CAL: Calculate data

- This is to execute calculations of numerals or each content between data registers and to substitute the results for the designated variables (data or point register). As the variables are shared with programmed operation, you must be very careful when execute this command simultaneously with the programmed operation.

Command-----

C	A	L		D	0	0	0		=		D	0	0	0		+		#	1	0
0	CR																			
C	A	L		D	0	0	0		=		D	0	0	0		–		#	1	0
0	CR																			

Response -----

M	O	T	>	*	CR

- If the results are unable to substitute for the variables, an alarm will be given.

Command-----

C	A	L		D	0	0	0		=		D	0	0	0		+		#	1	0
0	CR																			

Response -----

!	F	5	CR

← Alarm code

M	O	T	>	*	CR

- The following code for calculation are available.

+	← Addition
–	← Subtraction
*	← Multiplication
/	← Division
%	← Percentage

MOT mode TCH: Substitute current coordinate data for specified variable

- This is to substitute the current position data of a unit currently specified by UNT command for the specified variable. As the variables are shared with programmed operation, be careful when execute this command simultaneously with the programmed operation.

Command-----

T	C	H		P	0	0	0	0		=		X	1		Y	0	CR

Response -----

M	O	T	>	*	CR

- For reading out current position of a particular unit, specify its unit number to the command. (multi-axis combination only)

Command-----

T	C	H		P	0	0	0	0		=		U	1		X	1		Y	0	CR

Response -----

M	O	T	>	*	CR

MOT mode OUT: Set output port

- Sets the state of specified output port.

Command 1 --	O	U	T		O	P	0	0		=		;	0	0	0	1	0	1	1	0	CR
Command 2 --	O	U	T		O	P	1	1		=		D	0	0	0	CR					
Command 3 --	O	U	T		O	B	1	0	1		=		O	N	CR						
Response -----	M	O	T	>	*	CR															

- When specifying by bit, the following code are available. (command 3 above)

O	N	← Output is on.
O	F	F ← Output is off.
R	E	V ← Reverse the output.

MOT mode INP: Get input port state

- Gets the state of specified input port to a data register.

Command-----	I	N	P		D	0	0	0		=		I	P	1	0	CR
Response -----	M	O	T	>	*	CR										

MOT mode LCAL: Execute logical operation of data

- This is to execute logical operation of numerical data or contents between data registers then, to substitute results for the specified variables. As the variables are shared with the programmed operation, be careful when executing this command simultaneously with the programmed operation.

Command-----	L	C	A	L		D	0	0	0		=		D	0	0	0		O	R		D
	0	0	1	CR																	
Response -----	M	O	T	>	*	CR															

- If the results are unable to substitute for the designated variables, an alarm will be given.

Command-----	L	C	A	L		D	0	0	0		=		D	0	0	0		O	R		D
	0	0	1	CR																	
Response -----	!	F	5	CR	← Alarm code																
	M	O	T	>	*	CR															

- The following code to indicate the way of logical operation are available.

O	R	← Logical addition
A	N	D ← Logical multiplication
X	O	R ← Exclusive OR

MOT mode *DAT: Read out state of variable*

- Read out the contents of a specified variable.

Command 1 --	D	A	T		D	0	0	1	CR											
Response 1---	D	0	0	1		1	2	3	.	0	0	CR								
	M	O	T	>	*	CR														
Command 2 --	D	A	T		P	0	0	0	1	CR										
Response 2---	P	0	0	0	1		X	0	0	0	0	.	0	0		Y	0	0	0	0	.
	0	0	CR																	
	M	O	T	>	*	CR														

- You can specify the reading range by adding (-) (dash) to data numbers.

Command-----	D	A	T		D	0	0	1	–	D	0	0	3	.	0	0	CR
Response -----	D	0	0	1		1	2	3	.	0	0	CR					
	D	0	0	2		–	2	2	0	0	.	0	0	CR			
	D	0	0	3		1	0	0	.	0	0	CR					
	M	O	T	>	*	CR											

18.4.2.4. RUN Mode • Second Level Command

RUN mode ***LST: Read out program data***

- This is to read out program data for confirmation. For details, refer to LST command in EDT mode. When you need to confirm all program data of the desired program number, the following shows how to do it.

Command-----	L	S	T		0	0	CR															
Response -----	0	0	/	N	A	M		$	P	R	G	N	A	M	E	CR							
	0	0	/	M	E	M		U	S	E	R	_	P	R	O	G	_	M	E	M	O	CR
	0	0	/	S	T	E		3	1	9	CR											
	0	0	/	0	0	0		M	O	V		P	0	0	0	1	CR					
	0	0	/	0	0	1		M	O	V		P	0	0	0	2	CR					
				↓																		
	0	0	/	3	1	9		E	N	D	CR											
	R	U	N	>	*	CR																

RUN mode STA: Start programmed operation

1 Programmed operation

- This command is effective when "RUN mode" of the parameter related to set PMD is set to "PRG" (programmed operation).

- This is to start a programmed operation specified by a program number or name of program. When the operation has started normally, the prompt changes to "**#**" from "*".

  Command ---- | S | T | A | | 0 | 0 | CR | ← Specify program number

  | S | T | A | | $ | P | R | G | N | A | M | E | CR | ← Specify name of program.

  Response----- | R | U | N | > | # | CR |

- If you require a step operation, put "S" to the end of command.

  Command ---- | S | T | A | | 0 | 0 | | S | CR |

  Response----- | R | U | N | > | $ | CR |

- When resuming a program which has been on hold state, input "R" instead of program number. Possibility of resuming the programmed operation can be checked by "Output status of programmed operation" command (STS command).

  Command ---- | S | T | A | | R | CR | ← Resume programmed operation.

  Response----- | R | U | N | > | # | CR |

- For restart the operation in "cycle stop state" or "hold state" (when the prompt is $), input simply "STA" command. An error alarm arises when a step operation command is inputted to resume the operation in cycle stop or hold state.

  Command ---- | S | T | A | CR |

  Response----- | R | U | N | > | # | CR |

2 Direct operation

- The command is effective when "RUN mode" of the parameter related to set PMD is set to "POS" (direct operation).

- In "directly inputted coordinate operation", the command starts a motion to a position of designated point register. When the motion is successful, the prompt changes to "**#** " from "*".

  Command ---- | S | T | A | | 0 | 0 | 0 | 0 | CR | ← Designate a number.

  Response----- | R | U | N | > | # | CR |

  You may specify quantity of units which can be operated simultaneously.

  Command ---- | S | T | A | | 0 | 0 | 0 | 0 | | 0 | 0 | CR | ← Designate a number.

  Response----- | R | U | N | > | # | CR |

RUN mode STP: Stop programmed operation

- This is to terminate programmed operation. When the operation has stopped, the prompt returns to "*".

Command ----	S	T	P	CR		
Response -----	R	U	N	>	*	CR

- The prompt # is output first when it takes time to decelerate and stop completely because the motor has been running at the moment of the command is input. The prompt returns to "*" after the motor stops completely.

Command ----	S	T	P	CR			
Response -----	R	U	N	>	#	CR	← Decelerating.
	R	U	N	>	*	CR	← Stop operation completed.

- If the programmed operation is not executed at the moment of inputting the command (when the prompt is *), an error message will be responded.

Command -----	S	T	P	CR		
Response -----	?	0	2	0	7	CR
	R	U	N	>	*	CR

RUN mode CSTP: Cycle stop, programmed operation

- This is to execute cycle stop of a programmed operation. The prompt changes to "$" as the system gets in "cycle stop state."

Command -----	C	S	T	P	CR	
Response -----	R	U	N	>	$	CR

- When the motor requires time for deceleration and complete stop, as it's been moving, firstly the prompt changes to " # " then the prompt changes to "$" as the motor stops completely.

Command -----	C	S	T	P	CR		
Response -----	R	U	N	>	#	CR	← Decelerating.
	R	U	N	>	$	CR	← Cycle stop completed.

- If the programmed operation is not executed at the moment of inputting the command (when the prompt is *), an error message will be responded.

Command -----	C	S	T	P	CR	
Response -----	?	0	2	0	7	CR
	R	U	N	>	*	CR

* CSTP command is effective in hold stop state as well.

RUN mode HLD: Hold programmed operation

- Execute hold stop. The prompt is "$" when the hold stop state is established.

Command-----	H	L	D	CR		
Response -----	R	U	N	>	$	CR

- When the motor requires time for deceleration as it's been moving, the prompt "**#** " is returned, then the prompt changes to " $ " as the motor stops completely.

Command-----	H	L	D	CR			
Response -----	R	U	N	>	#	CR	← Decelerating.
	R	U	N	>	$	CR	← Hold stop completed.

- If a programmed operation is not executed at the moment of inputting the command (when prompt is "*".), an error message will be responded.

Command-----	H	L	D	CR		
Response -----	?	0	2	0	7	CR
	R	U	N	>	*	CR

* HLD command is not effective in cycle stop state.

RUN mode STS: Read out state of programmed operation

■ Indicates current state of a programmed operation.

Command-----	S	T	S	CR		
Response -----	R	A	CR	← In operation (in continuous operation).		
	R	U	N	>	#	CR

- The states indicated in operation are listed below.
  1) RA: In operation (continuous operation)
  2) RS: In operation (step operation)
  3) CA: Stopping by cycle stop (continuous operation)
  4) CS: Stopping by cycle stop (step operation)
  5) HA: Stopping by hold (continuous operation)
  6) HS: Stopping by hold (step operation)
  7) SE: Ceased operation (possible to resume.)
  8) SD: Ceased operation (impossible to resume)

■ Add letter S or C to the command if you require the details of operation state. Output data for "S" comprise sequence stage, program number and step number, while output data for "C" include the contents of command in addition to the output items for "S".

Command ----	S	T	S		S	CR													
Response-----	M		0	0	2	/	0	0	1	CR									
	R	U	N	>	#	CR													
Command ----	S	T	S		C	CR													
Response-----	M		0	0	2	/	0	0	1		M	O	V		P	0	0	3	CR
	R	U	N	>	#	CR													

● Three letters of M, S and C are used to indicate stage of sequence. Each letter corresponds to the stage as shown below.

M : Main program
S : Sub-sequence (executing simultaneous operation command)
C : Secondary sequence

■ As STS command indicates all operating sequence, the display will be as shown below when the program is operated in multitask mode (in case of multi axis combination). Be careful as number of lines differ according to operation state.

Command ----	S	T	S		C	CR														
Response-----	M		0	0	2	/	0	2	1		T	I	M		0	0	~	CR	(1)	
	S		0	0	2	/	0	0	3		M	O	V		P	0	~	CR	(2)	
	C		0	1	2	/	1	0	2		L	D		P	0	3	~	CR	(3)	
	S		0	1	2	/	1	0	0		M	O	V		P	0	~	CR	(4)	
	C		0	1	4	/	0	3	2		T	I	M		0	0	~	CR	(5)	
	R	U	N	>	#	CR														

● Above example shows that (1) and (2) are main programs, (3) and (4) are secondary sequence 1, and (5) as secondary sequence 2. A sub sequence indicated by "S" is for a main or secondary sequence previously reported.

■ Letters S and C are invalid for direct operation. However you may add "P" instead. Output data for "P" are the point number and number of units.

Command ----	S	T	S		P	CR			
Response-----	M		0	0	0	2	/	1	CR
	R	U	N	>	#	CR			

* Operating data is 1 when indication of number of unit is "0", while operating data is 2 when indication of unit is 1. Indication of number is fixed to "0" for a single axis system.

18.4.2.5. TCH Mode • Second Level Command

TCH mode ***LST: Read out coordinate data of point register***

- This is to read out all coordinate data. A point number is on the top and each axis coordinates data follow.

Command ---- | L | S | T | CR |

Response----- | 0 | 0 | 0 | 0 | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 |
0	CR
0	0	0	1		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0
0	CR
↓
3	9	9	9		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0
0	CR
T	C	H	>	*	CR

- For indication of a particular coordinate data, add its point number to the command.

Command ---- | L | S | T | | 0 | 0 | 0 | 1 | CR |

Response----- | 0 | 0 | 0 | 1 | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 |
| 0 | CR |
| T | C | H | > | * | CR |

- You can specify reporting range of point number using (-) (dash).

Command ---- | L | S | T | | 0 | 0 | 0 | 1 | – | 0 | 0 | 0 | 3 | CR |

Response----- | 0 | 0 | 0 | 1 | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 |
0	CR
0	0	0	2		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0
0	CR
0	0	0	3		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0
0	CR
T	C	H	>	*	CR

- For checking available quantity of point register for more input, add a code "DAT" to the command.

Command ---- | L | S | T | | D | A | T | CR |

Response----- | 4 | 0 | 0 | 0 | CR |
| T | C | H | > | * | CR |

- If coordinate data is accompanied by unit designation data or motion format, the responses are as shown below.

Command ----- | L | S | T | | 0 | 0 | 0 | 1 | CR |

Response----- | 0 | 0 | 0 | 1 | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 | 0 | . | 0 |
| 0 | | U | 1 | | & | A | CR |
| T | C | H | > | * | CR |

TCH mode SET: Set coordinate data of point register

- This is to set coordinate data of a point register. The command changes only the data setting of a specified coordinate axis, but not for an unspecified coordinate axis. (Set X axis coordinate only for a single axis system.)

Command ----	S	E	T		0	0	9	9		X	0	0	0	0	.	0	0		Y	0	0
	0	0	.	0	0	CR															
Response-----	T	C	H	>	*	CR															

- Add the letter C to a name of axis to indicate "non-motion" (does not move) coordinate data.

Command ----	S	E	T		0	0	9	9		X	C		Y	0	0	0	0	.	0	0	CR
Response-----	T	C	H	>	*	CR															

- Add the letter T to a coordinate axis for setting its current position to a point register. When a particular unit is specified, its current position coordinate data are set. If no particular unit is specified, the current position coordinates data of a specified motion unit by UNT command are registered.

Command ----	S	E	T		0	0	9	9		X	T		Y	0	0	0	0	.	0	0	CR
Response-----	T	C	H	>	*	CR															

- To specify a motion unit, add its unit number to the end of coordinate data, or declare only its unit number.

Command 1--	S	E	T		0	0	9	9		X	0	0	0	0	.	0	0		Y	0	0
	0	0	.	0	0		U	1	CR												
Response 1 --	T	C	H	>	*	CR															
Command 2--	S	E	T		0	0	9	9		U	1	CR									
Response 2 --	T	C	H	>	*	CR															

- Input C instead of unit number to cancel specifying motion unit.

Command ----	S	E	T		0	0	9	9		U	C	CR									
Response-----	T	C	H	>	*	CR															

- Add a format code to end of coordinate data, or declare only a motion format to specify. Only A (absolute coordinates) and I (relative coordinates) are effective as a motion format.

Command 1--	S	E	T		0	0	9	9		X	0	0	0	0	.	0	0		Y	0	0
	0	0	.	0	0		&	A	CR												
Response 1 --	T	C	H	>	*	CR															
Command 2--	S	E	T		0	0	9	9		&	A	CR									
Response 2 --	T	C	H	>	*	CR															

- To cancel to specify a motion format , input C to a part of format code.

Command ----	S	E	T		0	0	9	9		&	C	CR									
Response-----	T	C	H	>	*	CR															

TCH mode CLR: Initialize coordinate data

- Initializes all coordinate data and sets all coordinate axes to "non-motion" data.

  Command ---- | C | L | R | CR |

  Response ----- | T | C | H | > | * | CR |

- To initialize only one coordinate data, specify its point number to the command.

  Command ---- | C | L | R | | 0 | 0 | 9 | 9 | CR |

  Response ----- | T | C | H | > | * | CR |

- You may set the range of initializing point numbers using a (-) (dash).

  Command ---- | C | L | R | | 0 | 0 | 9 | 9 | – | 0 | 1 | 2 | 0 | CR |

  ← Point register: Number 99 to 120 are the objectives for initialization.

  Response ----- | T | C | H | > | * | CR |

TCH mode CPY: Copy coordinate data

- This is to copy a coordinate data of a point number (left) to the designated point number (right).

  Command ---- | C | P | Y | | 0 | 0 | 0 | 9 | | 0 | 0 | 1 | 2 | CR |

  ← Copy coordinate data of point number 9 to point number 12.

  Response ----- | T | C | H | > | * | CR |

- You can specify the range of point register numbers as the copy source using (-) (dash). When the copied data range is more than the specified copy area, only the data that can be copied within the specified area will be stored.

  Command ---- | C | P | Y | | 0 | 0 | 0 | 9 | – | 0 | 0 | 1 | 2 | | 0 | 0 | 2 | 2 | CR |

  ← Copy coordinate data of point number 9 to 12 to point number 22 to 25.

  Response ----- | T | C | H | > | * | CR |

TCH mode SAV: Store edited data

- This is to store all edited data not to lose due to the power shut down.

  Command ---- | S | A | V | CR |

  Response ----- | T | C | H | > | * | CR |

  Caution : *When data save is completed, the prompt mark (*) is returned. Do not shut down the power while saving data. Otherwise it leads to a memory error alarm and all data must be initialized for recovery.*

TCH mode LOD: Cancel edited data

- This is to cancel all edited data to return to the original state just after the data was saved last time. If the coordinate data, such as the data set for programmed operation, are not stored, all contents of settings will be lost.

  Command ---- | L | O | D | CR |

  Response ----- | T | C | H | > | * | CR |

18.4.2.6. EDT Mode • Second Level Command

EDT mode ***LST: Read out program data***

- This is to report all program data of the specified program number. The output reports the data in the order of name of program, user memo, total number of steps and step data.

```
Command ----  LST 00CR
Response-----  000/NAM $PRGNAMECR
               000/MEM USER_PROG_MEM
               OCR
               000/STE 319CR
               000/000 MOV P0001CR
               000/001 MOV P0002CR
                        ↓
               000/319 ENDCR
               EDT>*CR
```

- To output data of a particular line, add a code or step number after the program number.

```
Command 1 --  LST 00/NAMCR ← Program name.
Response 1 --  000/NAM $PRGNAMECR
               EDT>*CR

Command 2 --  LST 00/MEMCR ← Indication of user memo.
Response 2 --  000/MEM USER_PROG_MEM
               OCR
               EDT>*CR

Command 3 --  LST 00/STECR ← Total number of steps.
Response 3 --  000/STE 319CR
               EDT>*CR

Command 4 --  LST 00/001CR ← Data of specified step.
Response 4 --  000/001 MOV P0001CR
               EDT>*CR
```

- You may input the name of program, instead of the program number, to identify the program.

```
Command ----  LST $PRGNAMECR ← Name of program.
Response-----  039/NAM $PRGNAMECR
                        ↓
               039/319 ENDCR
               EDT>*CR
```

- Input "ALL" instead of the program number to report all program data.

Command ---- | L | S | T | | A | L | L | CR |

Response----- | 0 | 0 | 0 | / | N | A | M | | $ | P | R | G | N | A | M | E | CR |

↓

| 0 | 9 | 9 | / | 0 | 9 | 9 | | E | N | D | CR |

| E | D | T | > | * | CR |

- You may specify the range of step number using (-) (dash).

Command ---- | L | S | T | | 0 | 0 | 0 | / | 0 | 0 | 1 | – | 0 | 0 | 3 | CR |

Response----- | 0 | 0 | 0 | / | 0 | 0 | 1 | | M | O | V | | P | 0 | 0 | 0 | 3 | CR |

| 0 | 0 | 0 | / | 0 | 0 | 2 | | M | O | V | | P | 0 | 0 | 0 | 4 | CR |

| 0 | 0 | 0 | / | 0 | 0 | 3 | | M | O | V | | P | 0 | 0 | 0 | 5 | CR |

| E | D | T | > | * | CR |

- Add "DAT" to the command to confirm available room for storing more programs. The output indicates the allowable number of programs and steps in the program.

Command ---- | L | S | T | | D | A | T | CR |

Response----- | 1 | 2 | 8 | / | 1 | 0 | 0 | 0 | CR |

| E | D | T | > | * | CR |

EDT mode — *SET: Register program data*

- Specify a program number and a step number for registration of the program data.

Command ---- | S | E | T | | 0 | 0 | / | 0 | 9 | 9 | | M | O | V | | P | 0 | 0 | 0 | 3 | CR |

Response----- | E | D | T | > | * | CR |

- When setting a program data to the end step, input "END" instead of step number.

Command ---- | S | E | T | | 0 | 0 | 1 | / | E | N | D | | M | O | V | | P | 0 | 0 | 0 | 3 | CR |

Response----- | E | D | T | > | * | CR |

- When setting the name of program, input a code "NAM" instead of step number.

Command ---- | S | E | T | | 0 | 0 | 1 | / | N | A | M | | $ | P | R | G | N | A | M | E | CR |

Response----- | E | D | T | > | * | CR |

- When setting the user memo, input a code "MEM" instead of step number.

Command ---- | S | E | T | | 0 | 0 | 1 | / | M | E | M | | U | S | E | R | _ | M | E | M | O | CR |

Response----- | E | D | T | > | * | CR |

- When the name of the program has been set, it is possible to use the name instead of the program number to identify the program.

Command ---- | S | E | T | | $ | P | R | G | N | A | M | E | / | 0 | 9 | 9 | | M | O | V | | P |

| 0 | 0 | 0 | 3 | CR |

Response----- | E | D | T | > | * | CR |

EDT mode — CLR: Initialize program data

- This is to initialize all program data and user memo.

  Command ---- | C | L | R | CR |

  Response----- | E | D | T | > | * | CR |

- Add the program number to specify the program.

  Command ---- | C | L | R | | 0 | 1 | CR |

  Response----- | E | D | T | > | * | CR |

- When the name of program has been registered, you may use it to identify the program.

  Command ---- | C | L | R | | $ | P | R | G | N | A | M | E | CR |

  Response----- | E | D | T | > | * | CR |

EDT mode — DEL: Delete program data

- This is to delete the program data of the specified program number and step number. The rest of program steps move up one step.

  Command ---- | D | E | L | | 0 | 0 | 3 | / | 0 | 0 | 9 | CR |

  Response----- | E | D | T | > | * | CR |

- You may specify the range of numbers to be deleted using (-) (dash). The rest of steps move up to the positions of deleted ones.

  Command ---- | D | E | L | | 0 | 0 | 3 | / | 0 | 0 | 9 | – | 0 | 1 | 2 | CR |

  ← Delete step 9 to 12.

  Response----- | E | D | T | > | * | CR |

- When the name of the program has been registered, you may use it to identify the program.

  Command ---- | D | E | L | | $ | P | R | G | N | A | M | E | / | 0 | 0 | 9 | CR |

  Response----- | E | D | T | > | * | CR |

EDT mode — INS: Insert program data

- This is to insert a blank step to the specified step of the specified program. The step to where a blank step is inserted will move down one step.

  Command ---- | I | N | S | | 0 | 0 | 3 | / | 0 | 0 | 9 | CR |

  Response----- | E | D | T | > | * | CR |

- When the name of program has been registered, you may use it to identify the program.

  Command ---- | I | N | S | | $ | P | R | G | N | A | M | E | / | 0 | 0 | 9 | CR |

  Response----- | E | D | T | > | * | CR |

- When inserting multiple blank steps, add a desired quantity to the end of the step number.

  Command ---- | I | N | S | | 0 | 0 | 3 | / | 0 | 0 | 9 | | 0 | 5 | CR |

  Response----- | E | D | T | > | * | CR |

EDT mode CPY: Copy program data

- This is to copy a program data of a specified step number (left) to a specified step number (right). If the copying step number is bigger than the last step, the step data will be put behind it.

Command ---- | C | P | Y | | 0 | 0 | 3 | / | 0 | 0 | 1 | | 0 | 0 | 3 | / | 0 | 0 | 9 | CR |

← Copy step 1 to step 9.

Response----- | E | D | T | > | * | CR |

- You may specify the range of step numbers to be copied using (-) (dash).

Command ---- | C | P | Y | | 0 | 0 | 3 | / | 0 | 0 | 9 | – | 0 | 1 | 2 | | 0 | 0 | 3 | / | 0 |

| 2 | 2 | CR | ← Copy steps 9 to 12 to 22 and over.

Response----- | E | D | T | > | * | CR |

- When the name of program has been registered, you may use it to specify the program number.

Command ---- | C | P | Y | | $ | P | R | G | N | A | M | E | / | 0 | 0 | 1 | | 0 | 3 | / | 0 |

| 0 | 9 | CR |

Response----- | E | D | T | > | * | CR |

EDT mode SAV: Save edited program data

- This is to save all edited program data to prevent from losing due to power shut down.

Command ---- | S | A | V | CR |

Response----- | E | D | T | > | * | CR |

Caution : *The prompt mark (*) is returned when the data saving is completed. Do not shut down the power while saving data. Otherwise it leads to a memory error alarm and all data must be initialized for recovery.*

EDT mode LOD: Delete edited data

- Delete all edited program data to return to the original state just after the data was saved last time.

Command ---- | L | O | D | CR |

Response----- | E | D | T | > | * | CR |

18.4.2.7. PAL Mode • Second Level Command

PAL mode ***LST: Read out palletizing status*** ***[Multi-axis]***

- This is to output the setting data of a specified palletizing pattern number.

  * Refer to "18.4.2.7. PAL Mode • Second Level Command, SET: Set palletizing."

  Command ---- L S T ␣ 0 0 CR

  Response ----- 0 0 / N A M ␣ $ P A L N A M E CR

  0 0 / M E M ␣ U S E R _ P A L _ M E M O CR

  ↓

  0 0 / R E G ␣ I N I ␣ P 0 0 0 1 CR

  P A L > * CR

- Add a code to the end of palletizing pattern number to output only the setting state of a particular one.

  Command ---- L S T ␣ 0 0 / T Y P CR

  Response ----- 0 0 / T Y P ␣ M L T CR

  P A L > * CR

- It is possible to specify a particular palletizing pattern using its palletizing name instead of the number.

  Command ---- L S T ␣ $ P A L N A M E CR

  Response ----- 0 0 / N A M ␣ $ P A L N A M E CR

  ↓

  0 0 / R E G ␣ I N I ␣ P 0 0 0 1 CR

  P A L > * CR

- Add the code "ALL" to the command to indicate the all palletizing pattern data.

  Command ---- L S T ␣ A L L CR

  Response ----- 0 0 / N A M ␣ $ P A L N A M E CR

  ↓

  1 5 / R E G ␣ I N I ␣ P 0 0 0 1 CR

  P A L > * CR

- To check allowable quantity of palletizing patterns for registration, add the code "DAT" to the command.

  Command ---- L S T ␣ D A T CR

  Response ----- 1 6 CR

  P A L > * CR

PAL mode ***SET: Set palletizing pattern data*** ***[Multi-axis]***

- This is to specify a palletizing pattern number and codes, then to register its program data. Add a code "NAM" to the command to set a name of palletizing pattern.

Command ----	S	E	T		0	1	/	N	A	M		$	P	A	L	N	A	M	E	CR
Response-----	P	A	L	>	*	CR														

- Add "MEM" to the command to put user memo to it.

Command ----	S	E	T		0	1	/	M	E	M		U	S	E	R	_	M	E	M	O	CR
Response-----	P	A	L	>	*	CR															

- If the name of palletizing pattern is registered, specify it instead of the number to identify a particular palletizing pattern.

Command ----	S	E	T		$	P	A	L	N	A	M	E	/	S	P	S	~	CR
Response-----	P	A	L	>	*	CR												

- The following show the codes and data format for setting palletizing pattern data.

(1) NAM -----Sets name of palletizing pattern

N	A	M		$	P	A	L	N	A	M	E

(2) MEM-----Sets user memo.

M	E	M		U	S	E	R	_	M	E	M	O

(3) TYP ------Sets pallet positioning format.

T	Y	P		M	L	T	← Multiple regular interval format.
T	Y	P		D	I	V	← Divided sides format.
T	Y	P		P	N	T	← Three corners format.

(4) AXS------Change operating axis unit [normally two axes (X and Y) operation]

A	X	S		X	X		Y	Y

← Two axes palletizing (X and Y axis units operation)

(5) SPS------Set starting point.

(6) WDT -----Sets palletizing position interval.

(7) XPS------Sets corner point X.

(8) YPS------Sets corner point Y.

S	P	S		P	0	0	0	1

S	P	S		X	0	0	0	0	.	0	0		Y	0	0	0	0	.	0	0

(9) NUM -----Sets number of position intervals / number of steps.

N	U	M		P	0	0	0	1

N	U	M		X	0	0		Y	0	0

(10) PTN ----Sets moving direction.

P	T	N		X	N	R	M	← X axis unit takes precedence. Moves to one direction.
P	T	N		X	R	E	V	← X axis unit takes precedence. Moves in both direction.
P	T	N		Y	N	R	M	← Y axis unit takes precedence. Moves to one direction.
P	T	N		Y	R	E	V	← Y axis unit takes precedence. Moves in both direction.

(11) JMP ----Select jump format

J	M	P		N	O	P	← No jump
J	M	P		S	P	S	← Jump (Positioning to start point)
J	M	P		J	M	P	← Jump (No positioning to start point)

(12) REG----Set coordinate data development

R	E	G		N	O	P	← None data development					
R	E	G		I	N	I		P	0	0	0	1
								← Data development (Initialize data register.)				
R	E	G		T	C	H		P	0	0	0	1
								← Data development (without initializing data register)				

* If code "INI" is set, a point register is used after its initialization (calculate again) to execute PALI command. For code "TCH", it is necessary to teach the coordinate data to point registers in advance as they are not initialized (do not calculate again) when executing PALI command.

PAL mode CLR: Initialize palletizing data [Multi-axis]

- Initialize all palletizing data.

Command ----

C	L	R	CR

Response-----

P	A	L	>	*	CR

- When specify a particular palletizing pattern number, add its number to the command.

Command ----

C	L	R		0	1	CR

Response-----

P	A	L	>	*	CR

- When the name is set to a palletizing pattern number, it is possible to use the name instead of the number.

Command ----

C	L	R		$	P	A	L	N	A	M	E	CR

Response-----

P	A	L	>	*	CR

PAL mode CPY: Copy palletizing data [Multi-axis]

- This is to copy palletizing data of a palletizing pattern number (left) to a specified palletizing pattern number (right).

Command ----

C	P	Y		0	3		0	9	CR

← Copy palletizing pattern 3 to palletizing pattern 9.

Response-----

P	A	L	>	*	CR

- When the name is set to a palletizing pattern number, you may use it instead of number to specify the palletizing pattern.

Command ----

C	P	Y		$	P	A	L	N	A	M	E		0	3	CR

Response-----

P	A	L	>	*	CR

PAL mode TCH: Develop palletizing data to point register [Multi-axis]

- Develops all palletizing data to point registers. However, this is only valid when the setting of REG parameters is INI or TCH.

Command ----	T	C	H	CR		
Response -----	P	A	L	>	*	CR

- To specify a palletizing data, add its palletizing pattern number to the command.

Command ----	T	C	H		0	1	CR
Response -----	P	A	L	>	*	CR	

- If the name is set to the palletizing pattern number, you may use it instead of the number for identification.

Command ----	T	C	H		$	P	A	L	N	A	M	E	CR
Response -----	P	A	L	>	*	CR							

- When a palletizing pattern, which is not set to develop palletizing data to a point register, is specified (when REG parameter is NOP), TCH command is meaningless.

PAL mode SAV: Save edited data [Multi-axis]

- This is to save all edited data not to lose them due to power shut down.

Command ----	S	A	V	CR		
Response -----	P	A	L	>	*	CR

Caution : The prompt mark (*) responds when the data saving is completed. During saving data, do not turn the power off. Otherwise, it leads to a memory error alarm and all data must be initialized for recovery.

PAL mode LOD: Cancel edited data [Multi-axis]

- This is to cancel all edited contents to return to the state just after the power is turned on.

Command ----	L	O	D	CR		
Response -----	P	A	L	>	*	CR

18.4.2.8. SYS Mode • Second Level Command

SYS mode ***LST: Report parameter setting***

- This command is to report parameter setting of a motion unit currently specified by UNT command.

Command ---- L S T CR

Response ----- U 1 / L S P _ 0 6 0 0 . 0 CR

U 1 / L A C _ 0 0 . 5 CR

↓

U 1 / C O _ X 3 0 0 0 0 _ Y 3 0 0 0 0 CR

S Y S > * CR

- To check a particular parameter setting, add its code to the command.

Command ---- L S T _ L A C CR

Response ----- U 1 / L A C _ 0 0 . 5 CR

S Y S > * CR

- For checking parameter setting of a particular motion unit, specify its unit number. (multi-axis combination only)

Command ---- L S T _ U 2 CR

Response ----- U 2 / L S P _ 0 6 0 0 . 0 CR

↓

U 2 / C O _ X 3 0 0 0 0 _ Y 3 0 0 0 0 CR

S Y S > * CR

- It is possible to specify a unit number and a code simultaneously. (multi-axis combination only)

Command ---- L S T _ U 2 / L A C CR

Response ----- U 2 / L A C _ 0 0 . 5 CR

S Y S > * CR

- For reporting all parameters of all units, add the code "ALL" to the command. (multi-axis combination only)

Command ---- L S T _ A L L CR

Response ----- U 1 / L S P _ 0 6 0 0 . 0 CR

U 1 / L A C _ 0 0 . 5 CR

↓

U 4 / C O _ X 3 0 0 0 0 _ Y 3 0 0 0 0 CR

S Y S > * CR

SYS mode SET: Set parameter

- This is to set parameters of a motion unit currently specified by UNT command. The parameters in this setting are the same as the parameters described in "9. Initial Setting." Please refer to Chapter 10. Table 18-4 below shows the parameters described in Chapter 10 and their codes used in this setting.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Table 18-4

Initializing parameter	Code	Set to respective units.
Locus speed	LSP	
Locus accel	LAC	
Max speed	MSP	✓
Max accel	MAC	✓
Finish width	MFW	✓
Finish mode	MFM	
Fin control	FCT	
Fin out time	FTI	
Home speed	HSP	
Home accel	HAC	
Home search speed	HSS	
Home direction	HDR	✓
Home sequence	HSQ	✓
Home shift	HSF	✓
Home move	HMV	✓
Home move mode	HMM	
Home unit seq.	HUS	
Jog speed (H)	JSH	
Jog speed (L)	JSL	
Jog accel	JAC	
Overtravel (+)	OTP	✓
Overtravel (-)	OTM	✓
Escape (upr.Z)	EUZ	
Escape (lwr.Z)	ELZ	

Initializing parameter	Code	Set to respective units.
Escape (pos.Z)	EPZ	
Escape (upr.R)	EUR *	
Escape (lwr.R)	ELR *	
Escape (pos.R)	EPR *	
Payload	LO	✓
Servo Gain	SG *	✓
Feedfoward Gain	FF	✓
Torque Limit	TL	✓
Dead Band	DB	✓
Low Pass Filter 1	FP	✓
Low Pass Filter 2	FS	✓
Notch Filter	NP	✓
Gain Mode	– *	✓
Position Loop Gain	PG	✓
Velocity Loop Gain	VG	✓
Observer Gain	DO	✓
Observer Limit	DOL	✓
Notch Filter Q1	NPQ	✓
Position Direction	PD	✓
Hard. OT Timer	HOT	✓
Rated Current	RC	✓
Over Load	OL	✓
Position Error Over	CO	✓

Note: There are several parameters which shall be set to an axis unit respectively. It is indicated in the column of "Set to respective units."

* Reserved: do not set

- For the parameters those are set to each axis units respectively, an axis unit shall be specified to change their setting. Command. The command 1 below is an example that X and Y axis units are specified for a change, while the command 2 specifies Y axis unit only. Only X axis unit shall be specified for a single axis system.

Command 1 -- | S | E | T | | H | S | F | | X | 0 | 0 | 0 | 0 | . | 0 | 0 | | Y | 0 | 0 | 0 |
| 0 | . | 0 | 0 | CR |

Command 2 -- | S | E | T | | H | S | F | | Y | 0 | 0 | 0 | 0 | . | 0 | 0 | CR |

Response ----- | S | Y | S | > | * | CR |

- When changing a parameter of a particular motion unit, specify the unit number in front of a parameter code.

Command ---- | S | E | T | | U | 1 | / | H | S | F | | Y | 0 | 0 | 0 | 0 | . | 0 | 0 | CR |

Response ----- | S | Y | S | > | * | CR |

SYS mode CLR: Initialize parameter

- This is to initialize the parameters for system setting of all motion units.

  Command ---- | C | L | R | CR |

  Response ----- | S | Y | S | > | * | CR |

- When initializing parameters of a particular motion unit only, add its unit number to the command.

  Command ---- | C | L | R | | U | 1 | CR |

  Response ----- | S | Y | S | > | * | CR |

SYS mode SAV: Store edited parameter

- This is to store the edited parameters not to lose due to the power shut down.

  Command ---- | S | A | V | CR |

  Response ----- | S | Y | S | > | * | CR |

  Caution : *The prompt mark (*) responds when saving parameter is completed. Do not turn off the power during parameter saving. Otherwise it leads to memory error alarm and you must initialize all parameters for recovery.*

SYS mode LOD: Cancel edited parameter

- Cancel all edited parameters to return to the state just after the power is turned on.

  Command ---- | L | O | D | CR |

  Response ----- | S | Y | S | > | * | CR |

18.4.2.9. CTR Mode • Second Level Command

CTR mode ***LST: Report setting in controller***

- This is to report the module main unit setting of a motion unit currently specified by UNT command.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

Command ---- L S T CR

Response ----- U 1 / T Y P _ O F F CR

U 1 / M D X _ X Y – H R S 0 5 0 – P H 2 0

0 _ 1 _ X CR

↓

U 1 / M D R CR

C T R > * CR

- To have indications of a particular setting, add its code to the command.

Command ---- L S T _ M D X CR

Response ----- U 1 / M D X _ X Y – H R S 0 5 0 – P H 2 0

0 _ 1 _ X CR

C T R > * CR

- To have the setting state of a particular motion unit, add the unit number to the command.
  (multi-axis combination only)

Command ---- L S T _ U 2 CR

Response ----- U 2 / T Y P _ O F F CR

↓

U 2 / M D R _ X Y – H R S 0 2 0 – P M 2 0

0 _ 4 _ R CR

C T R > * CR

- It is possible to specify both unit number and the code simultaneously.
  (multi-axis combination only)

Command ---- L S T _ U 2 / T Y P CR

Response ----- U 2 / T Y P _ O F F CR

C T R > * CR

- To indicate the settings of all parameters, add the code "ALL" to the command.

Command ---- L S T ␣ A L L CR

Response ----- U 1 / T Y P ␣ O F F CR

U 1 / M D X ␣ X Y – H R S 0 5 0 – P H 2 0

0 ␣ 1 ␣ X CR

↓

U 4 / M D R ␣ X Y – H R S 0 2 0 – P M 2 0

0 ␣ 4 ␣ R CR

U S R 1 ␣ L 1 0 ␣ R 4 0 9 6 ␣ N R M ␣ W 1

0 0 CR

↓

U S R 8 ␣ L 1 0 ␣ R 4 0 9 6 ␣ N R M ␣ W 1

0 0 CR

E M S T ␣ N O P CR

↓

R U N ␣ P R G CR

I B 0 0 0 ␣ E M S T ␣ B CR

↓

I B 1 1 7 ␣ U S E R ␣ A CR

O B 0 0 0 ␣ R D Y ␣ B CR

↓

O B 1 1 7 ␣ U S E R ␣ A CR

C T R > * CR

- To indicate the settings of parameters of all units, add the code "UNT" to the command. (multi-axis combination only)

Command ---- L S T ␣ U N T CR

Response ----- U 1 / T Y P ␣ O F F CR

U 1 / M D X ␣ X Y – H R S 0 5 0 – P H 2 0

0 ␣ 1 ␣ X CR

↓

U 4 / M D R ␣ X Y – H R S 0 2 0 – P M 2 0

0 ␣ 4 ␣ R CR

C T R > * CR

- To have the settings of all user modules, add the code "USR" to the command.

Command ---- L S T ␣ U S R CR

Response ----- U S R 1 ␣ L 1 0 ␣ R 4 0 9 6 ␣ N R M ␣ W 1

0 0 CR

↓

U S R 8 ␣ L 1 0 ␣ R 4 0 9 6 ␣ N R M ␣ W 1

0 0 CR

C T R > ∗ CR

- To have the settings of a particular user module setting, add the code "USR∗" to the command. (∗ is a number of user module.)

Command ---- L S T ␣ U S R 1 CR

Response ----- U S R 1 ␣ L 1 0 ␣ R 4 0 9 6 ␣ N R M ␣ W 1

0 0 CR

C T R > ∗ CR

- To indicate only Input / Output setting state, add the "IOP" to the command.

Command ---- L S T ␣ I O P CR

Response ----- E M S T ␣ N O P CR

↓

R U N ␣ P R O G CR

I B 0 0 0 ␣ E M S T ␣ B CR

↓

I B 1 1 7 ␣ U S E R ␣ A CR

O B 0 0 0 ␣ R D Y ␣ A CR

↓

O B 1 1 7 ␣ U S E R ␣ A CR

C T R > ∗ CR

- To indicate only state of a particular Input / Output setting, add the "EMST" or "IB000" to the command.

Command ---- L S T ␣ I B 0 0 0 CR

Response ----- I B 0 0 0 ␣ E M S T ␣ B CR

C T R > ∗ CR

CTR mode SET: Specify controller setting

- This is to set the type of a module main unit to a motion unit which is currently specified by "UNT" command. Declare the code of module type or data to be set. The parameters to be set here are the same as the parameters that are described in "9. Initial Setting." Refer to Chapter 9 as well.
  (Unit name is omissible as the unit number is always "1" for a single axis system.)

  Command ---- | S | E | T | | T | Y | P | | O | F | F | CR |

  Response----- | C | T | R | > | * | CR |

- To set a type of module main unit of a particular motion unit, add its unit number in front of the code. You cannot specify a unit number for settings of user unit or input / output.

  Command ---- | S | E | T | | U | 1 | / | T | Y | P | | O | F | F | CR |

  Response----- | C | T | R | > | * | CR |

- The following show the code and data format used for setting a type of module main unit.

  (1) TYP ------Setting of module combination number: Set to OFF.
  (The same setting as factory set)

  | T | Y | P | | O | F | F |

  (2) MDX -----Setting X axis module main unit
  (3) MDY -----Setting Y axis module main unit
  (4) MDZ -----Setting Z axis module main unit
  (5) MDR -----Setting R axis module main unit

  Multi-axis combination only.

M	D	X		X	Y	–	H	R	S	0	5	0	–	P	H	2	0	0		1

	X

  * Specify module main unit reference number, connected power amplifier number and name of axis.

  Caution : *Make sure to set the axis units to be used. Incorrect setting leads to malfunction of the robot.*

- Specifying "OFF" to a setting of main unit type will reset to the factory set.

M	D	X		O	F	F		1		X

- If only "C" is specified to main unit type setting, the axis unit will be a "none-motion" axis. In this case, an indication by the command LST does not report the parameters as shown below.

  Command ---- | L | S | T | | M | D | X | CR |

  Response----- | U | 1 | / | M | D | X | CR | (multi-axis combination only)

C	T	R	>	*	CR

* If the same amplifier number is set to multiple axis units, or the same axis name is assigned to different axes in the one unit, the precedence is given in the order of (X → R) axes.

U1／MDX	OFF	1	X ----- Unit 1 X axis (Amplifier 1)
U1／MDY	OFF	2	Y ----- Unit 1 Y axis (Amplifier 2)
U1／MDZ	OFF	3	**X** ----- Invalid
U1／MDR			
U2／MDX	OFF	**1**	X ----- Invalid
U2／MDY	OFF	4	Y ----- Unit 2, Y axis (Amplifier 4)
U2／MDZ			
U2／MDR			

- The code and data format used to set user module are shown below. Adding a unit number to user module is meaningless.

(1) USER1 --User module 1

(2) USER2 --User module 2
↓
(3) USER8 --User module 8

} Multi-axis combination only.

U	S	R	1		L	1	0		R	4	0	9	6		N	R	M		W	1
0	0																			

* Specify ball screw lead, encoder resolution, motor mounting and motor power.

- The code and data format used in Input / Output setting are shown below. Adding unit number to this setting is meaningless.

(1) EMST ---Operation data saving mode at Emergency stop: Do not set other than NOP (factory set).

(2) STOP----Operation data saving mode at Stop: Do not set other than NOP (factory set).

(3) CSTP----Operation data saving mode at cycle stop: Do not set other than NOP (factory set).

(4) ALARM--Operation data saving mode at alarm stop: Do not set other than NOP (factory set).

E	M	S	T		N	O	P

(5) RUN -----Format of operation start command.

R	U	N		P	R	G

(6) IB000----Set state of input port 000.
↓
IB117----Set state of input port 117.
OB000 --Set state of output port 000.
↓
OB117 --Set state of output port 117.

I	B	0	0	0		E	M	S	T		B

* Specify state of the port usage and state of signal contact (open or close).

CTR mode CLR: Initialize controller setting

- This is to initialize controller setting.

  Command ---- C L R CR

  Response ----- C T R > * CR

- The following unit number will be set according to the axis combination of controller when the initialization is executed.

  One axis controller --------U1 (One axis)

  Two axes controller -------U1 (Two axes)

  Three axes controller -----U1 (Three axes)

  Four axes controller-------U1 (Two axes), U2 (Two axes)

- The following show the numbers of power amplifier.

  One axis controller --------U1 (PA1)

  Two axes controller -------U1 (PA1 • PA2)

  Three axes controller -----U1 (PA1 • PA2 • PA3)

  Four axes controller-------U1 (PA1 • PA2), U2 (PA3 • PA4)

- All combination numbers and reference numbers of main unit will be in "none-setting" state.

CTR mode MDL: Report list of module main unit

- This is to report a data list of robot module that can be connected to EXEA controller. The list consists of reference number, type of module main unit in parentheses, stroke and motor power. "USR 1 to 8" are for the data that should be set by user. The data do not contain the information of specifications of main unit such as stroke. ("USR1" only for a single axis system.)

  Command ---- M D L CR

  Response ----- 0 1 U S R 1 CR

  ↓

  2 5 6 X Y – H R S 0 1 0 – R S 1 4 2 (R

  S z 0 1 0 0 2 0 0) CR

  C T R > * CR

- To check only a particular module main unit, specify its list number to the command.

  Command ---- M D L 1 0 CR

  Response ----- 1 0 X Y – H R S 0 7 0 – P H 2 0 0 (P H

  0 7 0 0 2 0 0) CR

  C T R > * CR

- Add the code "DAT" to the command to check how many data is listed.

  Command ---- M D L D A T CR

  Response ----- 7 2 CR (2 4 9 CR for a single axis system)

  C T R > * CR

CTR mode TYL: Report list of module main unit combination (Reserved) [Multi-axis]

- This is to report the data list of module main unit combination. The list consists of combination style, number of axes and type of module main unit in parentheses (PH and PM). When there is not specified a module main unit for corresponding axes, the " - " (dash) is indicated instead.

Command ---- T Y L CR

Response ----- 0 1 U S E R CR

0 2 G H M – 1 (2 P H P M) CR

↓

0 4 D M M – 1 (2 P M P M) CR

C T R > * CR

- To have the report of a particular combination, add a list number to the command.

Command ---- T Y L 0 3 CR

Response ----- 0 3 D H M – 1 (2 P H P M) CR

C T R > * CR

- Add the code "DAT" to the command to confirm how many lists are available.

Command ---- T Y L D A T CR

Response ----- 0 4 CR

C T R > * CR

CTR mode PWL: Report power amplifier list

- This is to report the data list of the power amplifiers incorporated with EXEA controller. The data list consists of motor type and its power output. The data are indicated for each amplifier numbers.

Command ---- P W L CR

Response ----- 0 1 A C 2 0 0 A CR

0 2 A C 1 0 0 A CR

C T R > * CR

- To have a data of particular power amplifier, add its list number to the command.

Command ---- P W L 2 CR

Response ----- 0 2 A C 1 0 0 A CR

C T R > * CR

- To check how many data is listed, add code "DAT."

Command ----- P W L D A T CR

Response ----- 0 2 CR

C T R > * CR

CTR mode SAV: Save edited data

- This is to store all edited data not to lose them due to unexpected power shut down.

Command ---- | S | A | V | CR |

Response ----- | C | T | R | > | * | CR |

Caution : When saving data completes the prompt (*) is returned. Do not shut off the power while saving the data. Otherwise it leads to a memory error and the all data must be initialized for recovery.

CTR mode LOD: Cancel edited data

- Cancels all edited data to get back to the state just after the power is tuned on.

Command ---- | L | O | D | CR |

Response ----- | C | T | R | > | * | CR |

18.4.2.10. FNC Mode • Second Level Command

FUN mode ***INI: Initialize controller***

- Reset the controller to the factory set. All stored data will be deleted.

Command ---- | I | N | I | CR |

Response ----- | F | N | C | > | * | CR |

(Blank Page)

World-wide Manufacturing and Marketing Organization

NSK Ltd. INTERNATIONAL DIVISION
JAPAN: Tokyo Phone: 03-3779-7120

NSK CORPORATION
U.S.A.: Ann Arbor Phone: 313-761-9500

[Precision Products Business Unit]
U.S.A. : Chicago Phone: 630-924-8000
: Los Angeles Phone: 562-926-3578
: Ann Arbor Phone: 761-761-9500

NSK•RHP CANADA INC.
CANADA : Toront Phone: 905-890-0740
: Montreal Phone: 514-633-1240
: Vancouver Phone: 800-663-5445

NSK RODAMIENTOS MEXICANA, S.A. DE C.V.
MEXICO: Mexico City Phone: 5-301-2741,5-301-3115

NSK DO BRASIL INDUSTRIA E COMÉRCIO DE ROLAMENTOS LTDA.
BRASIL : São Paulo Phone: 001-269-4700
: Porto Alegre Phone: 051-222-1324
: Belo Horizonte Phone: 031-224-2508

NSK-RHP UK LTD
ENGLAND : Ruddington Phone: 0115-936-6600

NSK-RHP DEUTSCHLAND G.m.b.H
GERMANY : Düsseldorf Phone: 02102-4810
: Stuttgart Phone: 0711-79082-0
: Leipzig Phone: 0341-5631241

NSK-RHP FRANCE S.A.
FRANCE : Paris Phone:1.30.57.39.39
: Lyon Phone: 72.15.29.00

NSK-RHP NEDERLAND B.V.
NETHERLAND: Amsterdam Phone: 020-6470711

NSK-RHP ITALIA S.p.A.
ITALIA: Milano Phone: 02-995191

NSK-RHP IBERICA, S.A.
SPAIN: Barcelona Phone: 93-575-1662

NSK-RHP AUSTRALIA PTY, LTD.
AUSTRALIA : Melbourne Phone: 03-9764-8302
: Sydney Phone: 02-9893-8322
: Brisbane Phone: 07-3393-1388
: Adelaide Phone: 08-8373-4811
: Perth Phone: 089-434-1311

NSK-RHP BEARINGS NEW ZEALAND LTD.
NEW ZEALAND: Auckland Phone: 09-276-4992

NSK KOREA CO., LTD.
KOREA: Seoul Phone: 02-3287-6001

NSK SINGAPORE (PRIVATE) LTD.
SINGAPORE: Singapore Phone: 2781711

NSK BEARINGS (THAILAND) CO., LTD.
THAILAND : Bangkok Phone: 2-6412150-60
: Chiang mai Phone: 053-246993~4

TAIWAN NSK PRECISION CO., LTD.
TAIWAN: Taipei Phone: 02-591-0656

Robot Module System

- **P Series Module Main Unit**
- **R Series Module Main Unit**
- **EXEA Controller**

User's Manual 2

= Programming and Operation of EXEA Controller =

Document Number: K20079-01 EC-T

August 3, 2000 1st Edition 1st Printing

NSK Ltd.

NSK

1st Edition, 1st Printing August 3, 2000 Document Number: K20079-01