FANUC

GE Fanuc Automation

PowerMotion™ Products

Power Mate H Motion Controller

Operator's Manual

GFZ-62684EN/02 January 1996

GFL-001

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

SAFETY PRECAUTIONS

This section describes the safety precautions related to the use of CNC units. It is essential that these precautions be observed by users to ensure the safe operation of machines equipped with a CNC unit (all descriptions in this section assume this configuration). Note that some precautions are related only to specific functions, and thus may not be applicable to certain CNC units.
Users must also observe the safety precautions related to the machine, as described in the relevant manual supplied by the machine tool builder. Before attempting to operate the machine or create a program to control the operation of the machine, the operator must become fully familiar with the contents of this manual and relevant manual supplied by the machine tool builder.

Contents

1 DEFINITION OF WARNING, CAUTION, AND NOTE

This manual includes safety precautions for protecting the user and preventing damage to the machine. Precautions are classified into Warning and Caution according to their bearing on safety. Also, supplementary information is described as a Note. Read the Warning, Caution, and Note thoroughly before attempting to use the machine.

WARNING

Applied when there is a danger of the user being injured or when there is a damage of both the user being injured and the equipment being damaged if the approved procedure is not observed.

CAUTION

Applied when there is a danger of the equipment being damaged, if the approved procedure is not observed.

NOTE

The Note is used to indicate supplementary information other than Warning and Caution.

- **Read this manual carefully, and store it in a safe place.**

2 GENERAL WARNINGS AND CAUTIONS

WARNING

1. Never attempt to machine a workpiece without first checking the operation of the machine. Before starting a production run, ensure that the machine is operating correctly by performing a trial run using, for example, the single block, feedrate override, or machine lock function or by operating the machine with neither a tool nor workpiece mounted. Failure to confirm the correct operation of the machine may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

2. Before operating the machine, thoroughly check the entered data.
Operating the machine with incorrectly specified data may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

3. Ensure that the specified feedrate is appropriate for the intended operation. Generally, for each machine, there is a maximum allowable feedrate. The appropriate feedrate varies with the intended operation. Refer to the manual provided with the machine to determine the maximum allowable feedrate. If a machine is run at other than the correct speed, it may behave unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

4. When using a tool compensation function, thoroughly check the direction and amount of compensation.
Operating the machine with incorrectly specified data may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

5. The parameters for the CNC and PMC are factory–set. Usually, there is not need to change them. When, however, there is not alternative other than to change a parameter, ensure that you fully understand the function of the parameter before making any change.
Failure to set a parameter correctly may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

6. Immediately after switching on the power, do not touch any of the keys on the MDI panel until the position display or alarm screen appears on the CNC unit.
Some of the keys on the MDI panel are dedicated to maintenance or other special operations. Pressing any of these keys may place the CNC unit in other than its normal state. Starting the machine in this state may cause it to behave unexpectedly.

7. The operator’s manual and programming manual supplied with a CNC unit provide an overall description of the machine’s functions, including any optional functions. Note that the optional functions will vary from one machine model to another. Therefore, some functions described in the manuals may not actually be available for a particular model. Check the specification of the machine if in doubt.

WARNING

8. Some functions may have been implemented at the request of the machine–tool builder. When using such functions, refer to the manual supplied by the machine–tool builder for details of their use and any related cautions.

NOTE

Programs, parameters, and macro variables are stored in nonvolatile memory in the CNC unit. Usually, they are retained even if the power is turned off. Such data may be deleted inadvertently, however, or it may prove necessary to delete all data from nonvolatile memory as part of error recovery.
To guard against the occurrence of the above, and assure quick restoration of deleted data, backup all vital data, and keep the backup copy in a safe place.

3 WARNINGS AND CAUTIONS RELATED TO PROGRAMMING

This section covers the major safety precautions related to programming. Before attempting to perform programming, read the supplied operator's manual and programming manual carefully such that you are fully familiar with their contents.

WARNING

1. Coordinate system setting

If a coordinate system is established incorrectly, the machine may behave unexpectedly as a result of the program issuing an otherwise valid move command.
Such an unexpected operation may damage the tool, the machine itself, the workpiece, or cause injury to the user.

2. Positioning by nonlinear interpolation

When performing positioning by nonlinear interpolation (positioning by nonlinear movement between the start and end points), the tool path must be carefully confirmed before performing programming.
Positioning involves rapid traverse. If the tool collides with the workpiece, it may damage the tool, the machine itself, the workpiece, or cause injury to the user.

3. Function involving a rotation axis

When programming polar coordinate interpolation or normal–direction (perpendicular) control, pay careful attention to the speed of the rotation axis. Incorrect programming may result in the rotation axis speed becoming excessively high, such that centrifugal force causes the chuck to lose its grip on the workpiece if the latter is not mounted securely.
Such mishap is likely to damage the tool, the machine itself, the workpiece, or cause injury to the user.

4. Inch/metric conversion

Switching between inch and metric inputs does not convert the measurement units of data such as the workpiece origin offset, parameter, and current position. Before starting the machine, therefore, determine which measurement units are being used. Attempting to perform an operation with invalid data specified may damage the tool, the machine itself, the workpiece, or cause injury to the user.

5. Constant surface speed control

When an axis subject to constant surface speed control approaches the origin of the workpiece coordinate system, the spindle speed may become excessively high. Therefore, it is necessary to specify a maximum allowable speed. Specifying the maximum allowable speed incorrectly may damage the tool, the machine itself, the workpiece, or cause injury to the user.

WARNING

6. Stroke check

After switching on the power, perform a manual reference position return as required. Stroke check is not possible before manual reference position return is performed. Note that when stroke check is disabled, an alarm is not issued even if a stroke limit is exceeded, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the user.

7. Tool post interference check

A tool post interference check is performed based on the tool data specified during automatic operation. If the tool specification does not match the tool actually being used, the interference check cannot be made correctly, possibly damaging the tool or the machine itself, or causing injury to the user.
After switching on the power, or after selecting a tool post manually, always start automatic operation and specify the tool number of the tool to be used.

8. Absolute/incremental mode

If a program created with absolute values is run in incremental mode, or vice versa, the machine may behave unexpectedly.

9. Plane selection

If an incorrect plane is specified for circular interpolation, helical interpolation, or a canned cycle, the machine may behave unexpectedly. Refer to the descriptions of the respective functions for details.

10. Torque limit skip

Before attempting a torque limit skip, apply the torque limit. If a torque limit skip is specified without the torque limit actually being applied, a move command will be executed without performing a skip.

11. Programmable mirror image

Note that programmed operations vary considerably when a programmable mirror image is enabled.

12. Compensation function

If a command based on the machine coordinate system or a reference position return command is issued in compensation function mode, compensation is temporarily canceled, resulting in the unexpected behavior of the machine.
Before issuing any of the above commands, therefore, always cancel compensation function mode.

4 WARNINGS AND CAUTIONS RELATED TO HANDLING

This section presents safety precautions related to the handling of machine tools. Before attempting to operate your machine, read the supplied operator's manual and programming manual carefully, such that you are fully familiar with their contents.

WARNING

1. Manual operation

When operating the machine manually, determine the current position of the tool and workpiece, and ensure that the movement axis, direction, and feedrate have been specified correctly. Incorrect operation of the machine may damage the tool, the machine itself, the workpiece, or cause injury to the operator.

2. Manual reference position return

After switching on the power, perform manual reference position return as required. If the machine is operated without first performing manual reference position return, it may behave unexpectedly. Stroke check is not possible before manual reference position return is performed. An unexpected operation of the machine may damage the tool, the machine itself, the workpiece, or cause injury to the user.

3. Manual numeric command

When issuing a manual numeric command, determine the current position of the tool and workpiece, and ensure that the movement axis, direction, and command have been specified correctly, and that the entered values are valid.
Attempting to operate the machine with an invalid command specified may damage the tool, the machine itself, the workpiece, or cause injury to the operator.

4. Manual handle feed

In manual handle feed, rotating the handle with a large scale factor, such as 100, applied causes the tool and table to move rapidly. Careless handling may damage the tool and/or machine, or cause injury to the user.

5. Disabled override

If override is disabled (according to the specification in a macro variable) during threading, rigid tapping, or other tapping, the speed cannot be predicted, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the operator.

6. Origin/preset operation

Basically, never attempt an origin/preset operation when the machine is operating under the control of a program. Otherwise, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the tool, or causing injury to the user.

WARNING

7. Workpiece coordinate system shift

Manual intervention, machine lock, or mirror imaging may shift the workpiece coordinate system. Before attempting to operate the machine under the control of a program, confirm the coordinate system carefully.
If the machine is operated under the control of a program without making allowances for any shift in the workpiece coordinate system, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the operator.

8. Software operator's panel and menu switches

Using the software operator's panel and menu switches, in combination with the MDI panel, it is possible to specify operations not supported by the machine operator's panel, such as mode change, override value change, and jog feed commands.
Note, however, that if the MDI panel keys are operated inadvertently, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the user.

9. Manual intervention

If manual intervention is performed during programmed operation of the machine, the tool path may vary when the machine is restarted. Before restarting the machine after manual intervention, therefore, confirm the settings of the manual absolute switches, parameters, and absolute/incremental command mode.

10. Feed hold, override, and single block

The feed hold, feedrate override, and single block functions can be disabled using custom macro system variable #3004. Be careful when operating the machine in this case.

11. Dry run

Usually, a dry run is used to confirm the operation of the machine. During a dry run, the machine operates at dry run speed, which differs from the corresponding programmed feedrate. Note that the dry run speed may sometimes be higher than the programmed feed rate.

12. Cutter and tool nose radius compensation in MDI mode

Pay careful attention to a tool path specified by a command in MDI mode, because cutter or tool nose radius compensation is not applied. When a command is entered from the MDI to interrupt in automatic operation in cutter or tool nose radius compensation mode, pay particular attention to the tool path when automatic operation is subsequently resumed. Refer to the descriptions of the corresponding functions for details.

13. Program editing

If the machine is stopped, after which the machining program is edited (modification, insertion, or deletion), the machine may behave unexpectedly if machining is resumed under the control of that program. Basically, do not modify, insert, or delete commands from a machining program while it is in use.

5 WARNINGS RELATED TO DAILY MAINTENANCE

WARNING

1. Memory backup battery replacement

When replacing the memory backup batteries, keep the power to the machine (CNC) turned on, and apply an emergency stop to the machine. Because this work is performed with the power on and the cabinet open, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing the batteries, be careful not to touch the high–voltage circuits (marked ⚠ and fitted with an insulating cover).
Touching the uncovered high–voltage circuits presents an extremely dangerous electric shock hazard.

NOTE

The CNC uses batteries to preserve the contents of its memory, because it must retain data such as programs, offsets, and parameters even while external power is not applied.
If the battery voltage drops, a low battery voltage alarm is displayed on the machine operator's panel or CRT screen.
When a low battery voltage alarm is displayed, replace the batteries within a week. Otherwise, the contents of the CNC's memory will be lost.
Refer to the maintenance section of the operator's manual or programming manual for details of the battery replacement procedure.

WARNING

2. Absolute pulse coder battery replacement

When replacing the memory backup batteries, keep the power to the machine (CNC) turned on, and apply an emergency stop to the machine. Because this work is performed with the power on and the cabinet open, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing the batteries, be careful not to touch the high–voltage circuits (marked ⚠ and fitted with an insulating cover).
Touching the uncovered high–voltage circuits presents an extremely dangerous electric shock hazard.

NOTE

The absolute pulse coder uses batteries to preserve its absolute position.
If the battery voltage drops, a low battery voltage alarm is displayed on the machine operator's panel or CRT screen.
When a low battery voltage alarm is displayed, replace the batteries within a week. Otherwise, the absolute position data held by the pulse coder will be lost.
Refer to the maintenance section of the operator's manual or programming manual for details of the battery replacement procedure.

WARNING

3. Fuse replacement

For some units, the chapter covering daily maintenance in the operator’s manual or programming manual describes the fuse replacement procedure.
Before replacing a blown fuse, however, it is necessary to locate and remove the cause of the blown fuse.
For this reason, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing a fuse with the cabinet open, be careful not to touch the high–voltage circuits (marked ⚠ and fitted with an insulating cover).
Touching an uncovered high–voltage circuit presents an extremely dangerous electric shock hazard.

Table of Contents

III OPERATION

IV. MAINTENANCE

APEENDIX

I GENERAL

1 GENERAL

About this manual

This manual consists of the following parts:

I. GENERAL
Describes chapter organization, applicable models, related manuals, and notes for reading this manual.

II. PROGRAMMING
Describes each function: Format used to program functions in the NC language, characteristics, and restrictions.

III. OPERATION
Describes the manual operation and automatic operation of a machine, procedures for inputting and outputting data, and procedures for editing a program.

IV. MAINTENANCE
Describes alarms, self–diagnosis, and procedures for replacing fuses and batteries.

V. APPENDIX
Lists tape codes, valid data ranges, and error codes.

This manual does not describe parameters in detail. For details on parameters mentioned in this manual, refer to the Connection Manual (B–62683EN).

This manual describes all optional functions. Look up the options incorporated into your system in the manual written by the machine tool builder.

Applicable models

The models covered by this manual, and their abbreviations are:

Product name	Abbreviations	
Power Mate–MODEL H	Power Mate–H	Power Mate

Special symbols

This manual uses the following symbols:

IP_ : Indicates a combination of axes such as X__ Y__ Z (used in PROGRAMMING.).

; : Indicates the end of a block. It actually corresponds to the ISO code LF or EIA code CR.

Related manuals

The table below lists manuals related to the FANUC Power Mate–MODEL H. In the table, this manual is marked with an asterisk (*).

Table 1 Manuals Related to the FANUC Power Mate–MODEL H

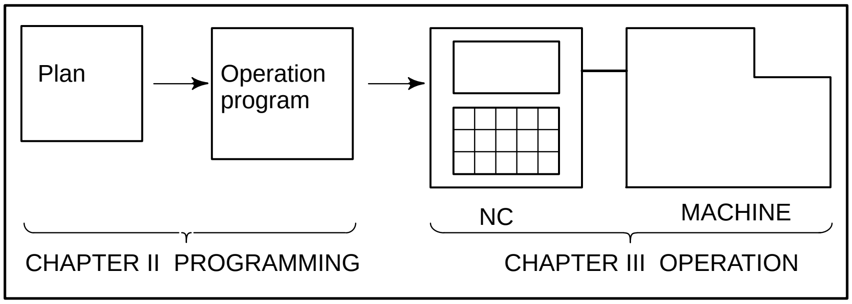
Manual name	Specification numbeNr	
DESCRIPTIONS	B–62682EN	
CONNECTION MANUAL	B–62683EN	
OPERATOR'S MANUAL	B–62684EN	*
MAINTENANCE MANUAL	B–62685EN	

1.1 GENERAL PROCEDURE FOR OPERATING MACHINE EQUIPPED WITH NC

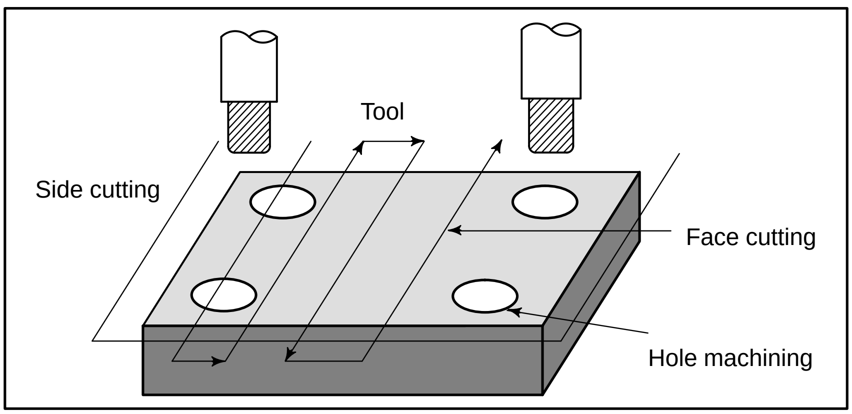
When operating a machine equipped with an NC, you must first create a program then operate the machine according to that program.

1) First, prepare the program from a operation plan to operate the NC machine tool.
   How to prepare the program is described in the Chapter II. PROGRAMMING.

2) The program is to be read into the NC system. Operate the tools according to the programming. Finally, execute the machining actually.
   How to operate the CNC system is described in the Chapter III. OPERATION.



Plan how the machine is to be operated before attempting the actual programming.
Operation plan
1. Determination of machine operation range
2. Method of mounting workpieces on the machine tool
3. Operation sequence in every process
4. Control condition of peripheral devies
Decide the operating method in every process.



Prepare the program of the tool path and feedrate according to the workpiece figure, for each process.

1.2 NOTES ON READING THIS MANUAL

NOTE

1 The function of a machine system depends not only on the NC, but on the combination of the machine tool, its magnetic cabinet, the servo system, the NC, the operator's panels, etc. It is too difficult to describe the function, programming, and operation relating to all combinations. This manual generally describes these from the stand–point of the NC. So, for details on a particular machine, refer to the manual issued by the machine tool builder, which should take precedence over this manual.

2 Headings are placed in the left margin so that the reader can easily access necessary information. When locating the necessary information, the reader can save time by searching though these headings.

3 Programs, parameters, variables, etc. are stored in the NC unit internal non–volatile memory. In general, these contents are not lost by the switching ON/OFF of the power. However, it is possible that a state can occur where precious data stored in the non–volatile memory has to be deleted, because of deletions from a maloperation, or by a failure restoration. In order to restore rapidly when this kind of mishap occurs, it is recommended that you create a copy of the various kinds of data beforehand.

4 This manual describes as many reasonable variations in equipment usage as possible. It cannot address every combination of features, options and commands that should not be attempted.
If a particular combination of operations is not described, it should not be attempted.

II PROGRAMMING

1 GENERAL

1.1 TOOL MOVEMENT ALONG WORKPIECE PARTS FIGURE– INTERPOLATION

The tool moves along straight lines constituting the workpiece parts figure (See II–4).

Explanations

The function of moving the tool along straight lines is called the interpolation.

- **Tool movement along a straight line**

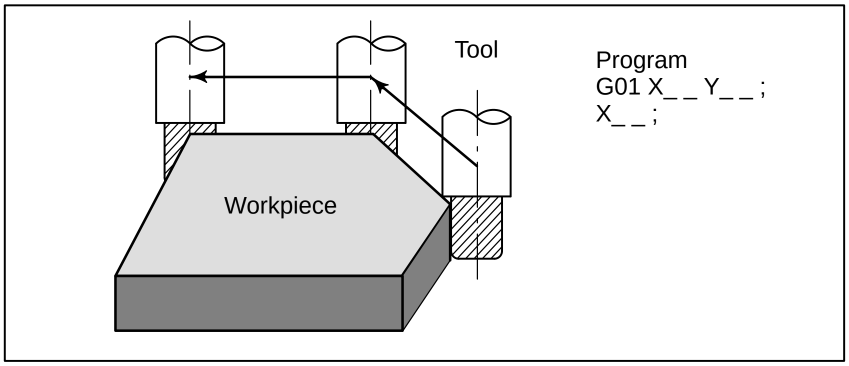


Fig.1.1 (a) Tool movement along a straight line

Symbols of the programmed commands G01, ... are called the preparatory function and specify the type of interpolation conducted in the control unit.

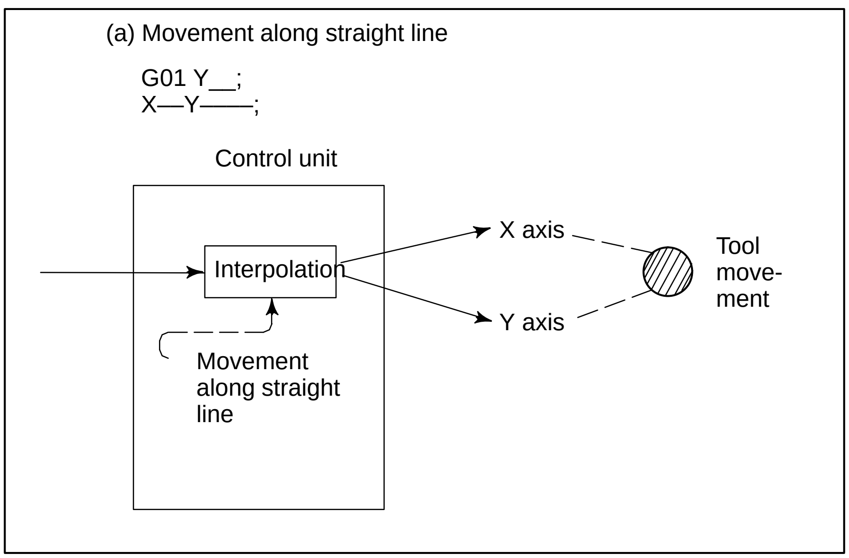


Fig. 1.1 (b) Interpolation function

1.2 FEED–FEED FUNCTION

Movement of the tool at a specified speed for machining a workpiece is called the feed.

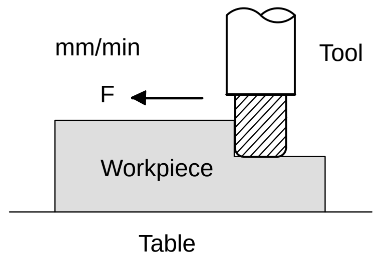


Fig. 1.2 (a) Feed function

Feedrates can be specified by using actual numerics. For example, to feed the tool at a rate of 150 mm/min, specify the following in the program:
F150.0
The function of deciding the feed rate is called the feed function (See II–5).

1.3 PART DRAWING AND TOOL MOVEMENT

1.3.1 Reference Position (Machine–Specific Position)

A NC machine is usually provided with a fixed position. Attachment change and programming of absolute zero point as described later are performed at this position. This position is called the reference position.

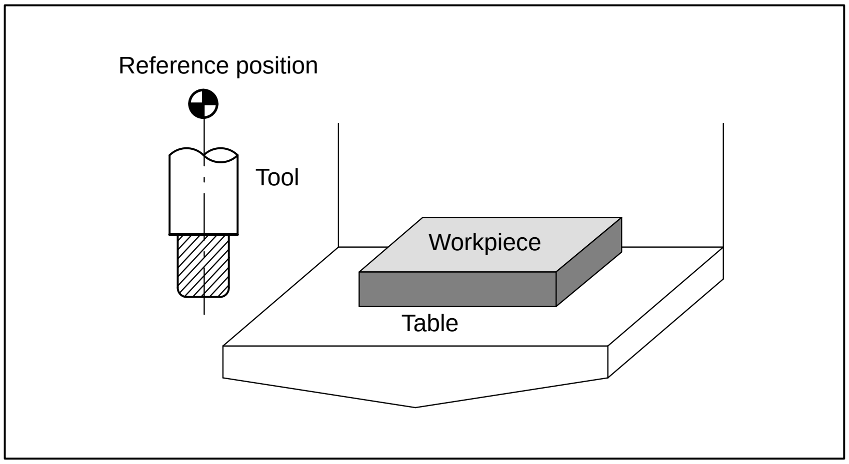


Fig. 1.3 (a) Reference position

Explanations

The tool can be moved to the reference position in two ways:

(1) Manual reference position return (See III–3.1)
Reference position return is performed by manual button operation.

(2) Automatic reference position return (See II–6)
In general, manual reference position return is performed first after the power is turned on. In order to move the tool to the reference position for attachment change or etc thereafter, the function of automatic reference position return is used.

1.3.2 Coordinate System on Drawing and Coordinate System Specified by NC – Coordinate System

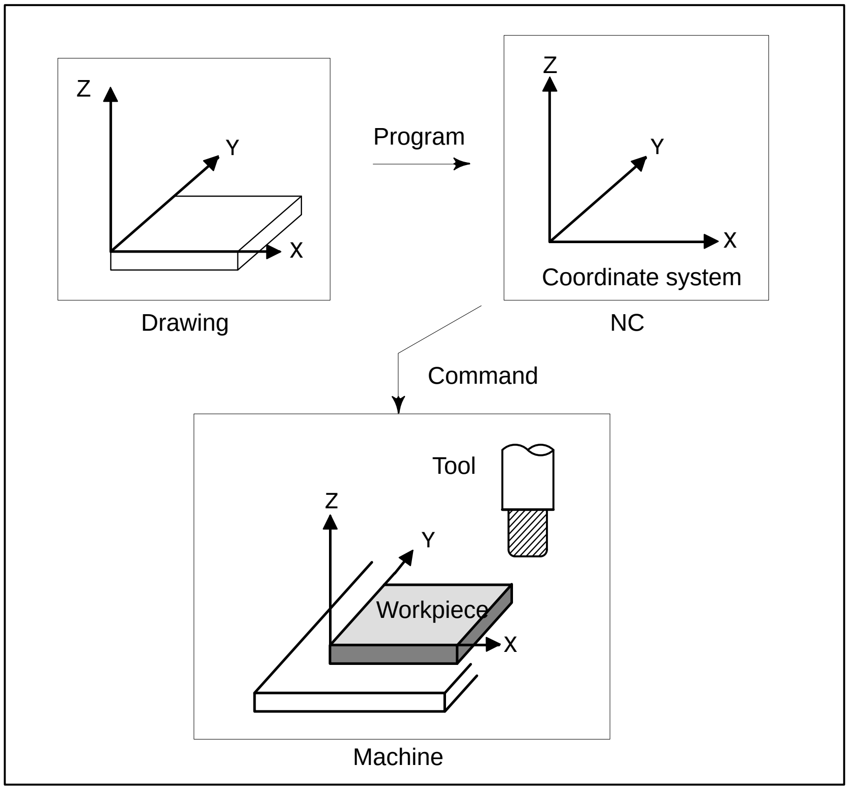


Fig. 1.3.2 (a) Coordinate system

Explanations

- **Coordinate system**

The following two coordinate systems are specified at different locations: (See II–7)

(1) Coordinate system on drawing
The coordinate system is written on the drawing. As the program data, the coordinate values on this coordinate system are used.

(2) Coordinate system specified by the NC
The coordinate system is prepared on the actual machine table. This can be achieved by programming the distance from the current position of the tool to the zero point of the coordinate system to be set.

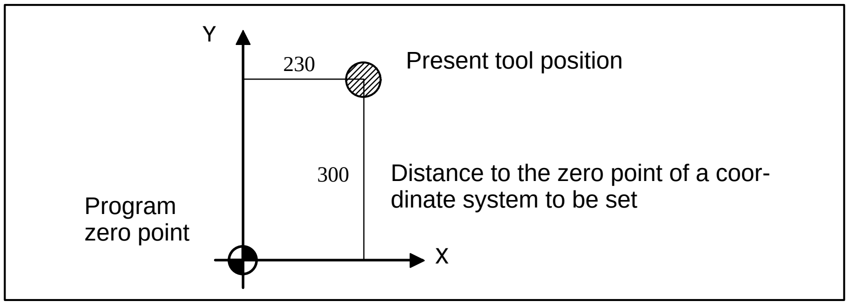


Fig. 1.3.2 (b) Coordinate system specified by the NC

The positional relation between these two coordinate systems is determined when a workpiece is set on the table.

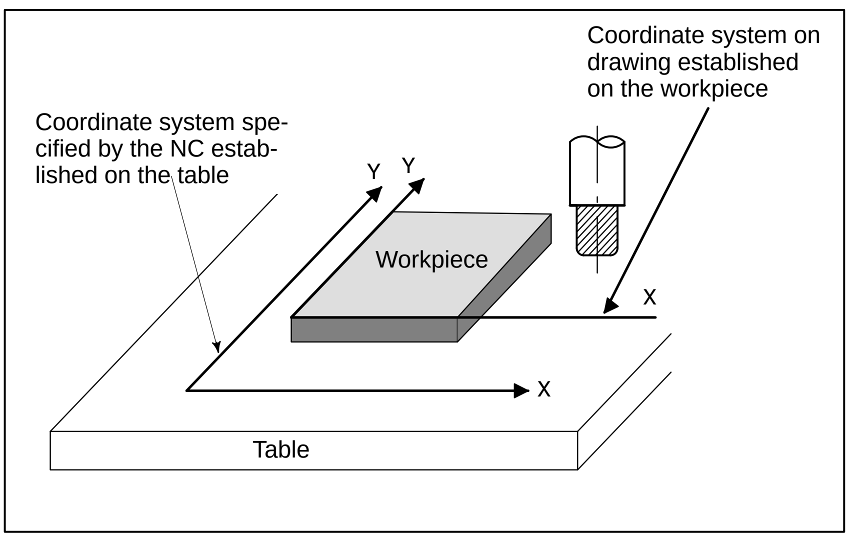


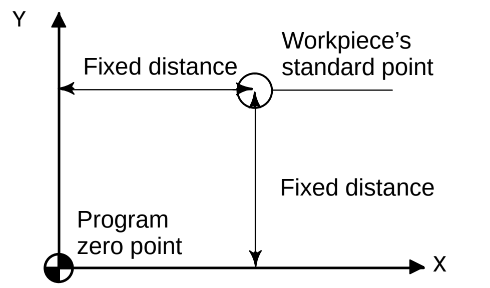
Fig. 1.3.2 (c) Coordinate system specified by NC and coordinate systemon drawing

The tool moves on the coordinate system specified by the NC in accordance with the command program generated with respect to the coordinate system on the drawing, and cuts a workpiece into a shape on the drawing.
Therefore, in order to correctly cut the workpiece as specified on the drawing, the two coordinate systems must be set at the same position.

- **Methods of setting the two coordinate systems in the same position**

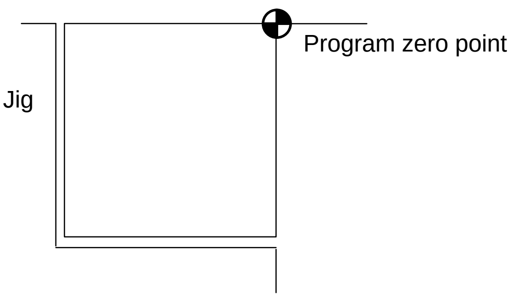
To set the two coordinate systems at the same position, simple methods shall be used according to workpiece shape, the number of machinings.

(1) Using a standard plane and point of the workpiece.



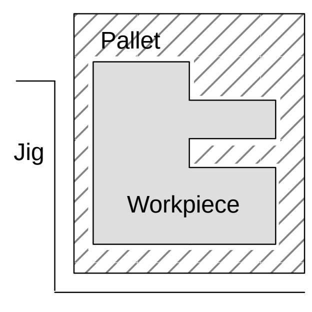
Bring the tool center to the workpiece standard point.
And set the coordinate system specified by NC at this position.

(2) Mounting a workpiece directly against the jig



Meet the tool center to the reference position. And set the coordinate system specified by NC at this position. (Jig shall be mounted on the predetermined point from the reference position.)

(3) Mounting a workpiece on a pallet, then mounting the workpiece and pallet on the jig



(Jig and coordinate system shall be specified by the same as (2)).

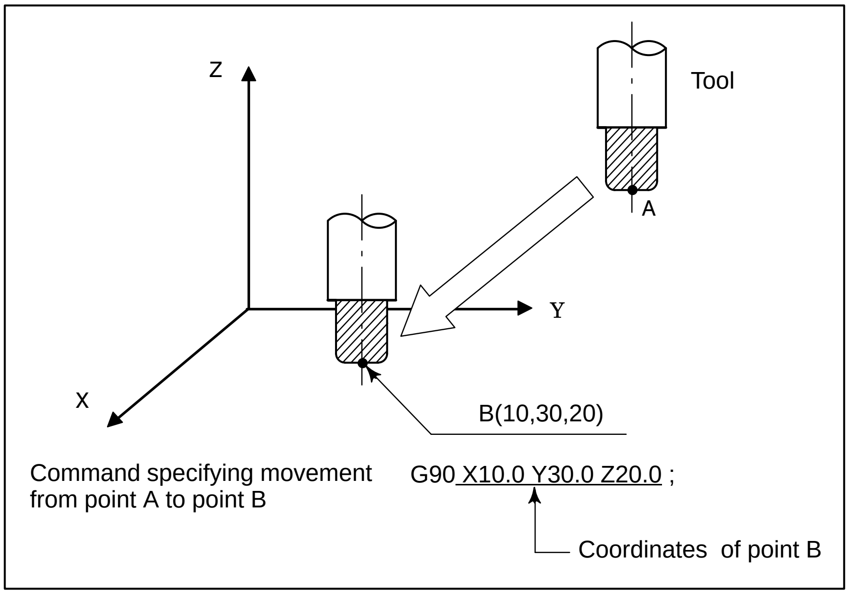
1.3.3 How to Indicate Command Dimensions for Moving the Tool – Absolute, Incremental Commands

Explanations

Coordinate values of command for moving the tool can be indicated by absolute or incremental designation (See II–9.1).

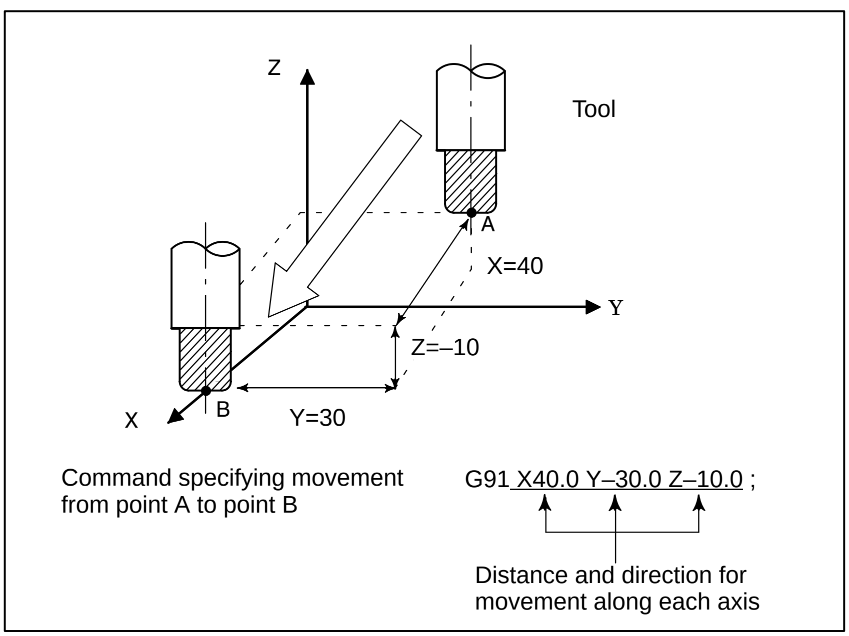
- **Absolute coordinates**

The tool moves to a point at ”the distance from zero point of the coordinate system” that is to the position of the coordinate values.



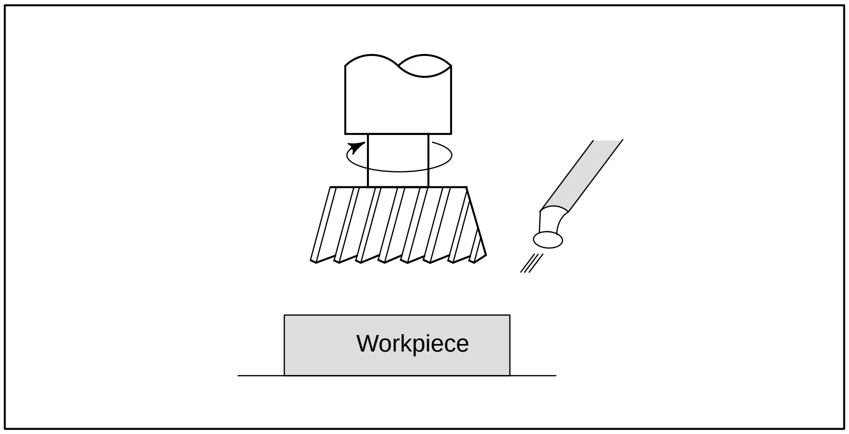
- **Incremental coordinates**

Specify the distance from the previous tool position to the next tool position.



1.4 COMMAND FOR MACHINE OPERATIONS – MISCELLANEOUS FUNCTION

When machining is actually started, it is necessary to rotate the spindle, and feed coolant. For this purpose, on–off operations of spindle motor and coolant valve should be controlled (See II–12).



The function of specifying the on–off operations of the components of the machine is called the miscellaneous function. In general, the function is specified by an M code.

1.5 PROGRAM CONFIGURATION

A group of commands given to the NC for operating the machine is called the program. By specifying the commands, the tool is moved along a straight line, or the spindle motor is turned on and off.
In the program, specify the commands in the sequence of actual tool movements.

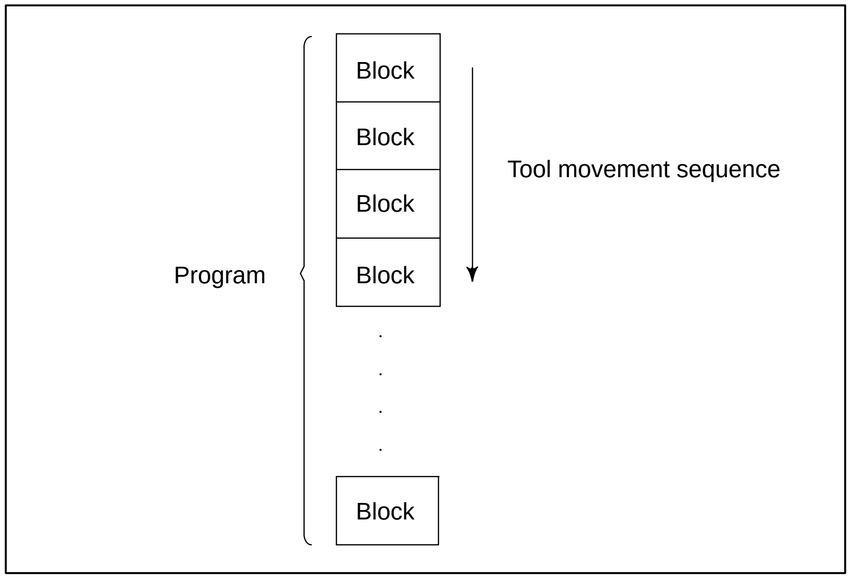


Fig.1.5(a) Program configuration

A group of commands at each step of the sequence is called the block. The program consists of a group of blocks for a series of machining. The number for discriminating each block is called the sequence number, and the number for discriminating each program is called the program number (See II–10).

Explanations

The block and the program have the following configurations.

- **Block**

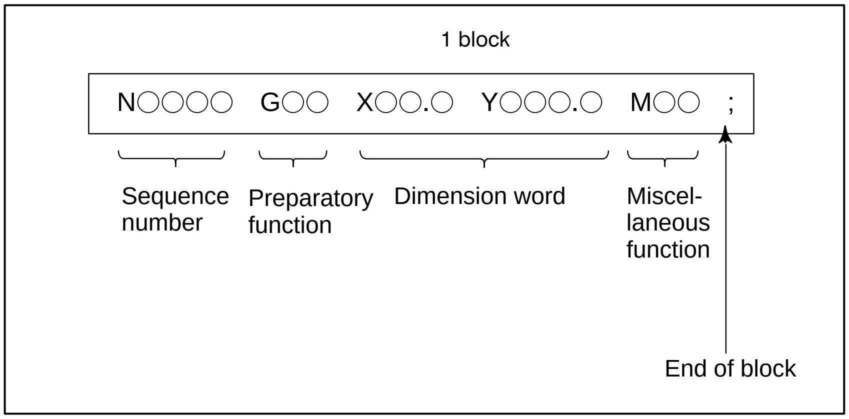


Fig. 1.5 (b) Block configuration

Each block contains a sequence number for indicating the NC operation sequence at the beginning of the block, and an end–of–block code for indicating the end of the block.
This manual indicates the end–of–block code by ; (LF in the ISO code and CR in the EIA code).

- **Program**

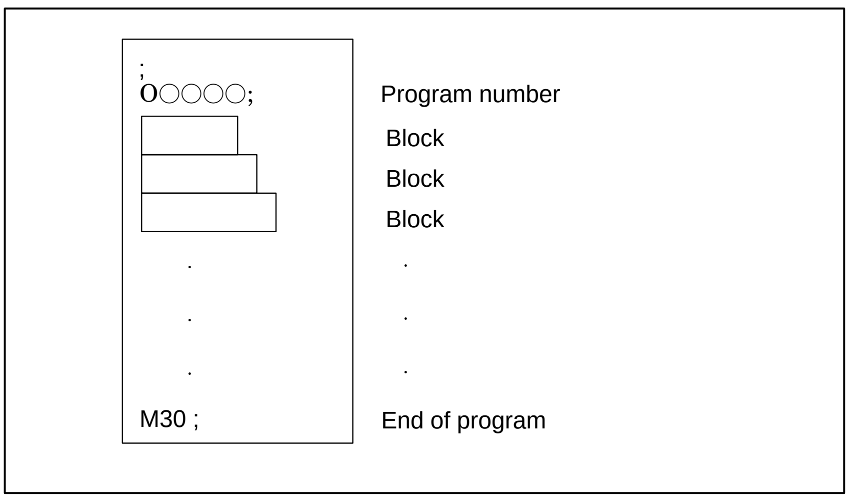
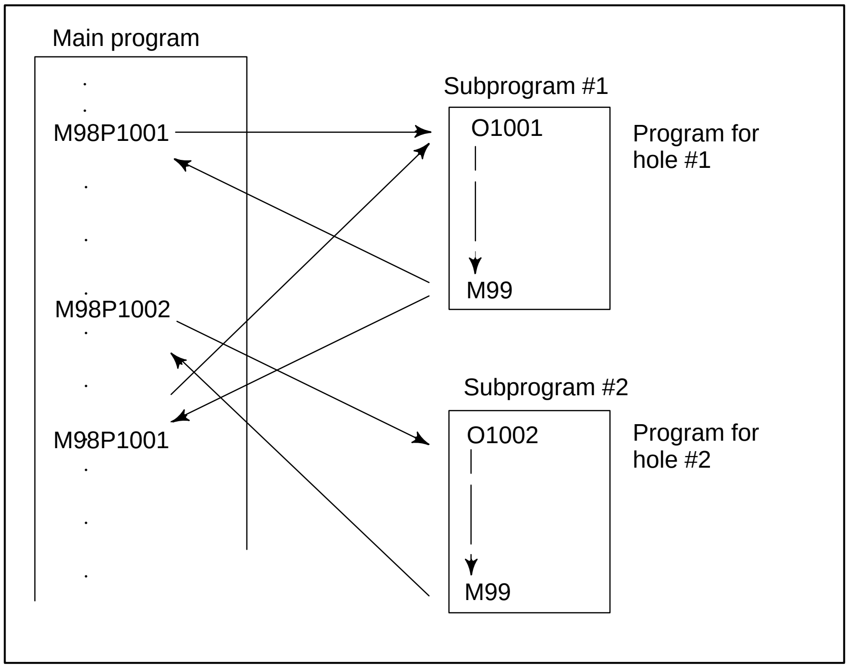


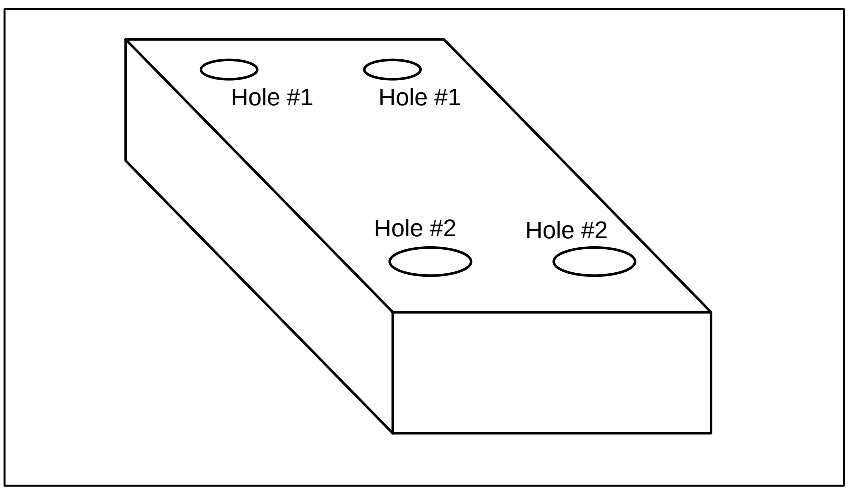
Fig. 1.5 (c) Program configuration

Normally, a program number is specified after the end–of–block (;) code at the beginning of the program, and a program end code (M02 or M30) is specified at the end of the program.

- **Main program and subprogram**

When machining of the same pattern appears at many portions of a program, a program for the pattern is created. This is called the subprogram. On the other hand, the original program is called the main program. When a subprogram execution command appears during execution of the main program, commands of the subprogram are executed. When execution of the subprogram is finished, the sequence returns to the main program.



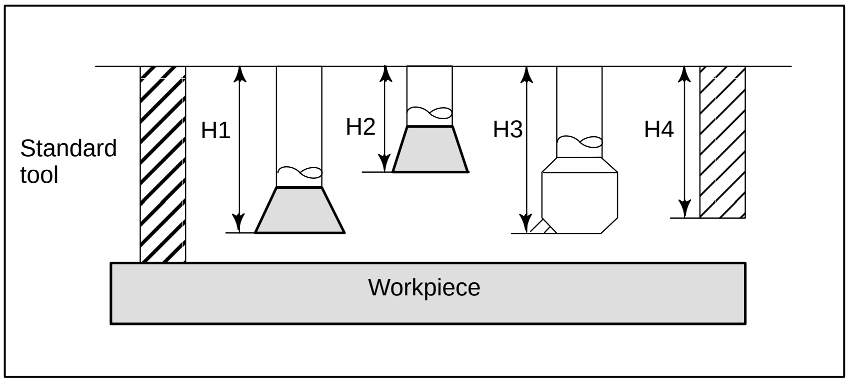


1.6 TOOL FIGURE AND TOOL MOTION BY PROGRAM

Explanations

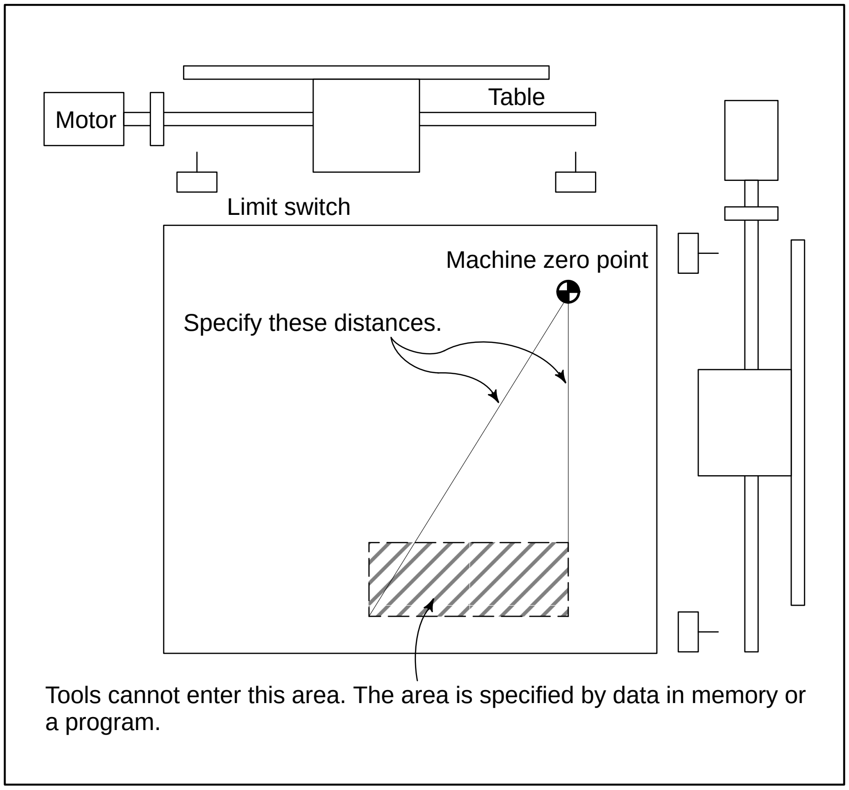
- **Machining using the end of cutter – Tool length compensation function (See II–11.1)**

Usually, several tools are used for machining one workpiece. The tools have different tool length. It is very troublesome to change the program in accordance with the tools.
Therefore, the length of each tool used should be measured in advance. By setting the difference between the length of the standard tool and the length of each tool in the CNC (data display and setting : see III–11), machining can be performed without altering the program even when the tool is changed. This function is called tool length compensation.



1.7 TOOL MOVEMENT RANGE – STROKE

Limit switches are installed at the ends of each axis on the machine to prevent tools from moving beyond the ends. The range in which tools can move is called the stroke. Besides the stroke limits, data in memory can be used to define an area which tools cannot enter.



Besides strokes defined with limit switches, the operator can define an area which the tool cannot enter using a program or data in memory (see Section III–11). This function is called stroke check.

2 CONTROLLED AXES

2.1 CONTROLLED AXES

No. of basic controlled axes	1 axis
Controlled axes expansion	Max. 5 axes (Max. 6 axes in total)
Basic simultaneously controlled axes	1 axis
Simultaneously controlled axes expansion	Max. 3 axes each path

NOTE
The number of simultaneously controllable axes for manual operation jog feed, manual reference position return, or manual rapid traverse) is 1 or 3 (1 when bit 0 (JAX) of parameter 1002 is set to 0 and 3 when it is set to 1).

2.2 AXIS NAME

The user can assign any one of the following nine characters as the axis name: A, B, C, U, V, W, X, Y, and Z.Parameter No. 1020 is used to determine the name of each axis.
When this parameter is set to 0 or a character other than the valid characters is specified, an axis name from 1 to 6 is assigned by default.

Limitations

- **Default axis name**

When a default axis name (1 to 6) is used, operation in the AUTO mode and MDI mode is disabled.

- **Duplicate axis names**

If a duplicate axis name is specified in the parameter, operation is enabled only for the axis specified first.

2.3 INCREMENT SYSTEM

Name of increment system	Least input increment	Least command increment	Maximum stroke
IS–A	0.01mm 0.001inch 0.01deg	0.01mm 0.001inch 0.01deg	999999.99mm 99999.999inch 999999.99deg
IS–B	0.001mm 0.0001inch 0.001deg	0.001mm 0.0001inch 0.001deg	99999.999mm 9999.9999inch 99999.999deg

Combined use of the inch system and the metric system is not allowed. There are functions that cannot be used between axes with different unit systems. For the increment system, see the machine tool builder's manual.

2.4 MAXIMUM STROKE

Maximum stroke = Least command increment $\times$ 99999999
See 2.3 Incremen System.

NOTE
1 A command exceeding the maximum stroke cannot be specified.
2 The actual stroke depends on the machine tool.

3 PREPARATORY FUNCTION (G FUNCTION)

A number following address G determines the meaning of the command for the concerned block.
G codes are divided into the following two types.

Type	Meaning
One–shot G code	The G code is effective only in the block in which it is specified.
Modal G code	The G code is effective until another G code of the same group is specified.

(Example)
G01 and G00 are modal G codes in group 01.

```
G01X – ;  ⎫
   Z  – ;  ⎬ G01 is effective in this range.
   X  – ;  ⎭
G00Z–;
```

Explanations

1. When the clear state (bit 6 (CLR) of parameter No. 3402) is set at power–up or reset, the modal G codes are placed in the states described below.
   (1) The modal G codes are placed in the states marked with ☆ as indicated in Table 3.
   (2) G20 and G21 remain unchanged when the clear state is set at power–up or reset.(3)
   (3) The user can select G00 or G01 by setting bit 0 (G01) of parameter No. 3402.
   (4) The user can select G90 or G91 by setting bit 3 (G91) of parameter No. 3402.
   (5) The user can select G17, G18, or G19 by setting bit 1 (G18) and bit 1 (G19) of parameter No. 3402.
2. G codes other than G10 and G11 are one–shot G codes.
3. When a G code not listed in the G code list is specified, or a G code that has no corresponding option is specified, alarm No. 010 is output.
4. Multiple G codes can be specified in the same block if each G code belongs to a different group. If multiple G codes that belong to the same group are specified in the same block, only the last G code specified is valid.
5. G codes are indicated by group.

Table 3 G code list

G code	Group	Function	
G00 ☆	01	Positioning	
G01 ☆		Linear interpolation	
G04	00	Dwell, Exact stop	
G10		Data setting	
G11		Data setting mode cancel	
G17 ☆	02	XpYp plane selection	Xp: X axis or its parallel axis
G18		ZpXp plane selection	Yp: Y axis or its parallel axis
G19		YpZp plane selection	Zp: Z axis or its parallel axis
G20	06	Input in inch	
G21		Input in mm	
G27	00	Reference position return check	
G28		Return to reference position	
G29		Return from reference position	
G30		2nd and 3rd reference position return	
G31		Skip function	
G43	08	Tool length compensation + direction	
G44		Tool length compensation – direction	
G49 ☆		Tool length compensation cancel	
G65	00	Macro call	
G66	12	Macro modal call	
G67 ☆		Macro modal call cancel	
G90 ☆	03	Absolute command	
G91 ☆		Increment command	
G92	00	Setting for work coordinate system	
G93	05	Rate feed	
G94 ☆		Feed per minute	
G95		Feed per rotation	

4 INTERPOLATION FUNCTIONS

4.1 POSITIONING (G00)

The G00 command moves a tool to the position in the workpiece system specified with an absolute or an incremental command at a rapid traverse rate.
In the absolute command, coordinate value of the end point is programmed.
In the incremental command the distance the tool moves is programmed.

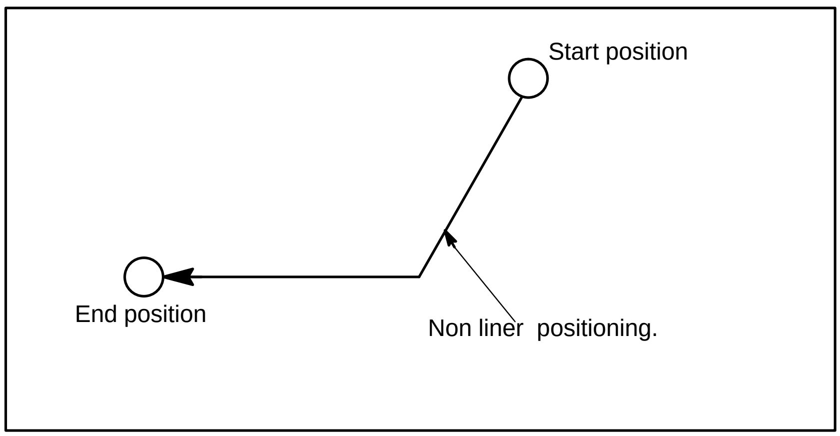
Format

G00IP_;

IP_: For an absolute command, the coordinates of an end position, and for an incremental commnad, the distance the tool moves.

Explanations

Tool path generally does not become a straight line.



The rapid traverse rate in the G00 command is set to the parameter No.1420 for each axis independently by the machine tool builder. In the positioning mode actuated by G00, the tool is accelerated to a predetermined speed at the start of a block and is decelerated at the end of a block. Execution proceeds to the next block after confirming the in–position.
”In–position” means that the feed motor is within the specified range.
This range is determined by the machine tool builder by setting to parameter No.1827.

Restrictions

The rapid traverse rate cannot be specified in the address F.

4.2 LINEAR INTERPOLATION (G01)

Tools can move along a line

Format

G01 IP_F_;

IP_:For an absolute command, the coordinates of an end point , and for an incremental commnad, the distance the tool moves.

F_:Speed of tool feed (Feedrate)

Explanations

A tools move along a line to the specified position at the feedrate specified in F.
The feedrate specified in F is effective until a new value is specified. It need not be specified for each block.
The feedrate commanded by the F code is measured along the tool path. If the F code is not commanded, the feedrate is regarded as zero.
The feedrate of each axis direction is as follows.

G01 $\alpha\underline{\alpha}$ $\beta\underline{\beta}$ $\gamma\underline{\gamma}$ $\zeta\underline{\zeta}$ F$\underline{f}$;

Feed rate of α axis direction : $F\alpha = \frac{\alpha}{L} \times f$

Feed rate of B axis direction : $F_\beta = \frac{\beta}{L} \times f$

Feed rate of Γ axis direction : $F\gamma = \frac{\gamma}{L} \times f$

Feed rate of Z axis direction : $F_\zeta = \frac{\zeta}{L} \times f$

$L = \sqrt{\alpha^2 + \beta^2 + \gamma^2 + \zeta^2}$

The feed rate of the rotary axis is commanded in the unit of deg/min (the unit is decimal point position).
When the straight line axis α(such as X, Y, or Z) and the rotating axisβ (such as A, B, or C) are linearly interpolated, the feed rate is that in which the tangential feed rate in the αandβ cartesian coordinate system is commanded by F(mm/min).
β–axis feedrate is obtained ; at first, the time required for distribution is calculated by using the above fromula, then the β –axis feedrate unit is changed to deg /min.

A calculation example is as follows.
G91 G01 X20.0B40.0 F300.0 ;
This changes the unit of the C axis from 40.0 deg to 40mm with metric input. The time required for distribution is calculated as follows:

$$\frac{\sqrt{20^2 + 40^2}}{300} \doteq 0.14907 \text{ (min)}$$

The feed rate for the C axis is

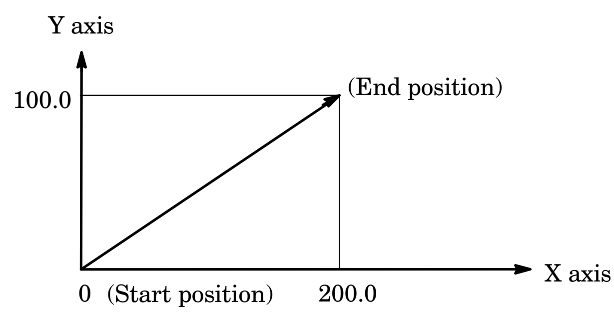
$$\frac{40}{0.14907} \doteq 268.3 \text{ deg/min}$$

In simultaneous 3 axes control, the feed rate is calculated the same way as in 2 axes control.

Examples

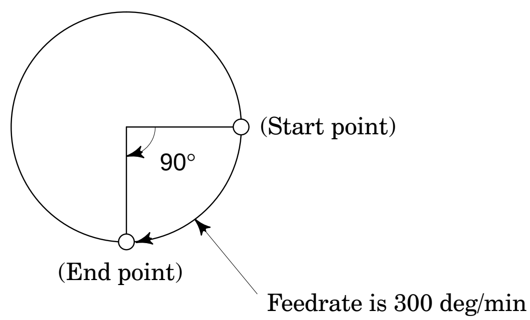
- **Linear interpolation**

(G91) G01X200.0Y100.0F200.0 ;



- **Feedrate for the rotation axis**

G91G01C–90.0 F300.0 ;Feed rate of 300deg/min



4.3 SKIP FUNCTION(G31)

Linear interpolation can be commanded by specifying axial move following the G31 command, like G01. If an external skip signal is input during the execution of this command, execution of the command is interrupted and the next block is executed.
The skip function is used when the end of moving is not programmed but specified with a signal from the machine, for example. It is used also for measuring the dimensions of a workpiece.

Format

G31 **IP_** ;

G31**:** One–shot G code (If is effective only in the block in which it is specified)

Explanations

The coordinate values when the skip signal is turned on can be used in a custom macro because they are stored in the custom macro system variable #5061 to #5066, as follows:

#5061 X axis coordinate value
#5062 Y axis coordinate value
#5063 Z axis coordinate value
#5064 4th axis coordinate value
#5065 5th axis coordinate value
#5066 6th axis coordinate value

WARNING
Disable feedrate override, dry run, and automatic acceleration /deceleration (with parameter No. 6200 and subsequent parameters) when the feedrate per minute is specified, allowing for an error in the position of the tool when a skip signal is input. These functions are enabled when the feedrate per rotation is specified.

Examples

- **The next block to G31 is an incremental command**

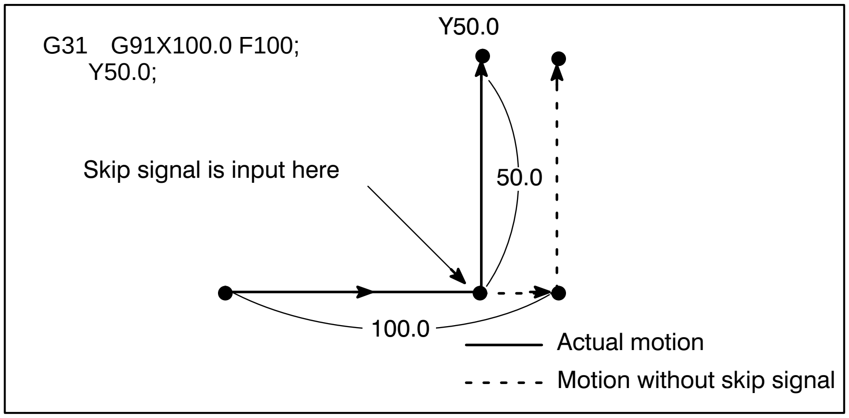


Fig.4.3 (a) The next block is an incremental command

- **The next block to G31 is an absolute command for 1 axis**

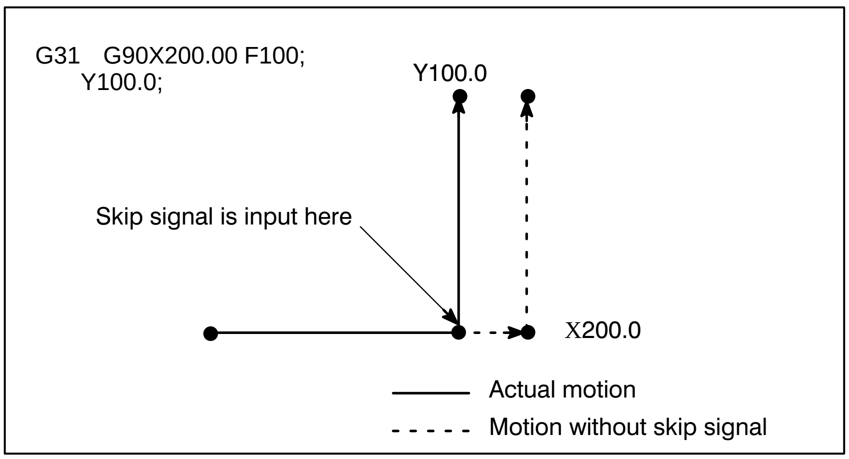


Fig.4.3 (b) The next block is an absolute command for 1 axis

- **The next block to G31 is an absolute command for 2 axes**

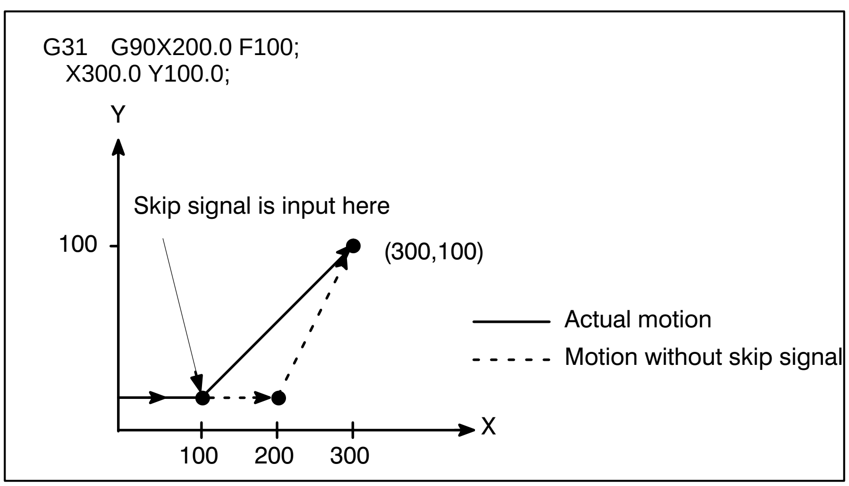


Fig. 4.3 (c) The next block is an absolute command for 2 axes

5 FEED FUNCTIONS

5.1 GENERAL

The feed functions control the feedrate of the tool. The following two feed functions are available:

• Feed functions

1. Rapid traverse
   When the positioning command (G00) is specified, the tool moves at a rapid traverse feedrate set in the CNC (parameter No. 1420).
2. Feed at programmed rate
   The tool moves at a programmed rate.

• Override

Override can be applied to a rapid traverse rate or programmed feedrate using the switch on the machine operator's panel.

• Automatic acceleration/ deceleration

To prevent a mechanical shock, acceleration/deceleration is automatically applied when the tool starts and ends its movement (Fig. 5.1 (a)).

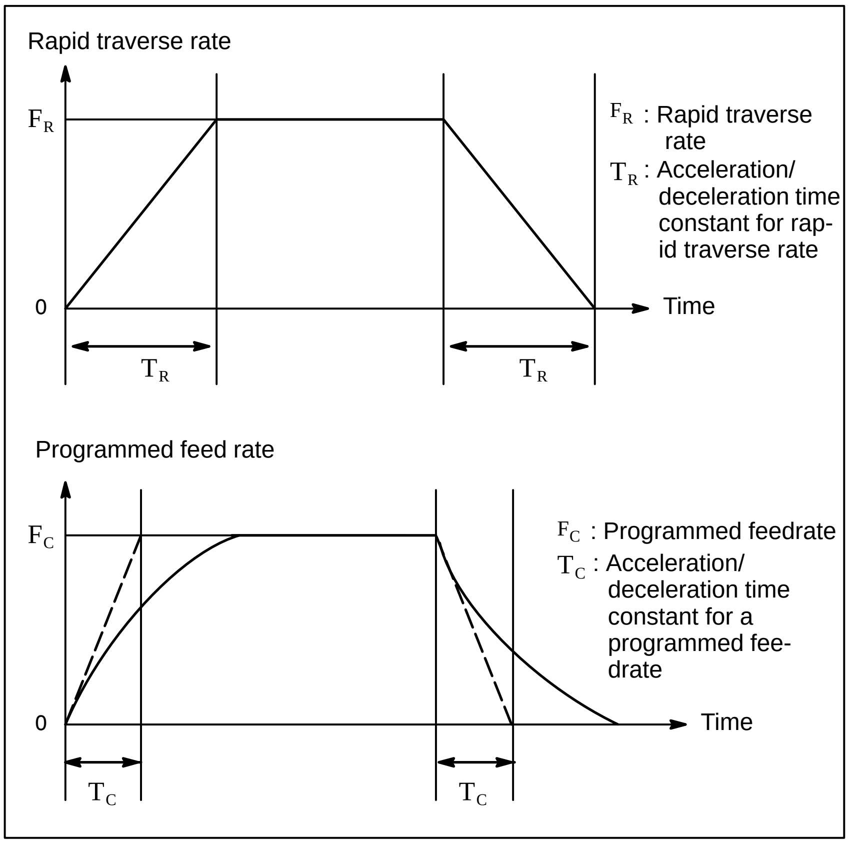


Fig. 5.1 (a) Automatic acceleration/deceleration (example)

- **Tool path in a feed at programmed rate**

If the direction of movement changes between specified blocks during feed at programmed rate, a rounded–corner path may result (Fig. 5.1 (b)).

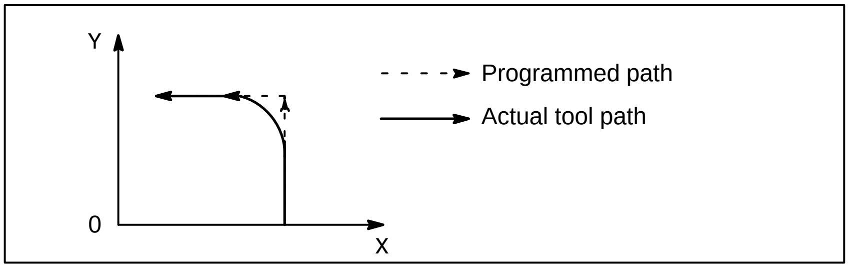


Fig. 5.1 (b) Example of Tool Path between Two Blocks

5.2 RAPID TRAVERSE

Format

G00 **IP_** ;

G00 : G code (group 01) for positioning (rapid traverse)
IP_ ; Dimension word for the end point

Explanations

The positioning command (G00) positions the tool by rapid traverse. In rapid traverse, the next block is executed after the specified feedrate becomes 0 and the servo motor reaches a certain range set by the machine tool builder (in–position check).

A rapid traverse rate is set for each axis by parameter No. 1420, so no rapid traverse feedrate need be programmed.

The following overrides can be applied to a rapid traverse rate with the switch on the machine operator's panel:F0, 25, 50, 100%

F0: Allows a fixed feedrate to be set for each axis by parameter No. 1421.

For detailed information, refer to the appropriate manual of the machine tool builder.

5.3 FEED AT PROGRAMMED RATE

Feedrate of linear interpolation (G01), etc. are commanded with numbers after the F code.
In feed at programmed rate, the next block is executed so that the feedrate change from the previous block is minimized.
Two modes of specification are available:

1. Feed per minute (G94)
   After F, specify the amount of feed of the tool per minute.
2. Feed per revolution (G95)
   After F, specify the amount of feed of the tool per position coder revolution.

Format

Feed per minute

G94 ;	**G code (group 05) for feed per minute**
F_ ;	**Feedrate command (mm/min or inch/min)**

Feed per revolution

G95 ;	**G code (group 05) for feed per revolution**
F_ ;	**Feedrate command (mm/rev or inch/rev)**

Explanations

• Tangential speed constant control

Feed at programmed rate is controlled so that the tangential feedrate is always set at a specified feedrate.

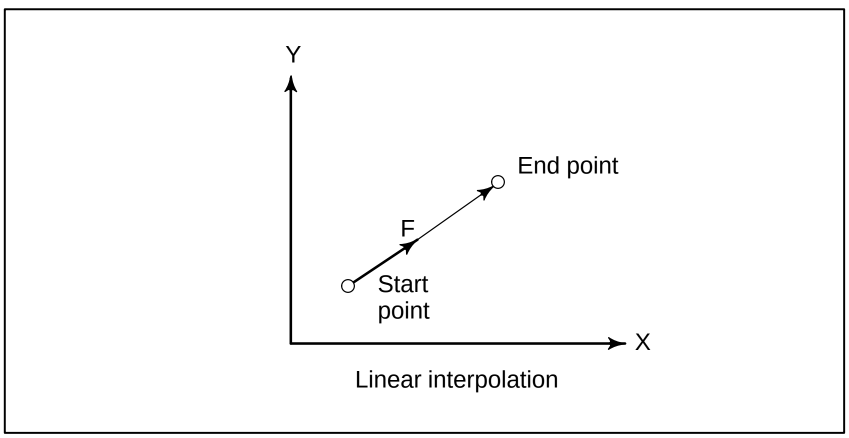


Fig. 5.3 (a) Tangential feedrate (F)

• Feed per minute (G94)

After specifying G94 (in the feed per minute mode), the amount of feed of the tool per minute is to be directly specified by setting a number after F. G94 is a modal code. Once a G94 is specified, it is valid until G95 (feed per revolution) is specified. At power–on, the feed per minute mode is set.
An override from 0% to 254% (in 1% steps) can be applied to feed per minute with the switch on the machine operator’s panel. For detailed information, see the appropriate manual of the machine tool builder.

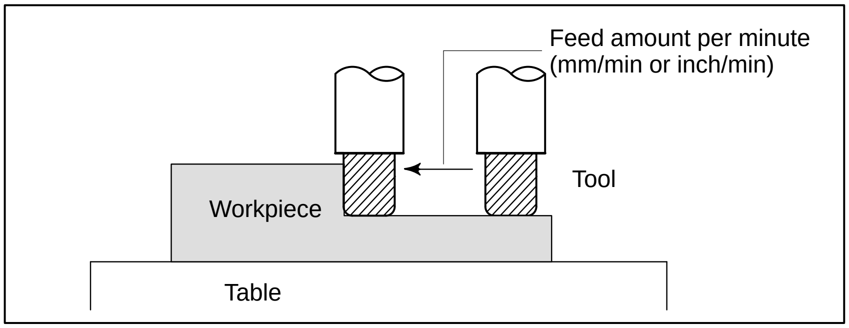


Fig. 5.3 (b) Feed per minute

WARNING
No override can be used for some commands.

- **Feed per revolution (G95)**

After specifying G95 (in the feed per revolution mode), the amount of feed of the tool per position coder revolution is to be directly specified by setting a number after F. G95 is a modal code. Once a G95 is specified, it is valid until G94 (feed per minute) is specified.
An override from 0% to 254% (in 1% steps) can be applied to feed per revolution with the switch on the machine operator's panel. For detailed information, see the appropriate manual of the machine tool builder.

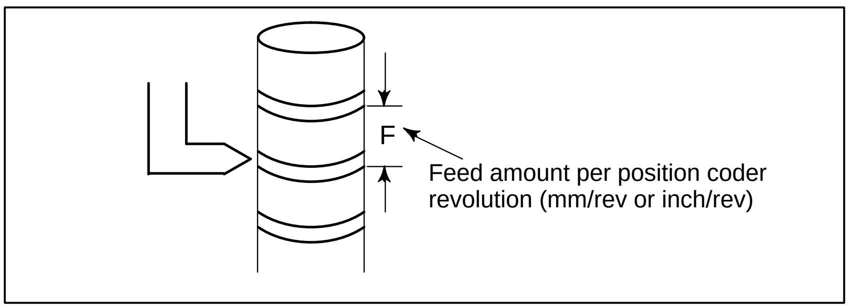


Fig. 5.3 (c) Feed per revolution

CAUTION
When the speed of the position coder is low, feedrate fluctuation may occur. The slower the spindle rotates, the more frequently feedrate fluctuation occurs.

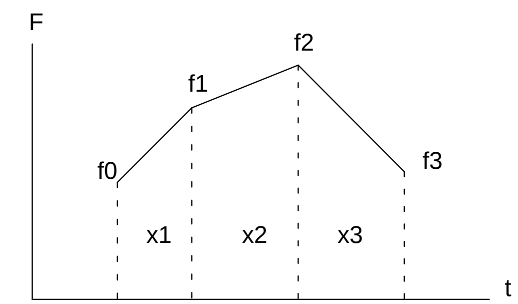
- **Programmed feedrate clamp**

A common upper limit can be set on the programmed feedrate along each axis with parameter No. 1422. If an actual programmed feedrate (with an override applied) exceeds a specified upper limit, it is clamped to the upper limit.

NOTE
An upper limit is set in mm/min or inch/min. CNC calculation may involve a feedrate error of ±2% with respect to a specified value. However, this is not true for acceleration/deceleration. To be more specific, this error is calculated with respect to a measurement on the time the tool takes to move 500 mm or more during the steady state:

5.4 RATE FEED (G93)

Specify the rate feed mode by G93, and specify the tool's final velocity directly by the numeric value following F. Taking the value of F of the preceding block as initial speed, accelerate or decelerate at a certain ratio. Specify the unit of the value of F by mm/min or inch/min. Once G93 is specified modal, it is valid unitl G94 (per minute feed) or G95 (per revolution dwell) is specified.




```
N10  G01  G93     F f0   ;
N20       X x1  F f1   ;
N30       X x2  F f2   ;
N40       X x3  F f3   ;
```

WARNING

1 The upper limit of acceleration speed is clamped by the parameter FEDMX (No. 1422) for the upper limit of normal feed at programmed rate.

2 Set the lower limit of deceleration speed by a parameter RFDMN (No. 1480). When nothing is set, 1000 is assumed. The unit is 0.001 mm/min or 0.00001 inch/min in the input unit system.

NOTE

You can not specify 0 for F.

5.5 DWELL (G04)

Format

Dwell G04 X_ ; or G04 P_ ;
X_ : Specify a time (decimal point permitted)
P_ : Specify a time (decimal point not permitted)

Explanations

By specifying a dwell, the execution of the next block is delayed by the specified time. In addition, a dwell can be specified to make an exact check in the feed at programmed rate.
When neither P nor X is specified, exact stop is performed.

Table 5.5 (a) Command value range of the dwell time (Command by X)

Increment system	Command value range	Dwell time unit
IS–A	0.01 to 999999.99	second
IS–B	0.001 to 99999.999	second

Table 5.5 (b) Command value range of the dwell time (Command by P)

Increment system	Command value range	Dwell time unit
IS–A	1 to 99999999	0.001 sec
IS–B	1 to 99999999	0.001 sec

6 REFERENCE POSITION

General

• Reference position

The reference position is a fixed position on a machine tool to which the tool can easily be moved by the reference position return function.
Up to three reference positions can be specified by setting coordinates in the machine coordinate system in parameters (No. 1240 to 1242).

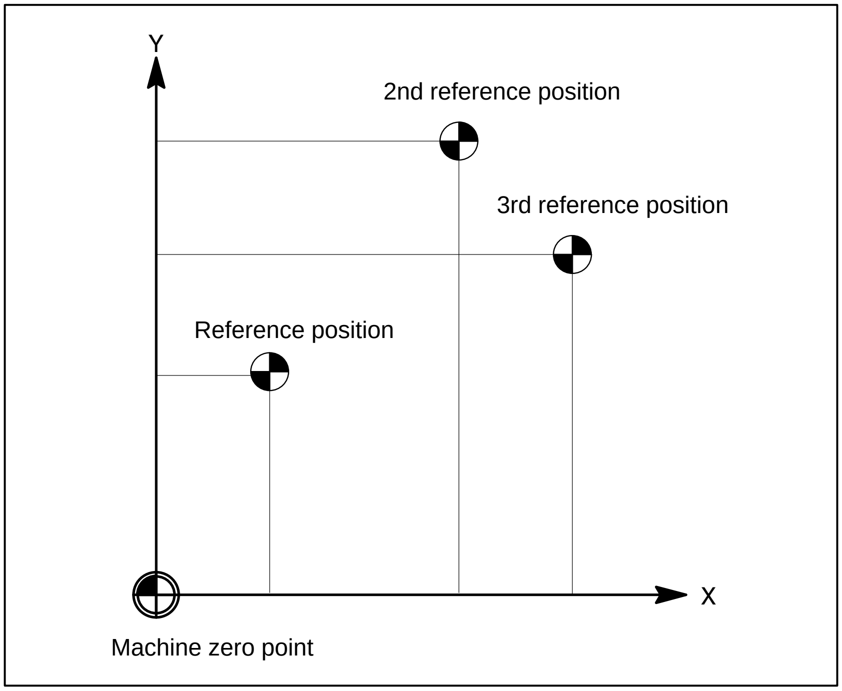


Fig. 6 (a) Machine zero point and reference positions

- **Reference position return and movement from the reference position**

Tools are automatically moved to the reference position via an intermediate position along a specified axis. Or, tools are automatically moved from the reference position to a specified position via an intermediate position along a specified axis. When reference position return is completed, the lamp for indicating the completion of return goes on.

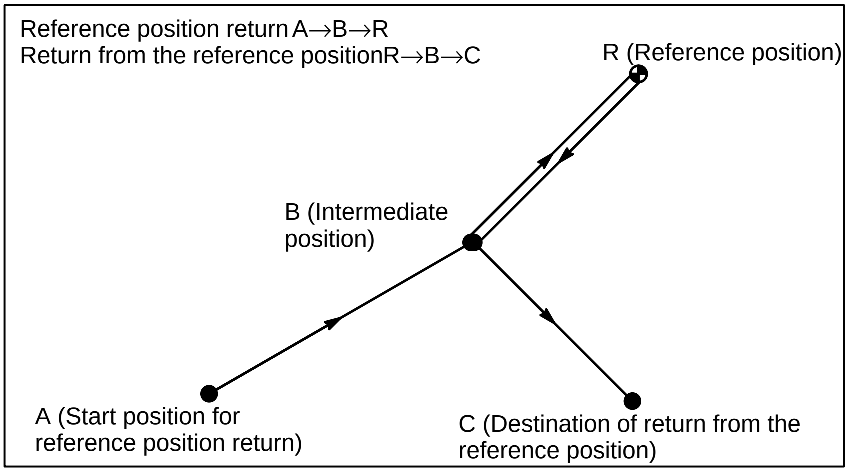


Fig. 6 (b) Reference position return and return form the reference position

- **Reference position return check**

The reference position return check (G27) is the function which checks whether the tool has correctly returned to the reference position as specified in the program. If the tool has correctly returned to the reference position along a specified axis, the lamp for the axis goes on.

Format

Reference position return

G28 IP_ **;**	Reference position return	
G30 P2 IP_ **;**	2nd reference position return	(P2 can be omitted.)
G30 P3 IP_ **;**	3rd reference position return	

IP : Command specifying the intermediate position (Absolute/incremental command)

Return from reference position

G29 IP_ **;**

IP : Command specifying the destination of return from reference position (Absolute/incremental command)

Reference position return check

G27 IP_ **;**

IP : Command specifying the reference position (Absolute/incremental command)

Explanations

- **Reference position return (G28)**

Positioning to the intermediate or reference positions are performed at the rapid traverse rate of each axis.
Therefore, for safety, the tool length compensation should be cancelled before executing this command.
The coordinates for the intermediate position are stored in the controller only for the axes for which a value is specified in a G28 block. For the other axes, the previously specified coordinates are used.

Example N1 G28 X40.0 ; Intermediate position (X40.0)
N2 G28 Y60.0 ; Intermediate position (X40.0, Y60.0)

- **2nd and 3rd reference position return (G30)**

In a system without an absolute–position detector, the second and third reference position return functions can be used only after the reference position return (G28) or manual reference position return (see III–3.1) is made.

- **Return from the reference position (G29)**

In general, it is commanded immediately following the G28 command or G30. For incremental programming, the command value specifies the incremental value from the intermediate point.
Positioning to the intermediate or reference points are performed at the rapid traverse rate of each axis.
When the workpiece coordinate system is changed after the tool reaches the reference position through the intermediate point by the G28 command, the intermediate point also shifts to a new coordinate system. If G29 is then commanded, the tool moves to to the commanded position through the intermediate point which has been shifted to the new coordinate system.
The same operations are performed also for G30 commands.

- **Reference position return check (G27)**

G27 command positions the tool at rapid traverse rate. If the tool reaches the reference position, the reference position return lamp lights up.
However, if the position reached by the tool is not the reference position, an alarm (No. 092) is displayed.

Restrictions

- **Status the machine lock being turned on**

The lamp for indicating the completion of return does not go on when the machine lock is turned on, even when the tool has automatically returned to the reference position. In this case, it is not checked whether the tool has returned to the reference position even when a G27 command is specified.

- **First return to the reference position after the power has been turned on (without an absolute position detector)**

When the G28 command is specified when manual return to the reference position has not been performed after the power has been turned on, the movement from the intermediate point is the same as in manual return to the reference position.
In this case, the tool moves in the direction for reference position return specified in parameter ZMIx (bit 5 of No. 1006). Therefore the specified intermediate position must be a position to which reference position return is possible.

- **Reference position return check in an offset mode**

In an offset mode, the position to be reached by the tool with the G27 command is the position obtained by adding the offset value. Therefore, if the position with the offset value added is not the reference position, the lamp does not light up, but an alarm is displayed instead. Usually, cancel offsets before G27 is commanded.

- **Lighting the lamp when the programmed position does not coincide with the reference position**

When the machine tool system is an inch system with metric input, the reference position return lamp may also light up even if the programmed position is shifted from the reference position by 1μ. This is because the least input increment of the machine tool system is smaller than its least command increment.

Reference

- **Manual reference position return**

See III–3.1.

Examples

G28G90X1000.0Y500.0 ; (Programs movement from A to B)
M06 ; (Changing the tool at the reference position)
G29X1300.0Y200.0 ; (Programs movement from B to C)

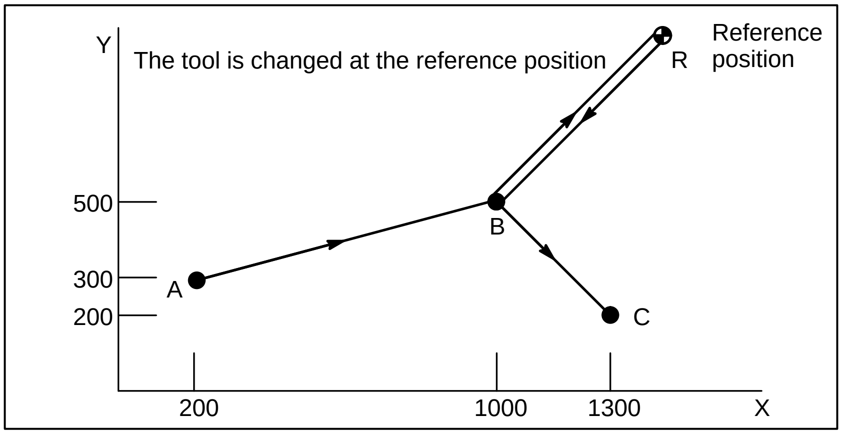


Fig. 6 (c) Reference position return and return from the reference position

7 COORDINATE SYSTEM

By teaching the controller a desired tool position, the tool can be moved to the position. Such a tool position is represented by coordinates in a coordinate system. Coordinates are specified using program axes.
When three program axes, the X–axis, Y–axis, and Z–axis, are used, coordinates are specified as follows:

X_Y_Z_

This command is referred to as a dimension word.

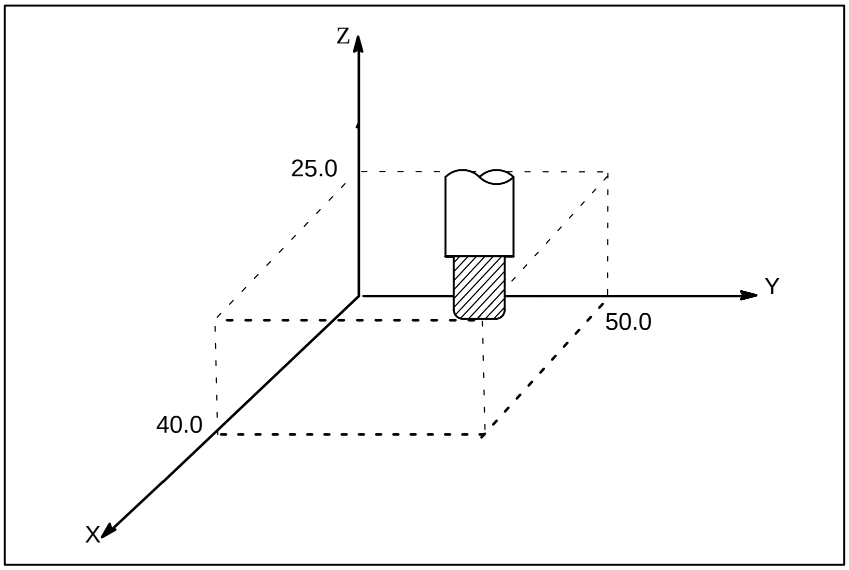


Fig. 7 Tool Position Specified by X40.0Y50.0Z25.0

Coordinates are specified in one of following two coordinate systems:
(1) Machine coordinate system
(2) Workpiece coordinate system

The number of the axes of a coordinate system varies from one machine to another. So, in this manual, a dimension word is represented as IP_.

7.1 MACHINE COORDINATE SYSTEM

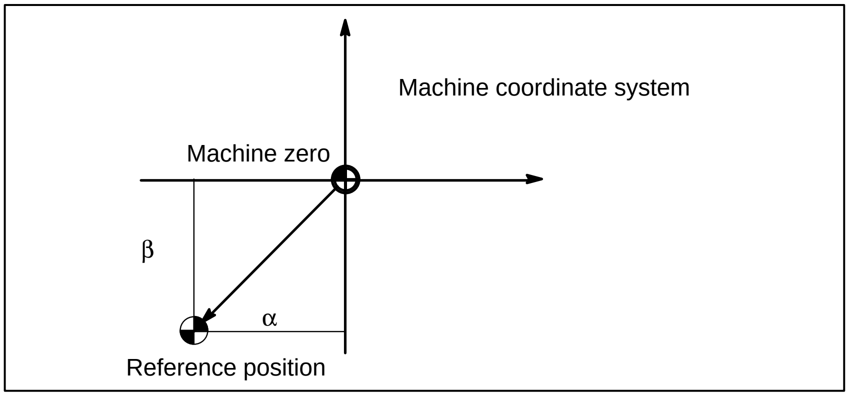
The point that is specific to a machine and serves as the reference of the machine is referred to as the machine zero point. A machine tool builder sets a machine zero point for each machine.

A coordinate system with a machine zero point set as its origin is referred to as a machine coordinate system.

A machine coordinate system is set by performing manual reference position return after power–on (see III–3.1). A machine coordinate system, once set, remains unchanged until the power is turned off.

Reference

When manual reference position return is performed after power–on, a machine coordinate system is set so that the reference position is at the coordinate values of (α, β) set using parameter No.1240.



7.2 WORKPIECE COORDINATE SYSTEM

A coordinate system used for operation of machine is referred to as a workpiece coordinate system. A workpiece coordinate system is to be set with the NC beforehand (setting a workpiece coordinate system).
Program the machine's operation based on the set workpiece coordinate system.
A set workpiece coordinate system can be changed by shifting its origin (changing a workpiece coordinate system).

7.2.1 Setting a Workpiece Coordinate System

A workpiece coordinate system can be set using one of two methods:

(1) Method using G92
A workpiece coordinate system is set by specifying a value after G92 in the program.

(2) Automatic setting
If bit 0 of parameter No. 1201 is set beforehand, a workpiece coordinate system is automatically set when manual reference position return is performed (see Part III–3.1.).

Format

- **Setting a workpiece coordinate system by G92**

```
G92 IP_
```

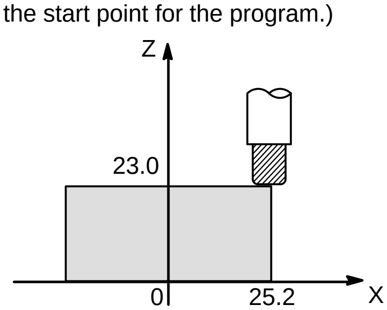
Explanations

A workpiece coordinate system is set so that a point on the tool, such as the tool tip, is at specified coordinates. If a coordinate system is set using G92 during tool length offset, a coordinate system in which the position before offset matches the position specified in G92 is set.

Examples

Example 1

Setting the coordinate system by the
G92X25.2Z23.0; command
(The tool tip is the start point for the program.)



7.2.2 Selecting a Workpiece Coordinate System

The user can choose from set workpiece coordinate systems as described below. (For information about the methods of setting, see Section 7.2.1.)

(1) Selecting a workpiece coordinate system set by G92 or automatic workpiece coordinate system setting
Once a workpiece coordinate system is selected, absolute commands work with the workpiece coordinate system.

7.3 PLANE SELECTION

Machining requires the use of a tool.

Explanations

Table 7.3 Plane selected by G code

G code	Selected plane	Xp	Yp	Zp
G17	Xp Yp plane	X–axis or an axis parallel to it	Y–axis or an axis parallel to it	Z–axis or an axis parallel to it
G18	Zp Xp plane			
G19	Yp Zp plane			

Xp, Yp, Zp are determined by the axis address appeared in the block in which G17, G18 or G19 is commanded.
When an axis address is omitted in G17, G18 or G19 block, it is assumed that the addresses of basic three axes are omitted.
Parameter No. 1022 is used to specify that an optional axis be parallel to the each axis of the X, Y, and Z–axes as the basic three axes.
The plane is unchanged in the block in which G17, G18 or G19 is not commanded.
When the power is turned on or the CNC is reset, G17 (XY plane), G18 (ZX plane), or G19 (YZ plane) is selected by bits 1 (G18) and 2 (G19) of parameter 3402.
The movement instruction is irrelevant to the plane selection.

Examples

Plane selection when the X–axis is parallel with the U–axis.

G17X_Y_ ;	XY plane,
G17U_Y_ ;	UY plane
G18X_Z_ ;	ZX plane
X_Y_ ;	Plane is unchanged (ZX plane)
G17 ;	XY plane
G18 ;	ZX plane
G17 U_ ;	UY plane
G18Y_ ;	ZX plane, Y axis moves regardless without any relation to the plane.

8 COORDINATE VALUE AND DIMENSION

This chapter contains the following topics.

8.1 ABSOLUTE AND INCREMENTAL PROGRAMMING (G90, G91)

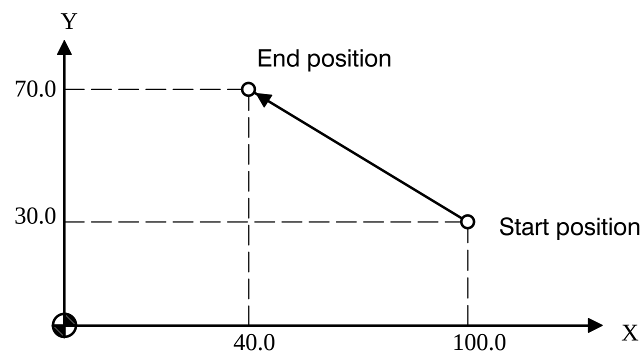
There are two ways to command travels of the tool; the absolute command, and the incremental command. In the absolute command, coordinate value of the end position is programmed; in the incremental command, move distance of the position itself is programmed. G90 and G91 are used to command absolute or incremental command, respectively.

Format

Absolute command	**G90** IP_ **;**
Incremental command	**G91** IP_ **;**

Examples

G90 X40.0 Y70.0 ;	Absolute command
G91 X–60.0 Y40.0 ;	Incremental command



8.2 INCH/METRIC CONVERSION(G20,G21)

Either inch or metric input can be selected by G code.

Format

G20 ;	**Inch input**
G21 ;	**mm input**

This G code must be specified in an independent block before setting the coordinate system at the beginning of the program. After the G code for inch/metric conversion is specified, the unit of input data is switched to the least inch or metric input increment of increment system IS–A or IS–B (Section 2.3). The unit of data input for degrees remains unchanged.The unit systems for the following values are changed after inch/metric conversion:

– Feedrate commanded by F code
– Positional command
– Tool compensation value
– Movement distance in incremental feed
– Some parameters

When the power is turned on, the G code is the same as that held before the power was turned off.

WARNING

1 G20 and G21 must not be switched during a program.
2 When switching inch input (G20) to metric input (G21) and vice versa, the tool compensation value must be re–set according to the least input increment.
3 Reference position return is performed at a low speed for the first G28 command after the inch input is switched to the metric input or vice versa.

NOTE

1 When the least input increment and the least command increment systems are different, the maximum error is half of the least command increment. This error is not accumulated.
2 The inch and metric input can also be switched using settings (See III–11.5.1).

8.3 DECIMAL POINT PROGRAMMING

Numerical values can be entered with a decimal point. A decimal point can be used when entering a distance, time, or speed. Decimal points can be specified with the following addresses:
X, Y, Z, U, V, W, A, B, C, I, J, K, Q, R, and F.

Explanations

There are two types of decimal point notation: calculator–type notation and standard notation.
When calculator–type decimal notation is used, a value without decimal point is considered to be specified in millimeters. When standard decimal notation is used, such a value is considered to be specified in least input increments.Select either calculator–type or standard decimal notation by using the DPI bit (bit 0 of parameter 3401).Values can be specified both with and without decimal point in a single program.

Examples

Program command	Pocket calculator type decimal point programming	Standard type decimal point programming
X1000 Command value without decimal point	1000mm Unit : mm	1mm Unit : Least input increment (0.001 mm)
X1000.0 Command value with decimal point	1000mm Unit : mm	1000mm Unit : mm

WARNING

In a single block, specify a G code before entering a value. The position of decimal point may depend on the command.

Examples:

G20; Input in inches

X1.0 G04; X1.0 is considered to be a distance and processed as X10000. This command is equivalent to G04 X10000. The tool dwells for 10 seconds.

G04 X1.0; Equivalent to G04 X1000. The tool dwells for one second.

NOTE

1 Fractions less than the least input increment are truncated.

Examples:

X1.2345; Truncated to X1.234 when the least input increment is 0.001 mm.
Processed as X1.2345 when the least input increment is 0.0001 inch.

2 When more than eight digits are specified, an alarm occurs. If a value is entered with a decimal point, the number of digits is also checked after the value is converted to an integer according to the least input increment.

Examples:

X1.23456789; Alarm 003 occurs because more than eight digits are specified.

X123456.7; If the least input increment is 0.001 mm, the value is converted to integer 123456700. Because the integer has more than eight digits, an alarm occurs.

9 AUXILIARY FUNCTION

There are two types of auxiliary functions ; miscellaneous function (M code) for specifying program end.
When a move command and miscellaneous function are specified in the same block, the commands are executed in one of the following two ways:

i) Simultaneous execution of the move command and miscellaneous function commands.
ii) Executing miscellaneous function commands upon completion of move command execution.

The selection of either sequence depends on the machine tool builder's specification. Refer to the manual issued by the machine tool builder for details.

9.1 AUXILIARY FUNCTION (M FUNCTION)

When a numeral is specified following address M, code signal and a strobe signal are sent to the machine. The machine uses these signals to turn on or off its functions.
Usually, only one M code can be specified in one block.
Which M code corresponds to which machine function is determined by the machine tool builder.
The machine processes all operations specified by M codes except those specified by M98 or M99. Refer to the machine tool builder's instruction manual for details.

Explanations

The following M codes have special meanings.

- **M02,M30 (End of program)**

This indicates the end of the main program
Automatic operation is stopped and the CNC unit is reset.
This differs with the machine tool builder.
After a block specifying the end of the program is executed, control returns to the start of the program.
Bit 5 of parameter 3404 (M02) can be used to disable M02 from returning control to the start of the program.

- **M00 (Program stop)**

Automatic operation is stopped after a block containing M00 is executed. When the program is stopped, all existing modal information remains unchanged. The automatic operation can be restarted by actuating the cycle operation. This differs with the machine tool builder.

- **M01 (Optional stop)**

Similarly to M00, automatic operation is stopped after a block containing M01 is executed. This code is only effective when the Optional Stop switch on the machine operator's panel has been pressed.

- **M98 (Calling of sub–program)**

This code is used to call a subprogram. The code and strobe signals are not sent. See the subprogram **section 10.3** for details .

- **M99 (End of subprogram)**

This code indicates the end of a subprogram.
M99 execution returns control to the main program. See the subprogram **section 10.3** for details.

> **NOTE**
> The block following M00, M01, M02 and M30, is not read into the input buffer register, if present. Similarly, ten M codes which do not buffer can be set by parameters (Nos. 3411 to 3421). Refer to the machine tool builder's instruction manual for these M codes.

9.2 WAITING FUNCTION (M801 TO M815)

This is the waiting function activated by M–code prepared for smooth waiting operation with another machine or peripheral equipments.
What event to wait for depends on the machine. Refer to the relevant manual from the machine tool builder for details.

9.3 SIMULTANEOUS BLOCK START (M821 TO M827)

This function allows the user to start several blocks simultaneously in up to four Power Mates.
How to start two or more blocks simultaneously depends on the machine. Refer to the relevant manual from the machine tool builder for details.

10 PROGRAM CONFIGURATION

General

- **Main program and subprogram**

There are two program types, main program and subprogram. Normally, the CNC operates according to the main program. However, when a command calling a subprogram is encountered in the main program, control is passed to the subprogram. When a command specifying a return to the main program is encountered in a subprogram, control is returned to the main program.

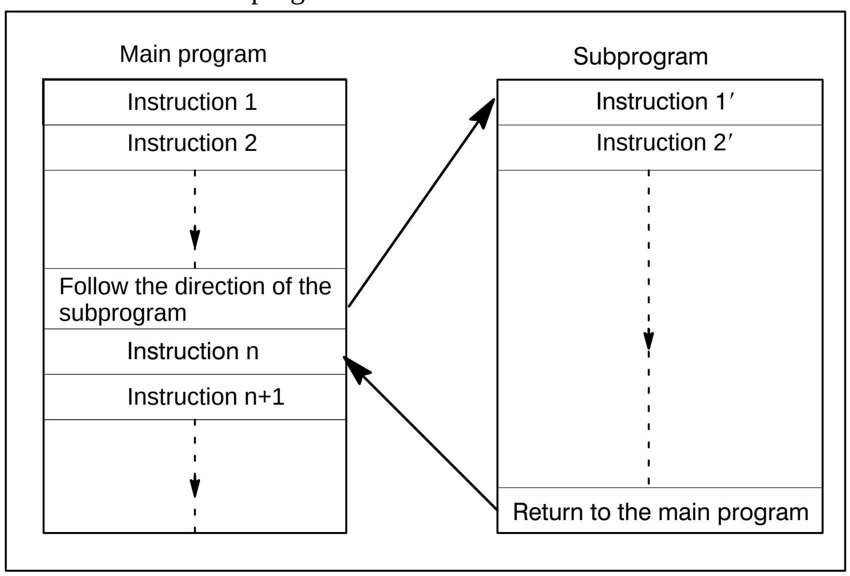


Fig. 10 (a) Main program and Subprogram

The CNC memory can hold up to 400 main programs and subprograms (63 as standard). A main program can be selected from the stored main programs to operate the machine. See Chapter 10 in OPERATION for the methods of registering and selecting programs.

- **Program components**

A program consists of the following components:

Table 10(a) Program components

Components	Descriptions
Tape start	Symbol indicating the start of a program file
Leader section	Used for the title of a program file, etc.
Program start	Symbol indicating the start of a program
Program section	Commands for machining
Comment section	Comments or directions for the operator
Tape end	Symbol indicating the end of a program file

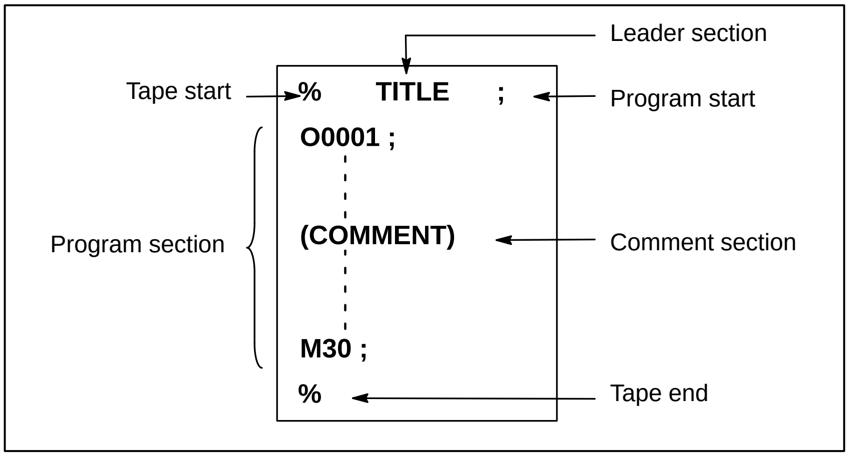


Fig. 10(b) Program configuration (Example of using ISO code)

- **Program section configuration**

A program section consists of several blocks. A program section starts with a program number and ends with a program end code.

<u>**Program section configuration**</u>	<u>**Program section (Example of using ISO code)**</u>
Program number	O0001 ;
Block 1	N1 G91 G00 X120.0 Y80.0 ;
Block 2	N2 G43 Z–32.0 H01 ;
:	:
Block n	Nn M2 ;
Program end	M30 ;

A block contains information necessary for machine operation, such as a move command or on/off command of peripheral device. Specifying a value following a slash (/) at the start of a block disables the execution of some blocks (see ”optional block skip” in Section 10.2).

10.1 PROGRAM COMPONENTS OTHER THAN PROGRAM SECTIONS

This section describes program components other than program sections. See Section 10.2 for a program section.

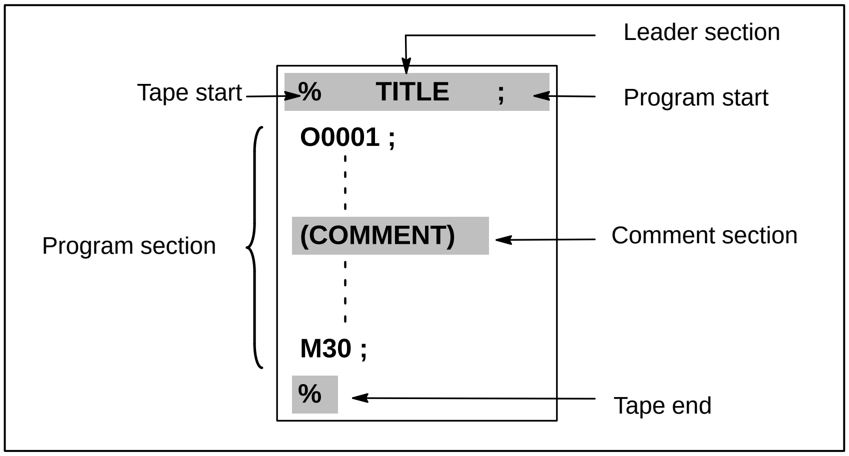


Fig. 10.1(a) Program configuration (Example of using ISO code)

Explanations

• Tape start

The tape start indicates the start of a file that contains NC programs.
The mark is not required when programs are entered using ordinary personal computers. The mark is not displayed on the CRT display screen. However, if the file is output,the mark is automatically output at the start of the file.

Table 10.1(a) Code of a tape start

Name	ISO code	EIA code	Notation in this manual
Tape start	%	ER	%

• Leader section

Data entered before the programs in a file constitutes a leader section.
When machine operation is started, the label skip state is usually set by turning on the power or resetting the system. In the label skip state, all information is ignored until the first end–of–block code is read. When a file is read into the controller from an I/O device, leader sections are skipped by the label skip function.
A leader section generally contains information such as a file header. When a leader section is skipped, even a TV parity check is not made. So a leader section can contain any codes except the EOB code.

• Program start

The program start code is to be entered immediately after a leader section, that is, immediately before a program section.
This code indicates the start of a program, and is always required to disable the label skip function.
With ordinary personal computers, this code can be entered by pressing the return key.

Table 10.1(b) Code of a program start

Name	ISO code	EIA code	Notation in this manual
Program start	LF	CR	;

WARNING

If one file contains multiple programs, the EOB code for label skip operation must not appear before a second or subsequent program number. However, an program start is required at the start of a program if the preceding program ends with %.

- **Comment section**

Any information enclosed by the control–out and control–in codes is regarded as a comment and skipped by the CNC.
The user can enter a header, comments, directions to the operator, etc. in a comment section using the EOB code or any other code.
There is no limit on the length of a comment section.

Table 10.1(c) Codes of a control–in and a control–out

Name	ISO code	EIA code	Notation in this manual	Meaning
Control–out	(	2–4–5	(	Start of comment section
Control–in	)	2–4–7	)	End of comment section

When a command tape is read into memory for memory operation, comment sections, if any, are not ignored but are also read into memory. Note, however, that codes other than those listed in the code table in Appendix A are ignored, and thus are not read into memory.
When data in memory is punched out on paper tape with the punch function, the comment sections are also punched out.
When a program is displayed on the screen, its comment sections are also displayed. However, those codes that were ignored when read into memory are not punched out or displayed.
During memory operation in memory command mode, all comment sections are ignored.
The TV check function can be used for a comment section by setting parameter CTV (bit 1 of No. 0100).

CAUTION

If a long comment section appears in the middle of a program section, a move along an axis may be suspended for a long time because of such a comment section. So a comment section should be placed where movement suspension may occur or no movement is involved.

NOTE

If only a control–in code is read with no matching control–out code, the read control–in code is ignored.

- **Tape end**

A tape end is to be placed at the end of a file containing NC programs.
If programs are entered using the automatic programming system, the mark need not be entered.
If an attempt is made to execute % when M02 or M03 is not placed at the end of the program, the alarm (No. 5010) is occurred.

Table 10.1(d) Code of a tape end

Name	ISO code	EIA code	Notation in this manual
Tape end	%	ER	%

10.2 PROGRAM SECTION CONFIGURATION

This section describes elements of a program section. See Section 10.1 for program components other than program sections.

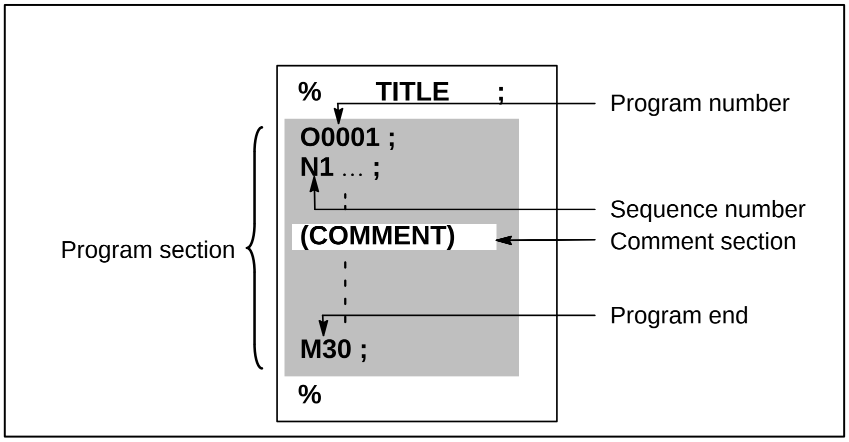


Fig. NO TAG(a) Program configuration (Example of using ISO code)

- **Program number**

A program number consisting of address O followed by a four–digit number is assigned to each program at the beginning registered in memory to identify the program.
In ISO code, the colon (:) can be used instead of O.
When no program number is specified at the start of a program, the sequence number (N....) at the start of the program is regarded as its program number. If a five–digit sequence number is used, the lower four digits are registered as a program number. If the lower four digits are all 0, the program number registered immediately before added to 1 is registered as a program number. Note, however, that N0 cannot be used for a program number.
If there is no program number or sequence number at the start of a program, a program number must be specified using the CRT/MDI panel when the program is stored in memory(See Section 9.3 in Part III.).

NOTE
Program numbers 8000 to 9999 may be used by machine tool builders, and the user may not be able to use these numbers.

- **Sequence number and block**

A program consists of several commands. One command unit is called a block. One block is separated from another with an EOB of end of block code.

Table 10.2(a) EOB code

Name	ISO code	EIA code	Notation in this manual
End of block (EOB)	LF	CR	;

At the head of a block, a sequence number consisting of address N followed by a number not longer than five digits (1 to 99999) can be placed. Sequence numbers can be specified in a random order, and any numbers can be skipped. Sequence numbers may be specified for all blocks or only for desired blocks of the program. In general, however, it is convenient to assign sequence numbers in ascending order in phase with the machining steps (for example, when a new tool is used by tool replacement, and machining proceeds to a new surface with table indexing.)

<u>N300</u> X200.0 Z300.0 ; A sequence number is underlined.

Fig. 10.2(b) Sequence number and block (example)

NOTE

N0 must not be used for the reason of file compatibility with other CNC systems.
Program number 0 cannot be used. So 0 must not be used for a sequence number regarded as a program number.

- **TV check (Vertical parity check along tape)**

A parity check is made for a block on input tape vertically. If the number of characters in one block (starting with the code immediately after an EOB and ending with the next EOB) is odd, an alarm (No.002) is output. No TV check is made only for those parts that are skipped by the label skip function. Bit 1 (CTV) of parameter No. 0100 is used to specify whether comments enclosed in parentheses are counted as characters during TV check. The TV check function can be enabled or disabled by setting on the MDI unit (See Subsec. 11.4.2 in Part III.).

• Block configuration (word and address)

A block consists of one or more words. A word consists of an address followed by a number some digits long. (The plus sign (+) or minus sign (–) may be prefixed to a number.)
Word = Address + number (Example : X–1000)
For an address, one of the letters (A to Z) is used ; an address defines the meaning of a number that follows the address. Table 10.2 (b) indicates the usable addresses and their meanings.
The same address may have different meanings, depending on the preparatory function specification.

Table 10.2(b) Major functions and addresses

Function	Address	Meaning
Program number	O (1)	Program number
Sequence number	N	Sequence number
Preparatory function	G	Specifies a motion mode (linear, etc.)
Dimension word	X, Y, Z, U, V, W, A, B, C	Coordinate axis move command
Feed function	F	Rate of feed per minute, Rate of feed per revolution
Auxiliary function	M	On/off control on the machine tool
Offset number	H	Offset number
Dwell	P, X	Dwell time
Program number designation	P	Subprogram number
Number of repetitions	P	Number of subprogram repetitions

NOTE
In ISO code, the colon (:) can also be used as the address of a program number.

N_	**G_**	**X_ Y_**	**F_**	**M_**	;
Sequence number	Preparatory function	Dimension word	Feed–function	Miscellaneous function	

Fig. 10.2 (c) 1 block (example)

- **Major addresses and ranges of command values**

Major addresses and the ranges of values specified for the addresses are shown below. Note that these figures represent limits on the motion controller side, which are totally different from limits on the machine tool side. For example, the motion controller allows a tool to traverse up to about 100 m (in millimeter input) along the X axis.
However, an actual stroke along the X axis may be limited to 2 m for a specific machine tool.
Similarly, the motion controller may be able to control a feedrate of up to 240 m/min, but the machine tool may not allow more than 3 m/min. When developing a program, the user should carefully read the manuals of the machine tool as well as this manual to be familiar with the restrictions on programming.

Table 10.2(c) Major addresses and ranges of command values

Function		Address	Input in mm	Input in inch
Program number		O (1)	1–9999	1–9999
Sequence number		N	1–99999	1–99999
Preparatory function		G	0–99	0–99
Dimension word	Increment system IS–A	X, Y, Z, U, V, W, A, B, C, I, J, K, R	±999999.99mm	±99999.999inch
	Increment system IS–B		±99999.999mm	±9999.9999inch
Feed per minute	Increment system IS–A	F	1–240000mm/min	0.01–9600.00 inch/min
	Increment system IS–B		1–240000mm/min	0.01–9600.00 inch/min
Feed per revolution		F	0.001–500.00 mm/rev	0.0001–9.9999 inch/rev
Auxiliary function		M	0–99999999	0–99999999
Offset number		H	0–99	0–99
Dwell	Increment system IS–A	X, P	0–999999.99s	0–999999.99s
	Increment system IS–B		0–99999.999s	0–99999.999s
Designation of a program number		P	1–9999	1–9999
Number of repetitions		P	1–9999	1–9999

NOTE
In ISO code, the colon (:) can also be used as the address of a program number.

- **Optional block skip**

When a slash followed by a number (/n (n=1 to 9)) is specified at the head of a block, and optional block skip switch n on the machine operator panel is set to on, the information contained in the block for which /n corresponding to switch number n is specified is ignored in memory operation.
When optional block skip switch n is set to off, the information contained in the block for which /n is specified is valid. This means that the operator can determine whether to skip the block containing /n.
Number 1 for /1 can be omitted. However, when two or more optional block skip switches are used for one block, number 1 for /1 cannot be omitted.

Example)

(Incorrect)	(Correct)
//3 G00X10.0;	/1/3 G00X10.0;

This function is ignored when programs are loaded into memory. Blocks containing /n are also stored in memory, regardless of how the optional block skip switch is set.
Programs held in memory can be output, regardless of how the optional block skip switches are set.
Optional block skip is effective even during sequence number search operation.
Depending on the machine tool, all optional block skip switches (1 to 9) may not be usable. Refer to manuals of the machine tool builder to find which switches are usable.

WARNING

1 Position of a slash

A slash (/) must be specified at the head of a block. If a slash is placed elsewhere, the information from the slash to immediately before the EOB code is ignored.

2 Disabling an optional block skip switch

Optional block skip operation is processed when blocks are read from memory or tape into a buffer. Even if a switch is set to on after blocks are read into a buffer, the blocks already read are not ignored.

NOTE

TV and TH check

When an optional block skip switch is on. TH and TV checks are made for the skipped portions in the same way as when the optional block skip switch is off.

- **Program end**

The end of a program is indicated by punching one of the following codes at the end of the program:

Table 10.2(d) Code of a program end

Code	Meaning usage
M02	For main program
M30	
M99	For subprogram

If one of the program end codes is executed in program execution, the CNC terminates the execution of the program, and the reset state is set. When the subprogram end code is executed, control returns to the program that called the subprogram.

NOTE

A block containing an optional block skip code such as /M02 ; , /M30 ; , or /M99 ; is not regarded as the end of a program, if the optional block skip switch on the machine operator's panel is set to on.
(See Section 10.2 for optional block skip.)

10.3 SUBPROGRAM

If a program contains a fixed sequence or frequently repeated pattern, such a sequence or pattern can be stored as a subprogram in memory to simplify the program.
A subprogram can be called from the main program.
A called subprogram can also call another subprogram.

Format

- **Subprogram configuration**

One subprogram

O □□□□ ;	Subprogram number (or the colon (:) optionally in the case of ISO)
⋮	
M99 ;	Program end

M99 need not constitute a separate block as indicated below.
Example) **X100.0 Y100.0 M99 ;**

- **Subprogram call**

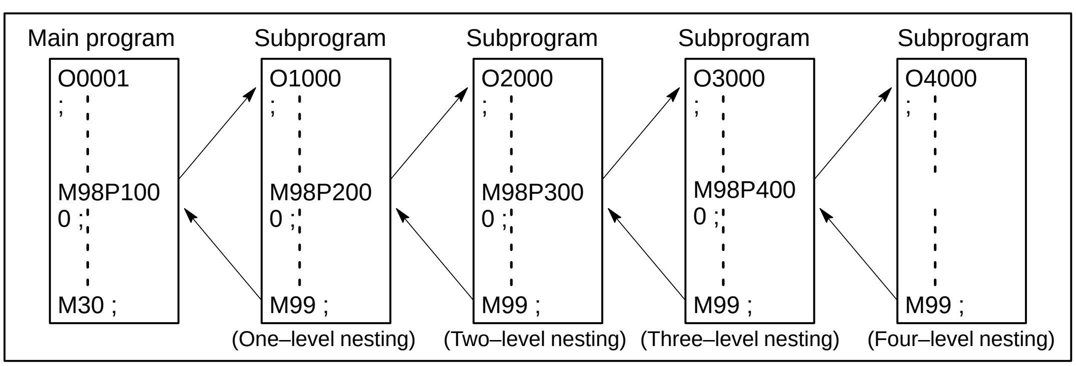
M98 P ○○○○ ○○○○ ;

○○○○ (first four)	○○○○ (last four)
Number of times the subprogram is called repeatedly	Subprogram number

When no repetition data is specified, the subprogram is called just once.

Explanations

When the main program calls a subprogram, it is regarded as a one–level subprogram call. Thus, subprogram calls can be nested up to four levels as shown below.



A single call command can repeatedly call a subprogram up to 9999 times. For compatibility with automatic programming systems, in the first block, Nxxxx can be used instead of a subprogram number that follows O (or :). A sequence number after N is registered as a subprogram number.

Reference

See Chapter 10 in Part III for the method of registering a subprogram.

> **NOTE**
> 1. The M98 and M99 signals are not output to the machine tool.
> 2. If the subprogram number specified by address P cannot be found, an alarm (No. 078) is output.

Examples

☆ M98 P51002 ;

This command specifies ”Call the subprogram (number 1002) five times in succession.” A subprogram call command (M98P_) can be specified in the same block as a move command.

☆ X1000.0 M98 P1200 ;

This example calls the subprogram (number 1200) after an X movement.

☆ Execution sequence of subprograms called from a main program

Main program	1 2 3	Subprogram
N0010 0 ;		O1010 0 ;
N0020 0 ;		N1020 0 ;
N0030 M98 P21010 ;		N1030 0 ;
N0040 0 ;		N1040 0 ;
N0050 M98 P1010 ;		N1050 0 ;
N0060 0 ;		N1060 0 M99 ;

A subprogram can call another subprogram in the same way as a main program calls a subprogram.

Special Usage

- **Specifying the sequence number for the return destination in the main program**

If P is used to specify a sequence number when a subprogram is terminated, control does not return to the block after the calling block, but returns to the block with the sequence number specified by P. Note, however, that P is ignored if the main program is operating in a mode other than memory operation mode.

This method consumes a much longer time than the normal return method to return to the main program.

Main program	Subprogram
N0010 … ;	O0010 … ;
N0020 … ;	N1020 … ;
N0030 M98 P1010 ;	N1030 … ;
N0040 … ;	N1040 … ;
N0050 … ;	N1050 … ;
N0060 … ;	N1060 M99 P0060 ;

- **Using M99 in the main program**

If M99 is executed in a main program, control returns to the start of the main program. For example, M99 can be executed by placing /M99 ; at an appropriate location of the main program and setting the optional block skip function to off when executing the main program. When M99 is executed, control returns to the start of the main program, then execution is repeated starting at the head of the main program.
Execution is repeated while the optional block skip function is set to off. If the optional block skip function is set to on, the /M99 ; block is skipped ; control is passed to the next block for continued execution.
If/M99Pn ; is specified, control returns not to the start of the main program, but to sequence number n. In this case, a longer time is required to return to sequence number n.

```
                        N0010 … ;
                        N0020 … ;
                        N0030 … ;
                        N0040 … ;
Optional block skip     N0050 … ;
OFF                   / N0060 M99 P0030 ;     Optional block skip
                        N0070 … ;             ON
                        N0080 M02 ;
```

- **Using a subprogram only**

A subprogram can be executed just like a main program by searching for the start of the subprogram with the MDI.
(See Section 9.3 in Part III for information about search operation.)
In this case, if a block containing M99 is executed, control returns to the start of the subprogram for repeated execution. If a block containing M99Pn is executed, control returns to the block with sequence number n in the subprogram for repeated execution. To terminate this program, a block containing /M02 ; or /M30 ; must be placed at an appropriate location, and the optional block switch must be set to off ; this switch is to be set to on first.

```
   N1010 … ;
   N1020 … ;
   N1030 … ;
 / N1040 M02 ;            Optional block skip
   N1050 M99 P1020 ;      ON
```

11 COMPENSATION FUNCTION

This chapter describes the following compensation functions:

11.1 TOOL LENGTH OFFSET (G43,G44,G49)

This function can be used by setting the difference between the tool length assumed during programming and the actual tool length of the tool used into the offset memory. It is possible to compensate the difference without changing the program.

Specify the direction of offset with G43 or G44. Select a tool length offset value from the offset memory by entering the corresponding address and number (H code).

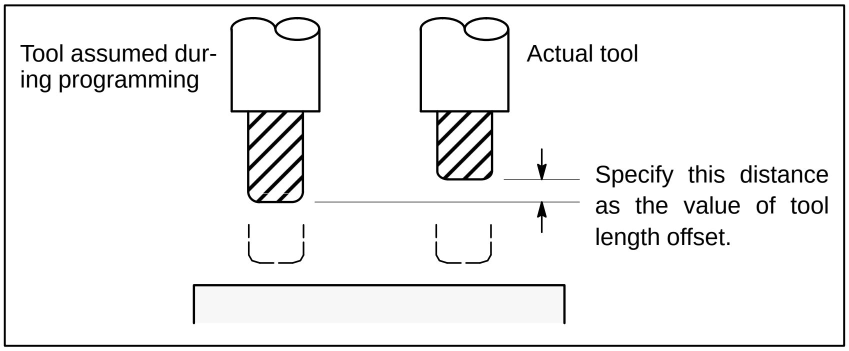


Fig11.1(a) Tool length offset

The following three methods of tool length offset can be used, depending on the axis along which tool length offset can be made.

- **Tool length offset A**
  Compensates for the difference in tool length along the Z–axis.
- **Tool length offset B**
  Compensates for the difference in tool length along the X–,Y–,or Z–axis.
- **Tool length offset C**
  Compensates for the difference in tool length along a specified axis.

Format

Tool length offset A	G43 Z_ H_ ; G44 Z_ H_ ;	Explanation of each address G43 : Positive offset G44 : Negative offset G17 : XY plane selection G18 : ZX plane selection G19 : YZ plane selection α : Address of a specified axis H : Address for specifying the tool length offset value
Tool length offset B	G17 G43 Z_ H_ ; G17 G44 Z_ H_ ; G18 G43 Y_ H_ ; G18 G44 Y_ H_ ; G19 G43 X_ H_ ; G19 G44 X_ H_ ;	
Tool length offset C	G43 α_ H_ ; G44 α_ H_ ;	
Tool length offset cancel	G49 ; or H0 ;	

Explanations

- **Selection of tool length offset**

Select tool length offset A, B, or C, by setting bits 0 and 1 of parameter No. 5001.

- **Direction of the offset**

When G43 is specified, the tool length offset value (stored in offset memory) specified with the H code is added to the coordinates of the end position specified by a command in the program. When G44 is specified, the same value is subtracted from the coordinates of the end position. The resulting coordinates indicate the end position after compensation, regardless of whether the absolute or incremental mode is selected.
If movement along an axis is not specified, the system assumes that a move command that causes no movement is specified. When a positive value is specified for tool length offset with G43, the tool is moved accordingly in the positive direction. When a positive value is specified with G44, the tool is moved accordingly in the negative direction. When a negative value is specified, the tool is moved in the opposite direction.
G43 and G44 are modal G codes. They are valid until another G code belonging to the same group is used.

- **Specification of the tool length offset value**

The tool length offset value assigned to the number (offset number) specified in the H code is selected from offset memory and added to or subtracted from the coordinates specified by a command in the program.
The tool length offset value may be set in the offset memory through the CRT/MDI panel.
The range of values that can be set as the tool length offset value is as follows.

	Metric input	Inch input
Tool length offset value	0 to ±999.999mm	0 to ±99.9999inch

WARNING

When the tool length offset value is changed due to a change of the offset number, the offset value changes to the new tool length offset value, the new tool length offset value is not added to the old tool length offset value.

H1 : tool length offset value 20.0
H2 : tool length offset value 30.0
G90 G43 Z100.0 H1 ; Z will move to 120.0
G90 G43 Z100.0 H2 ; Z will move to 130.0

NOTE

The tool length offset value corresponding to offset No. 0 , that is, H0 always means 0. It is impossible to set any other tool length offset value to H0.

- **Performing tool length offset along two or more axes**

Tool length offset B can be executed along two or more axes when the axes are specified in two or more blocks.

Offset in X and Y axes.
G19 G43 H_ ;   Offset in X axis
G18 G43 H_ ;   Offset in Y axis
(Offsets in X and Y axes are performed)

If the TAL bit (bit 3 of parameter No. 5001) is set to 1, an alarm will not occur even when tool length offset C is executed along two or more axes at the same time.

- **Tool length offset cancel**

To cancel tool length offset, specify G49 or H0. After G49 or H0 is specified, the system immediately cancels the offset mode.

- **G28 in tool length offset mode**

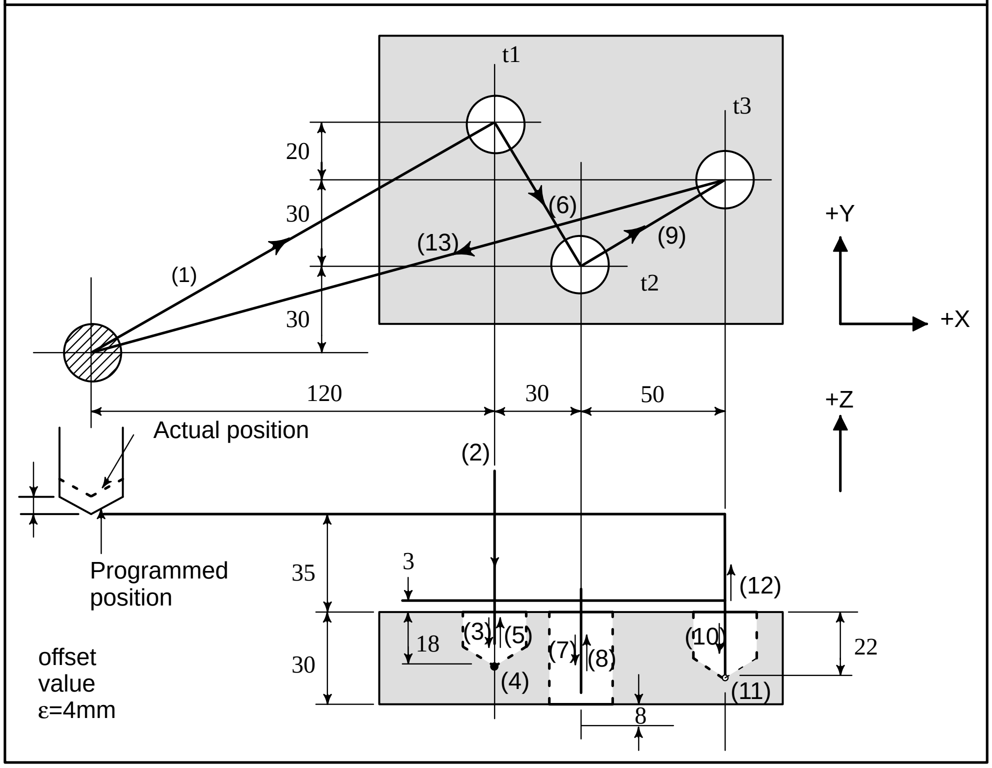
When G28 (automatic reference position return) is executed in tool length offset mode, the tool length offset vector is canceled.
In tool length offset A and B, the canceled tool length offset vector is recovered in the next block if the EVO bit (bit 6 of parameter No. 5001) is set to 1.

WARNING
After tool length offset B is executed along two or more axes, offset along all the axes is canceled by specifying G49. If H0 is specified, only offset along an axis perpendicular to the specified plane is canceled.

NOTE
In the case of the offset in three axes or more, if the offset is canceled by G49 code, the P/S alarm 015 is generated. Cancel the offset by using G49 and H00.

Examples

Tool length offset (in boring holes No.1, 2, and 3)



·Program

```
H1=–4.0 (Tool length offset value)
N1   G91 G00 X120.0 Y80.0 ; .......................... (1)
N2   G43 Z–32.0 H1 ; ................................. (2)
N3   G01 Z–21.0 F1000 ; .............................. (3)
N4   G04 P2000 ; ..................................... (4)
N5   G00 Z21.0 ; ..................................... (5)
N6   X30.0 Y–50.0 ; .................................. (6)
N7   G01 Z–41.0 ; .................................... (7)
N8   G00 Z41.0 ; ..................................... (8)
N9   X50.0 Y30.0 ; ................................... (9)
N10  G01 Z–25.0 ; .................................... (10)
N11  G04 P2000 ; ..................................... (11)
N12  G00 Z57.0 H0 ; .................................. (12)
N13  X–200.0 Y–60.0 ; ................................ (13)
N14  M2 ;
```

11.2 TOOL COMPENSATION VALUES, NUMBER OF COMPENSATION VALUES, AND ENTERING VALUES FROM THE PROGRAM (G10)

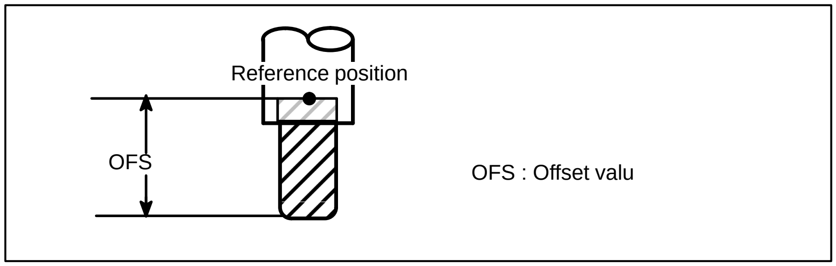


Fig11.2(a) Compensation

Tool compensation values can be entered into CNC memory from the CRT/MDI panel (see section III–8.5) or from a program.
A tool compensation value is selected from the CNC memory when the corresponding code is specified after address H in a program.
The value is used for tool length compensation.

Explanations

- **Valid range of tool compensation values**

Table 11.2(a) shows the valid input range of tool compensation values.

Table11.2 (a) The valid input range of tool compensation value

Increment system	Geometric compensation value	
	Metric input	Inch input
IS–A	±9999.99 mm	±999.999inch
IS–B	±999.999 mm	±99.9999inch

- **Number of tool compensation values and the addresses to be specified**

The memory can hold 99 tool compensation values (option).
Address H is used in the program.
The range of the number that comes after the address (H) depens on the number of tool compensation values : 0 to 99.

- **Input of tool compensation value by programing**

Table11.2 (b) Setting range of Tool compensation memory and Tool compensation value

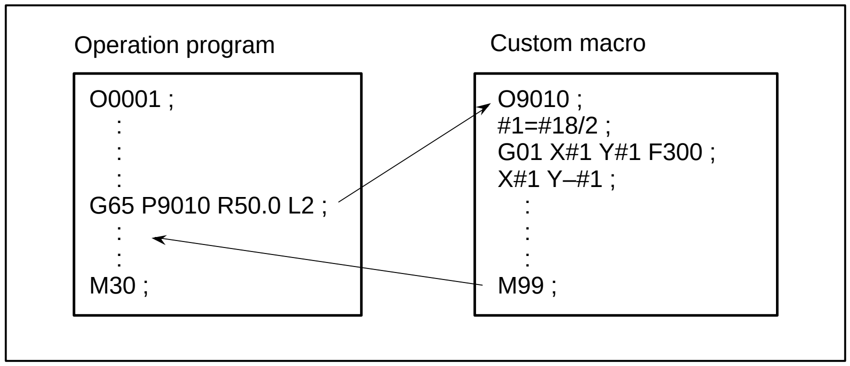
Tool compensation memory	Format
Tool compensation value	G10L11P_R_;

P : Number of tool compensation

R : Tool compensation value in the absolute command(G90) mode Value to be added to the specified tool compensation value in the incremental command(G91) mode (the sum is also a tool compensation value.)

12 CUSTOM MACRO

Although subprograms are useful for repeating the same operation, the custom macro function also allows use of variables, arithmetic and logic operations, and conditional branches for easy development of general programs.A operation program can call a custom macro with a simple command, just like a subprogram.



12.1 VARIABLES

An ordinary operation program specifies a G code and the travel distance directly with a numeric value; examples are G100 and X100.0.
With a custom macro, numeric values can be specified directly or using a variable number. When a variable number is used, the variable value can be changed by a program or using operations on the MDI panel.

```
#1=#2+100 ;

G01 X#1 F300 ;
```

Explanation

• Variable representation

When specifying a variable, specify a number sign (#) followed by a variable number. Personal computers allow a name to be assigned to a variable, but this capability is not available for custom macros.
Example: #1
An expression can be used to specify a variable number. In such a case, the expression must be enclosed in brackets.
Example: #[#1+#2–12]

• Range of variable values

Local and common variables can have value 0 or a value in the following ranges :
-10^{47} to -10^{-29}
0
$+10^{-29}$ to $+10^{47}$
If the result of calculation turns out to be invalid, an alarm No. 111 is issued.

• Omission of the decimal point

When a variable value is defined in a program, the decimal point can be omitted.
Example:
When #1=123; is defined, the actual value of variable #1 is 123.000.

• Undefined variable

When the value of a variable is not defined, such a variable is referred to as a ”null” variable. Variable #0 is always a null variable. It cannot be written to, but it can be read.

• Types of variables

Variables are classified into four types by variable number.

Table 12.1 Types of variables

Variable number	Type of variable	Function
#0	Always null	This variable is always null. No value can be assigned to this variable.
#1 – #33	Local variables	Local variables can only be used within a macro to hold data such as the results of operations. When the power is turned off, local variables are initialized to null. When a macro is called, arguments are assigned to local variables.
#100 – #199 #500 – #699	Common variables	Common variables can be shared among different macro programs. When the power is turned off, variables #100 to #199 are initialized to null. Variables #500 to #699 hold data even when the power is turned off.
#1000 –	System variables	System variables are used to read and write a variety of NC data items such as the current position and tool compensation values.

- **Referencing variables**

To reference the value of a variable in a program, specify a word address followed by the variable number. When an expression is used to specify a variable, enclose the expression in brackets.

Example: G01X[#1+#2]F#3;

A referenced variable value is automatically rounded according to the least input increment of the address.

Example:
When G00X#1; is executed on a 1/1000–mm CNC with 12.3456 assigned to variable #1, the actual command is interpreted as G00X12.346;.

To reverse the sign of a referenced variable value, prefix a minus sign (–) to #.

Example: G00X–#1;

When an undefined variable is referenced, the variable is ignored up to an address word.

Example:
When the value of variable #1 is 0, and the value of variable #2 is null, execution of G00X#1Y#2; results in G00X0;.

• Displaying variable values

Procedure for displaying variable values

Procedure

1 Press the OFFSET SETTING key to display the tool compensation screen.

2 Press the continuous menu key ▷.

3 Press the soft key **[MACRO]** to display the macro variable screen.

4 Enter a variable number, then press soft key **[NO.SRH]**.
The cursor moves to the position of the entered number.

```
 VAR.:                                      O0000 N00000
   NO.        DATA      NO.        DATA
   100       123.456    108
   101         0.000    109
   102                  110
   103      ********    111
   104                  112
   105                  113
   106                  114
   107                  115

 ACTUAL POSITION (WORK)
      X        0.000          Y       0.000
      Z        0.000          B       0.000

 AUTO **** *** ***
[ MACRO ] [ MENU ] [         ] [     ] [ (OPRT) ]
```

·When the value of a variable is blank, the variable is null.
·The mark ******** indicates an overflow (when the absolute value of a variable is greater than 99999999) or an underflow (when the absolute value of a variable is less than 0.0000001).

Limitations

Program numbers, sequence numbers, and optional block skip numbers cannot be referenced using variables.

Example:
Variables cannot be used in the following ways:
O#1;
/#2G00X100.0;
N#3Y200.0;

12.2 SYSTEM VARIABLES

System variables can be used to read and write internal controller data such as tool compensation values and current position data. Note, however, that some system variables can only be read. System variables are essential for automation and general–purpose program development.

Explanations

- **Interface signals**

Signals can be exchanged between the programmable machine controller (PMC) and custom macros.

Table 12.2(a) System variables for interface signals

Variable number	Function
#1000–#1015 #1032	A 16–bit signal can be sent from the PMC to a custom macro. Variables #1000 to #1015 are used to read a signal bit by bit. Variable #1032 is used to read all 16 bits of a signal at one time.
#1100–#1115 #1132	A 16–bit signal can be sent from a custom macro to the PMC. Variables #1100 to #1115 are used to write a signal bit by bit. Variable #1132 is used to write all 16 bits of a signal at one time.
#1133	Variable #1133 is used to write all 32 bits of a signal at one time from a custom macro to the PMC. Note, that values from –99999999 to +99999999 can be used for #1133.

For detailed information, refer to the Connection manual (B–62683EN).

- **PMC D/R area information**

System variables #1200 to #1959 can be used to write and read data to and form the PMC D/R area.
The system variables area associated with the location in the D/R area, as follows :
#1200 to #1219 : Variable time (T)
#1220 to #1224 : Keep relay (K)
#1225 to #1244 : Counter (C)
#1245 to #1709 : Data table (D)
#1710 to #1959 : Internal relay (R)

Table 12.2(b) PMC D/R area information

Variable	Modal information
#1200 #1201 : #1219	T0000, T0001, T0002, T0003 T0004, T0005, T0006, T0007 : T0076, T0077, T0078, T0079
#1220 : #1224	K0000, K0001, K0002, K0003 : K0016, K0017, K0018, K0019
#1225 : #1244	C0000, C0001, C0002, C0003 : C0076, C0077, C0078, C0079
#1245 : #1709	D0000, D0001, D0002, D0003 : D1856, D1857, D1858, D1859
#1710 : #1959	R0000, R0001, R0002, R0003 : R0996, R0997, R0998, R0999

NOTE
The input range is –2147483648 to 2147483647.

- **Tool compensation values**

Tool compensation values can be read and written using system variables. Variables #2001 to #2400 can also be used.

Table 12.2(b) System variables for tool compensation memory

Compensation number	System variable
1 : 99	#10001 (#2001) : #10099 (#2099)

- **Macro alarms**

Table 12.2(c) System variable for macro alarms

Variable number	Function
#3000	When a value from 0 to 200 is assigned to variable #3000, the NC stops with an alarm. After an expression, an alarm message not longer than 26 characters can be described. The CRT screen displays alarm numbers by adding 3000 to the value in variable #3000 along with an alarm message.

Example:
#3000=1(TOOL NOT FOUND);
→ The alarm screen displays ”3001 TOOL NOT FOUND.”

- **Time information**

Time information can be read and written.

Table 12.2(d) System variables for time information

Variable number	Function
#3001	This variable functions as a timer that counts in 1–millisecond increments at all times. When the power is turned on, the value of this variable is reset to 0. When 65535 milliseconds is reached, the value of this timer returns to 0.
#3002	This variable functions as a timer that counts in 1–hour increments when the cycle start lamp is on. This timer preserves its value even when the power is turned off. When 1145324.612 hours is reached, the value of this timer returns to 0.

- **Automatic operation control**

The control state of automatic operation can be changed.

Table 12.2(e) System variable (#3003) for automatic operation control

#3003	Single block	Completion of an auxiliary function
0	Enabled	To be awaited
1	Disabled	To be awaited
2	Enabled	Not to be awaited
3	Disabled	Not to be awaited

- When the power is turned on, the value of this variable is 0.
- When single block stop is disabled, single block stop operation is not performed even if the single block switch is set to ON.
- When a wait for the completion of auxiliary functions (M function) is not specified, program execution proceeds to the next block before completion of auxiliary function. Also, distribution completion signal DEN is not output.

Table 12.2(f) System variable (#3004) for automatic operation control

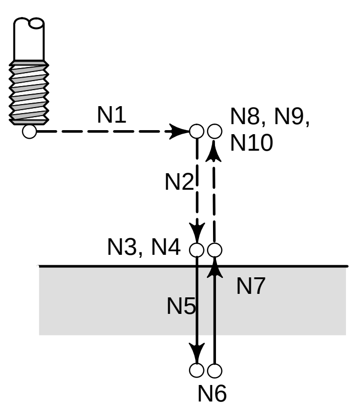
#3004	Feed hold	Feedrate Override	Exact stop
0	Enabled	Enabled	Enabled
1	Disabled	Enabled	Enabled
2	Enabled	Disabled	Enabled
3	Disabled	Disabled	Enabled
4	Enabled	Enabled	Disabled
5	Disabled	Enabled	Disabled
6	Enabled	Disabled	Disabled
0	Enabled	Enabled	Enabled
7	Disabled	Disabled	Disabled

- When the power is turned on, the value of this variable is 0.
- When feed hold is disabled:

(1) When the feed hold button is held down, the machine stops in the single block stop mode. However, single block stop operation is not performed when the single block mode is disabled with variable #3003.

(2) When the feed hold button is pressed then released, the feed hold lamp comes on, but the machine does not stop; program execution continues and the machine stops at the first block where feed hold is enabled.

- When feedrate override is disabled, an override of 100% is always applied regardless of the setting of the feedrate override switch on the machine operator's panel.
- When exact stop check is disabled, no exact stop check (position check) is made even in blocks including those which do not perform feed at programmed rate.




```
O0001 ;
N1  G00 G91 X#24 Y#25 ;
N2  Z#18 ;
    G04 ;
N3  #3003=3 ;
N4  #3004=7 ;
N5  G01 Z#26 F#9 ;
N6  M04 ;
N7  G01
Z–[ROUND[#18]+ROUND[#26]] ;
    G04 ;
N8  #3004=0 ;
N9  #3003=0 ;
N10 M03 ;
    M99 ;
```

Fig. 12.2(a) Example of using variable #3004 in a tapping cycle

- **Settings**

Settings can be read and written. Binary values are converted to decimals.

#3005

	#15	#14	#13	#12	#11	#10	#9	#8
Setting								
	#7	**#6**	**#5**	**#4**	**#3**	**#2**	**#1**	**#0**
Setting			SEQ			INI	ISO	TVC

#5 (SEQ) : Whether to automatically insert sequence numbers
#2 (INI) : Millimeter input or inch input
#1 (ISO): Whether to use EIA or ISO as the output code
#0 (TVC) : Whether to make a TV check

- **Mirror image**

The mirror–image status for each axis set using an external switch or setting operation can be read through the output signal (mirror–image check signal). The mirror–image status present at that time can be checked. (See Section 4.3 in III.)
The value obtained in binary is converted into decimal notation.

#3007

	#7	#6	#5	#4	#3	#2	#1	#0
Setting			6th axis	5th axis	4th axis	3th axis	2th axis	1th axis

For each bit, 0 (mirror–image function is disabled) or 1 (mirror–image function is enabled) is indicated.

Example : If #3007 is 3, the mirror–image function is enabled for the first and second axes.

- When the mirror–image function is set for a certain axis by both the mirror–image signal and setting, the signal value and setting value are ORed and then output.
- When mirror–image signals for axes other than the controlled axes are turned on, they are still read into system variable #3007.
- System variable #3007 is a write–protected system variable. If an attempt is made to write data in the variable, P/S 116 alarm ”WRITE PROTECTED VARIABLE” is issued.

- **Number of machined parts**

The number (target number) of parts required and the number (completion number) of machined parts can be read and written.

Table 12.2(g) System variables for the number of parts required and the number of machined parts

Variable number	Function
#3901	Number of machined parts (completion number)
#3902	Number of required parts (target number)

NOTE
Do not substitute a negative value.

• Modal information

Modal information specified in blocks up to the immediately preceding block can be read.

Table 12.2(h) System variables for modal information

Variable number	Function	
#4001	G00, G01	(Group 01)
#4002	G17, G18, G19	(Group 02)
#4003	G90, G91	(Group 03)
#4004		(Group 04)
#4005	G94, G95	(Group 05)
#4006	G20, G21	(Group 06)
#4008	G43, G44, G49	(Group 08)
#4012	G65, G66, G67	(Group 12)
#4109	F code	
#4111	H code	
#4113	M code	
#4114	Sequence number	
#4115	Program number	

Example:
When #1=#4001; is executed, the resulting value in #1 is 0, 1, 2 or 3.

• Current position

Position information cannot be written but can be read.

Table 12.2(i) System variables for position information

<table>
<tr><th>Variable number</th><th>Position information</th><th>Coordinate system</th><th>Tool com-pensation value</th><th>Read operation during movement</th></tr>
<tr><td>#5001–#5006</td><td>Block end point</td><td>Workpiece coordinate system</td><td>Not included</td><td>Enabled</td></tr>
<tr><td>#5021–#5026</td><td>Current position</td><td>Machine coordinate system</td><td rowspan="3">Included</td><td rowspan="2">Disabled</td></tr>
<tr><td>#5041–#5046</td><td>Current position</td><td rowspan="2">Workpiece coordinate system</td></tr>
<tr><td>#5061–#5066</td><td>Skip signal position</td><td>Enabled</td></tr>
<tr><td>#5081–#5086</td><td>Tool offset value</td><td></td><td></td><td rowspan="2">Disabled</td></tr>
<tr><td>#5101–#5106</td><td>Deviated servo position</td><td></td><td></td></tr>
</table>

· The first digit (from 1 to 8) represents an axis number.

· The tool offset value currently used for execution rather than the immediately preceding tool offset value is held in variables #5081 to 5088.

· The tool position where the skip signal is turned on in a G31 (skip function) block is held in variables #5061 to #5068. When the skip signal is not turned on in a G31 block, the end point of the specified block is held in these variables.

· When read during movement is ”disabled,” this means that expected values cannot be read due to the buffering (preread) function.

12.3 ARITHMETIC AND LOGIC OPERATION

The operations listed in Table 12.3(a) can be performed on variables. The expression to the right of the operator can contain constants and/or variables combined by a function or operator. Variables #j and #K in an expression can be replaced with a constant. Variables on the left can also be replaced with an expression.

Table 12.3(a) Arithmetic and logic operation

Function	Format	Remarks
Definition	#i=#j	
Sum Difference Product Quotient	#i=#j+#k; #i=#j–#k; #i=#j*#k; #i=#j/#k;	
Sine Cosine Tangent Arctangent	#i=SIN[#j]; #i=COS[#j]; #i=TAN[#j]; #i=ATAN[#j]/[#k];	An angle is specified in degrees. 90 degrees and 30 minutes is represented as 90.5 degrees.
Square root Absolute value Rounding off Rounding down Rounding up	#i=SQRT[#j]; #i=ABS[#j]; #i=ROUND[#j]; #i=FIX[#j]; #i=FUP[#j];	
OR XOR AND	#i=#j OR #k; #i=#j XOR #k; #i=#j AND #k;	A logical operation is performed on binary numbers bit by bit.
Conversion from BCD to BIN Conversion from BIN to BCD	#i=BIN[#j]; #i=BCD[#j];	Used for signal exchange to and from the PMC

Explanations

- **Angle units**

The units of angles used with the SIN, COS, TAN, and ATAN functions are degrees. For example, 90 degrees and 30 minutes is represented as 90.5 degrees.

- **ATAN function**

After the ATAN function, specify the lengths of two sides separated by a slash. A result is found where 0 ≦ result < 360.

Example :
When #1=ATAN[1]/[–1], the value of #1 is 135.0

- **ROUND function**

· When the ROUND function is included in an arithmetic or logic operation command, IF statement, or WHILE statement, the ROUND function rounds off at the first decimal place.

Example:
When #1=ROUND[#2]; is executed where #2 holds 1.2345, the value of variable #1 is 1.0.

· When the ROUND function is used in NC statement addresses, the ROUND function rounds off the specified value according to the least input increment of the address.

Example:

Creation of a drilling program that cuts according to the values of variables #1 and #2, then returns to the original position
Suppose that the increment system is 1/1000 mm, variable #1 holds 1.2345, and variable #2 holds 2.3456. Then,
G00 G91 X–#1; Moves 1.235 mm.
G01 X–#2 F300; Moves 2.346 mm.
G00 X[#1+#2];
Since 1.2345 + 2.3456 = 3.5801, the travel distance is 3.580, which does not return the tool to the original position.
This difference comes from whether addition is performed before or after rounding off. G00X–[ROUND[#1]+ROUND[#2]] must be specified to return the tool to the original position.

- **Rounding up and down to an integer**

When the absolute value of the integer produced by an operation on a number is greater than the absolute value of the original number, such an operation is referred to as rounding up to an integer. Conversely, when the absolute value of the integer produced by an operation on a number is less than the absolute value of the original number, such an operation is referred to as rounding down to an integer. Be particularly careful when handling negative numbers.

Example:

Suppose that #1=1.2 and #2=–1.2.
When #3=FUP[#1] is executed, 2.0 is assigned to #3.
When #3=FIX[#1] is executed, 1.0 is assigned to #3.
When #3=FUP[#2] is executed, –2.0 is assigned to #3.
When #3=FIX[#2] is executed, –1.0 is assigned to #3.

- **Abbreviations of arithmetic and logic operation commands**

When a function is specified in a program, the first two characters of the function name can be used to specify the function.

Example:

ROUND → RO
FIX → FI

- **Priority of operations**

(1) Functions
(2) Operations such as multiplication and division (*, /, AND, MOD)
(3) Operations such as addition and subtraction (+, –, OR, XOR)

Example) $\#1=\underbrace{\#2+\underbrace{\#3*\underbrace{SIN[\#4]}_{(1)}}_{(2)}}_{(3)};$

(1), (2), and (3) indicate the order of operations.

- **Bracket nesting**

Brackets are used to change the order of operations. Brackets can be used to a depth of five levels including the brackets used to enclose a function. When a depth of five levels is exceeded, alarm No. 118 occurs.

Example) #1=SIN [[[#2+#3] *#4 +#5] *#6] ;

(1)

(2)

(3)

(4)

(5)

(1) to (5) indicate the order of operations.

Limitations

- **Brackets**

Brackets ([,]) are used to enclose an expression. Note that parentheses (,) are used for comments.

- **Operation error**

Errors may occur when operations are performed.

Table 12.3(b) Errors involved in operations

Operation	Average error	Maximum error	Type of error
a = b*c	1.55×10^{-10}	4.66×10^{-10}	Relative error(*1) $\left\lvert\frac{\varepsilon}{a}\right\rvert$
a = b / c	4.66×10^{-10}	1.88×10^{-9}	
$a = \sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
a = b + c a = b – c	2.33×10^{-10}	5.32×10^{-10}	(*2) Min $\left\lvert\frac{\varepsilon}{b}\right\rvert\left\lvert\frac{\varepsilon}{c}\right\rvert$
a = SIN [b] a = COS [b]	5.0×10^{-9}	1.0×10^{-8}	Absolute error(*3) $\lvert\varepsilon\rvert$ degrees
a = ATAN [b] / [c] (*4)	1.8×10^{-6}	3.6×10^{-6}	

NOTE

1. The relative error depends on the result of the operation.
2. Smaller of the two types of errors is used.
3. The absolute error is constant, regardless of the result of the operation.
4. Function TAN performs SIN/COS.

· The precision of variable values is about 8 decimal digits. When very large numbers are handled in an addition or subtraction, the expected results may not be obtained.

Example:

When an attempt is made to assign the following values to variables #1 and #2:
#1=9876543210123.456
#2=9876543277777.777
the values of the variables become:
#1=9876543200000.000
#2=9876543300000.000
In this case, when #3=#2–#1; is calculated, #3=100000.000 results. (The actual result of this calculation is slightly different because it is performed in binary.)

· Also be aware of errors that can result from conditional expressions using EQ, NE, GE, GT, LE, and LT.

Example:

IF[#1 EQ #2] is effected by errors in both #1 and #2, possibly resulting in an incorrect decision.
Therefore, instead find the difference between the two variables with IF[ABS[#1–#2]LT0.001].
Then, assume that the values of the two variables are equal when the difference does not exceed an allowable limit (0.001 in this case).

· Also, be careful when rounding down a value.

Example:

When #2=#1*1000; is calculated where #1=0.002;, the resulting value of variable #2 is not exactly 2 but 1.99999997.
Here, when #3=FIX[#2]; is specified, the resulting value of variable #1 is not 2.0 but 1.0. In this case, round down the value after correcting the error so that the result is greater than the expected number, or round it off as follows:
#3=FIX[#2+0.001]
#3=ROUND[#2]

- **Divisor**

When a divisor of zero is specified in a division or TAN[90], alarm No. 112 occurs.

12.4 MACRO STATEMENTS AND NC STATEMENTS

The following blocks are referred to as macro statements:

· **Blocks containing an arithmetic or logic operation (=)**

· **Blocks containing a control statement (such as GOTO, DO, END)**

· **Blocks containing a macro call command (such as macro calls by G65, G66, G67, or other G codes, or by M codes)**

Any block other than a macro statement is referred to as an NC statement.

Explanations

- **Differences from NC statements**

· Even when single block mode is on, the machine does not stop. Note, however, that the machine stops in the single block mode when bit 5 of parameter 6000 is 1.

- **NC statements that have the same property as macro statements**

·NC statements that include a subprogram call command (such as subprogram calls by M98 or other M codes) and also include an O, N, P, or L address have the same property as macro statements.

· NC statements that include M99 and an O, N, L, or P address have the same property as macro statements.

12.5 BRANCH AND REPETITION

In a program, the flow of control can be changed using the GOTO statement and IF statement. Three types of branch and repetition operations are used:

Branch and repetition
- GOTO statement (unconditional branch)
- IF statement (conditional branch: if ..., then...)
- WHILE statement (repetition while ...)

12.5.1 Unconditional Branch (GOTO Statement)

A branch to sequence number n occurs. When a sequence number outside of the range 1 to 99999 is specified, alarm No. 128 occurs. A sequence number can also be specified using an expression.

GOTO n ;	n: Sequence number (1 to 99999)

Example:
GOTO1;
GOTO#10;

12.5.2 Conditional Branch (IF Statement)

Specify a conditional expression after IF. If the specified conditional expression is satisfied, a branch to sequence number n occurs. If the specified condition is not satisfied, the next block is executed.

If the value of variable #1 is greater than 10, a branch to sequence number N2 occurs.

If the condition is not satisfied

```
IF [#1 GT 10] GOTO 2 ;
Processing
N2 G00 G91 X10.0 ;
:
```

If the condition is satisfied

Explanations

- **Conditional expression**

A conditional expression must include an operator inserted between two variables or between a variable and constant, and must be enclosed in brackets ([,]). An expression can be used instead of a variable.

- **Operators**

Operators each consist of two letters and are used to compare two values to determine whether they are equal or one value is smaller or greater than the other value. Note that the inequality sign cannot be used.

Table 12.5.2 Operators

Operator	Meaning
EQ	Equal to(=)
NE	Not equal to(≠)
GT	Greater than(>)
GE	Greater than or equal to(≧)
LT	Less than(<)
LE	Less than or equal to(≦)

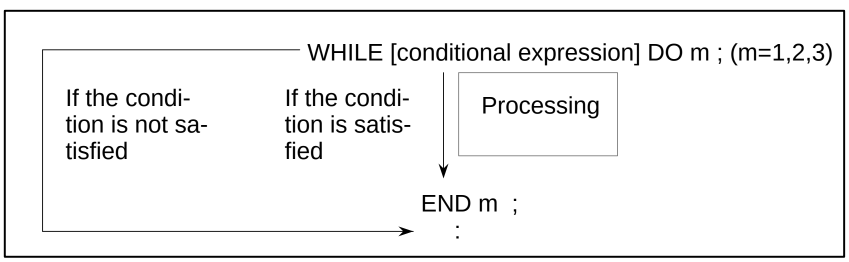
Sample program

The sample program below finds the total of numbers 1 to 10.

```
O9500;
 #1=0;Initial value of the variable to hold the sum
 #2=1;Initial value of the variable as an addend
N1 IF[#2 GT 10] GOTO 2;   Branch to N2 when the addend is greater than
10
 #1=#1+#2;  Calculation to find the sum
 #2=#2+1; Next addend
 GOTO 1;  Branch to N1
N2 M30;End of program
```

12.5.3 Repetition (WHILE Statement)

Specify a conditional expression after WHILE. While the specified condition is satisfied, the program from DO to END is executed. If the specified condition is not satisfied, program execution proceeds to the block after END.

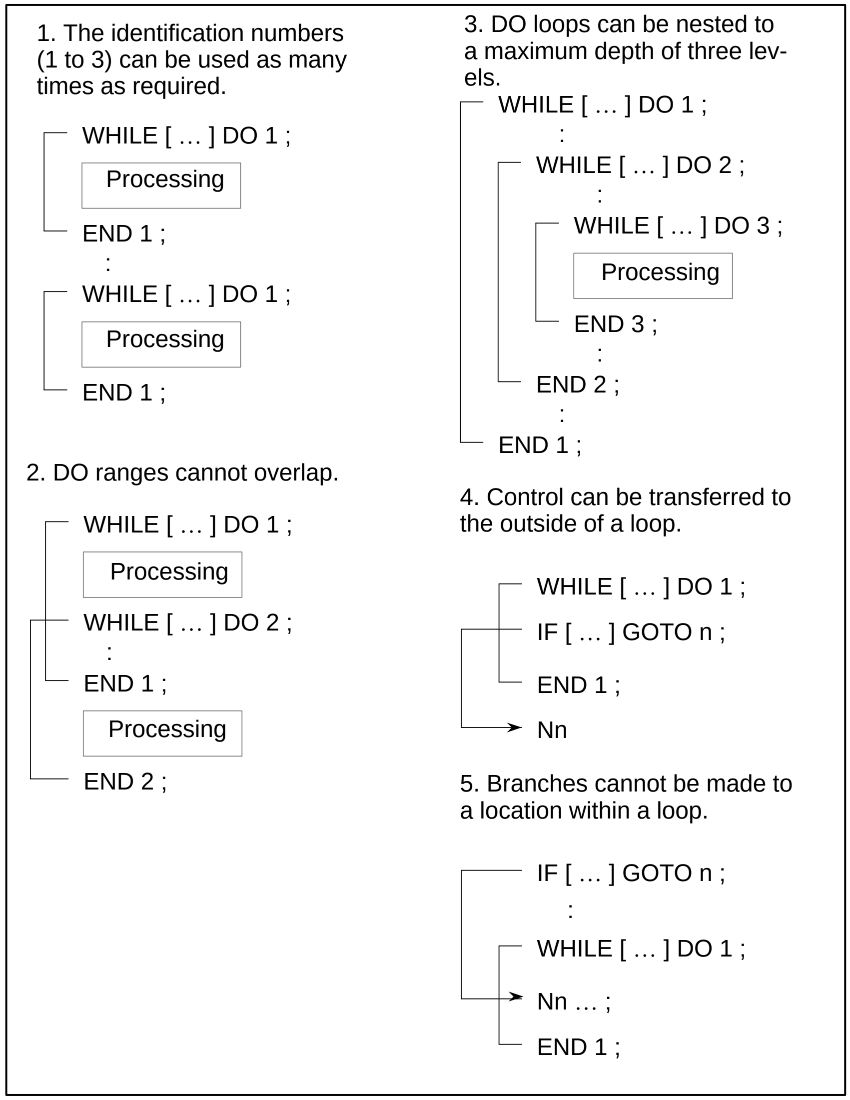


Explanations

While the specified condition is satisfied, the program from DO to END after WHILE is executed. If the specified condition is not satisfied, program execution proceeds to the block after END. The same format as for the IF statement applies. A number after DO and a number after END are identification numbers for specifying the range of execution. The numbers 1, 2, and 3 can be used. When a number other than 1, 2, and 3 is used, alarm No. 126 occurs.

- **Nesting**

The identification numbers (1 to 3) in a DO–END loop can be used as many times as desired. Note, however, when a program includes crossing repetition loops (overlapped DO ranges), alarm No. 124 occurs.



Limitations

- **Infinite loops**

When DO m is specified without specifying the WHILE statement, an infinite loop ranging from DO to END is produced.

- **Processing time**

When a branch to the sequence number specified in a GOTO statement occurs, the sequence number is searched for. For this reason, processing in the reverse direction takes a longer time than processing in the forward direction. Using the WHILE statement for repetition reduces processing time.

- **Undefined variable**

In a conditional expression that uses EQ or NE, a null value and zero have different effects. In other types of conditional expressions, a null value is regarded as zero.

Sample program

The sample program below finds the total of numbers 1 to 10.

```
O0001;
#1=0;
#2=1;
WHILE[#2 LE 10]DO
1;
#1=#1+#2;
#2=#2+1;
END 1;
M30;
```

12.6 MACRO CALL

A macro program can be called using the following methods:

Macro call
- Simple call ((G65)
- modal call (G66, G67)
- Macro call with G code
- Macro call with M code
- Subprogram call with M code

Limitations

● Differences between macro calls and subprogram calls

Macro call (G65) differs from subprogram call (M98) as described below.

· With G65, an argument (data passed to a macro) can be specified. M98 does not have this capability.

· When an M98 block contains another NC command (for example, G01 X100.0 M98Pp), the subprogram is called after the command is executed. On the other hand, G65 unconditionally calls a macro.

· When an M98 block contains another NC command (for example, G01 X100.0 M98Pp), the machine stops in the single block mode. On the other hand, G65 does not stops the machine.

· With G65, the level of local variables changes. With M98, the level of local variables does not change.

12.6.1 Simple Call (G65)

When G65 is specified, the custom macro specified at address P is called. Data (argument) can be passed to the custom macro program.

G65 P p L ℓ <argument–specification> ;

P :	Number of the program to call
ℓ :	Repetition count (1 by default)
Argument :	Data passed to the macro

```
O0001 ;
   :
G65 P9010 L2 A1.0 B2.0 ;
   :
M30 ;
```

```
O9100 ;
#3=#1+#2 ;
IF [#3 GT 360] GOTO 9 ;
G00 G91 X#3 ;
N9 M99 ;
```

Explanations

● Call

· After G65, specify at address P the program number of the custom macro to call.

· When a number of repetitions is required, specify a number from 1 to 9999 after address L. When L is omitted, 1 is assumed.

· By using argument specification, values are assigned to corresponding local variables.

- **Argument specification**

Two types of argument specification are available. Argument specification I uses letters other than G, L, O, N, and P once each. Argument specification II uses A, B, and C once each and also uses I, J, and K up to ten times. The type of argument specification is determined automatically according to the letters used.

Argument specification I

Address	Variable number
A	#1
B	#2
C	#3
D	#7
E	#8
F	#9
H	#11

Address	Variable number
I	#4
J	#5
K	#6
M	#13
Q	#17
R	#18
S	#19

Address	Variable number
T	#20
U	#21
V	#22
W	#23
X	#24
Y	#25
Z	#26

· Addresses G, L, N, O, and P cannot be used in arguments.

· Addresses that need not be specified can be omitted. Local variables corresponding to an omitted address are set to null.

Argument specification II

Argument specification II uses A, B, and C once each and uses I, J, and K up to ten times. Argument specification II is used to pass values such as three–dimensional coordinates as arguments.

Address	Variable number
A	#1
B	#2
C	#3
I_1	#4
J_1	#5
K_1	#6
I_2	#7
J_2	#8
K_2	#9
I_3	#10
J_3	#11

Address	Variable number
K_3	#12
I_4	#13
J_4	#14
K_4	#15
I_5	#16
J_5	#17
K_5	#18
I_6	#19
J_6	#20
K_6	#21
I_7	#22

Address	Variable number
J_7	#23
K_7	#24
I_8	#25
J_8	#26
K_8	#27
I_9	#28
J_9	#29
K_9	#30
I_{10}	#31
J_{10}	#32
K_{10}	#33

· Subscripts of I, J, and K for indicating the order of argument specification are not written in the actual program.

Limitations

- **Format**

G65 must be specified before any argument.

- **Mixture of argument specifications I and II**

The NC internally identifies argument specification I and argument specification II. If a mixture of argument specification I and argument specification II is specified, the type of argument specification specified later takes precedence.

- **Position of the decimal point**

The units used for argument data passed without a decimal point correspond to the least input increment of each address. The value of an argument passed without a decimal point may vary according to the system configuration of the machine. It is good practice to use decimal points in macro call arguments to maintain program compatibility.

- **Call nesting**

Calls can be nested to a depth of four levels including simple calls (G65) and modal calls (G66). This does not include subprogram calls (M98).

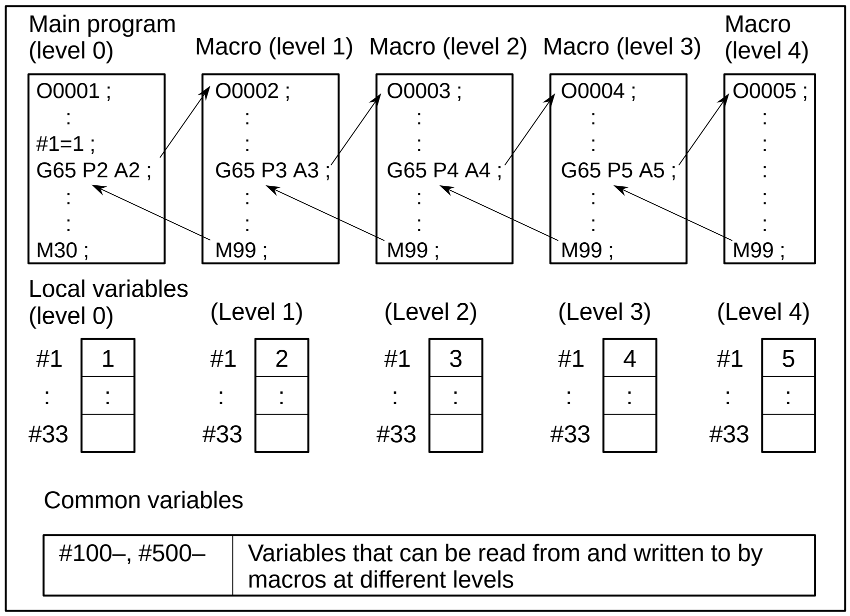
- **Local variable levels**

· Local variables from level 0 to 4 are provided for nesting.

· The level of the main program is 0.

· Each time a macro is called (with G65 or G66), the local variable level is incremented by one. The values of the local variables at the previous level are saved in the NC.

· When M99 is executed in a macro program, control returns to the calling program. At that time, the local variable level is decremented by one; the values of the local variables saved when the macro was called are restored.



12.6.2 Modal Call (G66)

Once G66 is issued to specify a modal call a macro is called after a block specifying movement along axes is executed. This continues until G67 is issued to cancel a modal call.

```
G66  P p  L  ℓ  <argument–specification> ;

  P :          Number of the program to call
  ℓ :          Repetition count (1 by default)
  Argument :   Data passed to the macro
```

```
O0001 ;                          O9100 ;
  :                                :
G66 P9100 L2 A1.0 B2.0 ;         G00 Z–#1 ;
G00 G90 X100.0 ;                 G01 Z–#2 F300 ;
Y200.0 ;                           :
X150.0 Y300.0 ;                    :
G67 ;                              :
  :                                :
M30 ;                            M99 ;
```

Explanations

- **Call**

· After G66, specify at address P a program number subject to a modal call.

· When a number of repetitions is required, a number from 1 to 9999 can be specified at address L.

· As with a simple call (G65), data passed to a macro program is specified in arguments.

- **Cancellation**

When a G67 code is specified, modal macro calls are no longer performed in subsequent blocks.

- **Call nesting**

Calls can be nested to a depth of four levels including simple calls (G65) and modal calls (G66). This does not include subprogram calls (M98).

- **Modal call nesting**

Modal calls can be nested by specifying another G66 code during a modal call.

Limitations

· In a G66 block, no macros can be called.

· G66 needs to be specified before any arguments.

· No macros can be called in a block which contains a code such as a miscellaneous function that does not involve movement along an axis.

· Local variables (arguments) can only be set in G66 blocks. Note that local variables are not set each time a modal call is performed.

Sample program

The same operation as the drilling canned cycle is created using a custom macro and the machining program makes a modal macro call. For program simplicity, all drilling data is specified using absolute values.

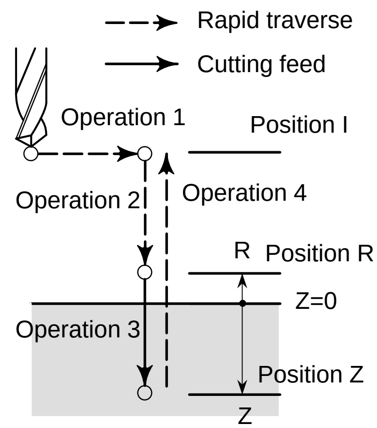
The canned cycle consists of the following basic operations:

Operation 1:
Positioning along the X–axis and Y–axis

Operation 2:
Rapid traverse to point R

Operation 3:
Cutting feed to point Z

Operation 4:
Rapid traverse to point R or I



• Calling format

```
G65 P9110 Xx Yy Zz Rr Ff Ll;
```

X: X coordinate of the hole (absolute specification only) ...... (#24)
Y: Y coordinate of the hole (absolute specification only) ...... (#25)
Z: Coordinates of position Z (absolute specification only) ..... (#26)
R: Coordinates of position R (absolute specification only) ..... (#18)
F: Cutting feedrate .... (#9)
L: Repetition count

• Program that calls a macro program

```
O0001;
G28 G91 X0 Y0 Z0;
G92 X0 Y0 Z50.0;
G00 G90 X100.0 Y50.0;
G66 P9110 Z–20.0 R5.0 F500;
G90 X20.0 Y20.0;
X50.0;
Y50.0;
X70.0 Y80.0;
G67;
M30;
```

• Macro program (program called)

```
O9110;
  #1=#4001; ................................ Stores G00/G01.
  #3=#4003; ................................ Stores G90/G91.
  #4=#4109; ......................... Stores the cutting feedrate.
  #5=#5003; .......... Stores the Z coordinate at the start of drilling.
  G00 G90 Z#18; ...................... Positioning at position R
  G01 Z#26 F#9; ...................... Cutting feed to position Z
  IF[#4010 EQ 98]GOTO 1; ................... Return to position I
  G00 Z#18; .......................... Positioning at position R
  GOTO 2;
N1 G00 Z#5; .......................... Positioning at position I
N2 G#1 G#3 F#4; .................... Restores modal information.
  M99;
```

12.6.3 Macro Call Using G Code

By setting a G code number used to call a macro program in a parameter, the macro program can be called in the same way as for a simple call (G65).

```
O0001 ;                        O9010 ;
   :                              :
G81 X10.0 Y20.0 Z–10.0 ;          :
   :                              :
M30 ;                          N9 M99 ;
Parameter 6050 = 81
```

Explanations

By setting a G code number from 1 to 255 used to call a custom macro program (9010 to 9019) in the corresponding parameter (6050 to 6059), the macro program can be called in the same way as with G65.
For example, when a parameter is set so that macro program O9010 can be called with G81, a user–specific cycle created using a custom macro can be called without modifying the machining program.

- **Correspondence between parameter numbers and program numbers**

Program number	Parameter number
O9010	6050
O9011	6051
O9012	6052
O9013	6053
O9014	6054
O9015	6055
O9016	6056
O9017	6057
O9018	6058
O9019	6059

- **Repetition**

As with a simple call, a number of repetitions from 1 to 9999 can be specified at address L.

- **Argument specification**

As with a simple call, two types of argument specification are available: Argument specification I and argument specification II. The type of argument specification is determined automatically according to the addresses used.

Limitations

- **Nesting of calls using G codes**

In a program called with a G code, no macros can be called using a G code. A G code in such a program is treated as an ordinary G code. In a program called as a subprogram with an M code, no macros can be called using a G code. A G code in such a program is also treated as an ordinary G code.

12.6.4 Macro Call Using an M Code

By setting an M code number used to call a macro program in a parameter, the macro program can be called in the same way as with a simple call (G65).

```
O0001 ;
  :
M50 A1.0 B2.0 ;
  :
M30 ;
```

```
O9020 ;
  :
  :
  :
M99 ;
```

Parameter 6080 = 50

Explanations

By setting an M code number from 1 to 255 used to call a custom macro program (9020 to 9029) in the corresponding parameter (6080 to 6089), the macro program can be called in the same way as with G65.

- **Correspondence between parameter numbers and program numbers**

Program number	Parameter number
O9020	6080
O9021	6081
O9022	6082
O9023	6083
O9024	6084
O9025	6085
O9026	6086
O9027	6087
O9028	6088
O9029	6089

- **Repetition**

As with a simple call, a number of repetitions from 1 to 9999 can be specified at address L.

- **Argument specification**

As with a simple call, two types of argument specification are available: Argument specification I and argument specification II. The type of argument specification is determined automatically according to the addresses used.

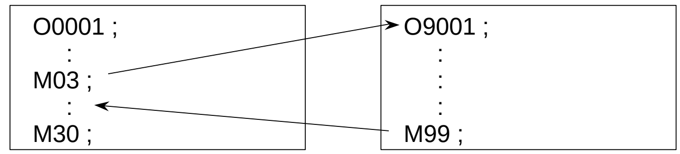
Limitations

– An M code used to call a macro program must be specified at the start of a block.

– In a macro called with a G code or in a program called as a subprogram with an M code, no macros can be called using an M code. An M code in such a macro or program is treated as an ordinary M code.

12.6.5 Subprogram Call Using an M Code

By setting an M code number used to call a subprogram (macro program) in a parameter, the macro program can be called in the same way as with a subprogram call (M98).



Parameter 6071 = 03

Explanations

By setting an M code number from 1 to 255 used to call a subprogram in a parameter (6071 to 6079), the corresponding custom macro program (9001 to 9009) can be called in the same way as with M98.

- **Correspondence between parameter numbers and program numbers**

Program number	Parameter number
O9001	6071
O9002	6072
O9003	6073
O9004	6074
O9005	6075
O9006	6076
O9007	6077
O9008	6078
O9009	6079

- **Repetition**

As with a simple call, a number of repetitions from 1 to 9999 can be specified at address L.

- **Argument specification**

Argument specification is not allowed.

- **M code**

An M code in a macro program that has been called is treated as an ordinary M code.

Limitations

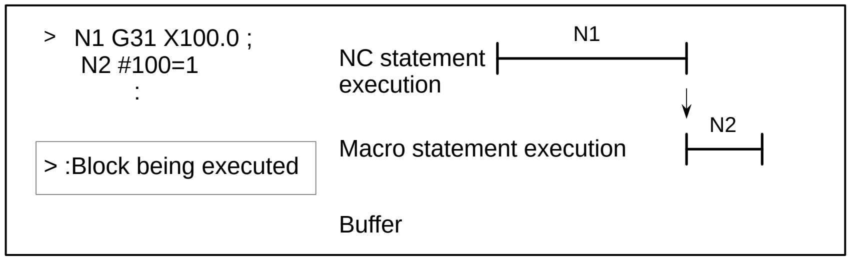
In a macro called with a G code or in a program called with an M code, no subprograms can be called using an M code. An M code in such a macro or program is treated as an ordinary M code.

12.7 PROCESSING MACRO STATEMENTS

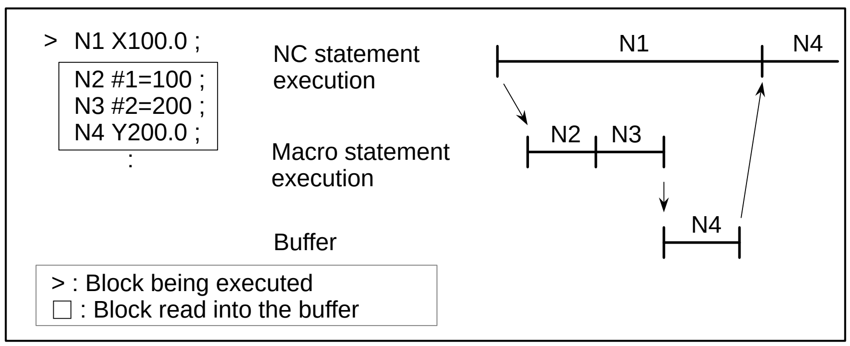
For smooth operation, the NC prereads the NC statement to be performed next. This operation is referred to as buffering. Macro statements for arithmetic expressions and conditional branches are processed as soon as they are read into the buffer. Blocks containing M00, M01, M02, or M30, blocks containing M codes for which buffering is suppressed by setting parameters 3411 to 3420, and blocks containing G31 are not preread.

Explanations

- **When the next block is not buffered (M codes that are not buffered, G31, etc.)**



- **Buffering the next block (normally prereading one block)**



When N1 is being executed, the next NC statement (N4) is read into the buffer. The macro statements (N2, N3) between N1 and N4 are processed during execution of N1.

12.8 REGISTERING CUSTOM MACRO PROGRAMS

Custom macro programs are similar to subprograms. They can be registered and edited in the same way as subprograms. The storage capacity is determined by the total length of tape used to store both custom macros and subprograms.

12.9 LIMITATIONS

- **MDI operation**

The macro call command can be specified in MDI mode. During automatic operation, however, it is impossible to switch to the MDI mode for a macro program call.

- **Sequence number search**

A custom macro program cannot be searched for a sequence number.

- **Single block**

Even while a macro program is being executed, blocks can be stopped in the single block mode (except blocks containing macro call commands, arithmetic operation commands, and control commands).
A block containing a macro call command (G65, G66, or G67) does not stop even when the single block mode is on. Blocks containing arithmetic operation commands and control commands can be stopped in single block mode by setting SBKM (bit 5 of parameter 6000) to 1.
Single block stop operation is used for testing custom macro programs. When SBKM (bit 5 of parameter 6000) is set to 1, a single block stop takes place at every macro statement. (Strictly speaking, the block is regarded as specifying a movement with a travel distance 0.)

- **Optional block skip**

A / appearing in the middle of an <expression> (enclosed in brackets [ ] on the right–hand side of an arithmetic expression) is regarded as a division operator; it is not regarded as the specifier for an optional block skip code.

- **Operation in EDIT mode**

Registered custom macro programs and subprograms should be protected from being destroyed by accident. By setting NE8 (bit 0 of parameter 3202) and NE9 (bit 4 of parameter 3202) to 1, deletion and editing are disabled for custom macro programs and subprograms with program numbers 8000 to 8999 and 9000 to 9999. When the entire memory is cleared (by pressing the RESET and DELETE keys at the same time to turn on the power), the contents of memory such as custom macro programs are deleted.

- **Reset**

When memory is cleared with a reset operation, local variables and common variables #100 to #199 are cleared to null values. They can be prevented from being cleared by setting, CLV and CCV (bits 7 and 6 of parameter 6001). System variables #1000 to #1133 are not cleared.
A reset operation clears any called states of custom macro programs and subprograms, and any DO states, and returns control to the main program.

- **Display of the PROGRAM RESTART page**

As with M98, the M codes used for subprogram calls are not displayed.

- **Feed hold**

When a feed hold is enabled during execution of a macro statement, the machine stops after execution of the macro statement. The machine also stops when a reset or alarm occurs.

- **Constant values that can be used in <expression>**

+0.0000001 to +99999999
–99999999 to –0.0000001
The number of significant digits is 8 (decimal). If this range is exceeded, alarm No. 003 occurs.

12.10 EXTERNAL OUTPUT COMMANDS

In addition to the standard custom macro commands, the following macro commands are available. They are referred to as external output commands.

– **BPRNT**
– **DPRNT**
– **POPEN**
– **PCLOS**

These commands are provided to output variable values and characters through the reader/punch interface.

Explanations

Specify these commands in the following order:

Open command: POPEN

Before specifying a sequence of data output commands, specify this command to establish a connection to an external input/output device.

Data output command: BPRNT or DPRNT

Specify necessary data output.

Close command: PCLOS

When all data output commands have completed, specify PCLOS to release a connection to an external input/output device.

• Open command POPEN

POPEN

POPEN establishes a connection to an external input/output device. It must be specified before a sequence of data output commands. The NC outputs a DC2 control code.

• Data output command BPRNT

BPRNT [a #b [c] …]

Number of significant decimal places
Variable
Character

The BPRNT command outputs characters and variable values in binary.

(i) Specified characters are converted to corresponding ISO codes according to the setting (ISO) that is output at that time.
Specifiable characters are as follows:
– **Letters (A to Z)**
– **Numbers**
– **Special characters (*, /, +, –, etc.)**
An asterisk (*) is output by a space code.

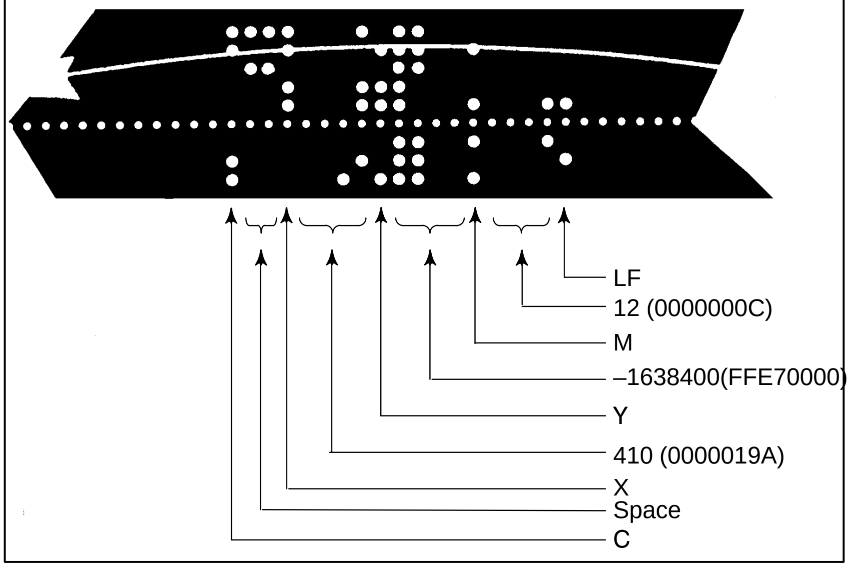
(ii) All variables are stored with a decimal point. Specify a variable followed by the number of significant decimal places enclosed in brackets. A variable value is treated as 2–word (32–bit) data, including the decimal digits. It is output as binary data starting from the highest byte.

(iii) When specified data has been output, an EOB code is output according to the ISO code settings on the parameter screen.

(iv) Null variables are regarded as 0.

Example)

BPRINT [C** X#100 [3] Y#101 [3] M#10 [0]]
Variable value
#100=0.40596
#101=–1638.4
#10=12.34



- **Data output command DPRNT**

DPRNT [a #b [c d] …]

- Number of significant decimal places (d)
- Number of significant digits in the integer part (c)
- Variable (#b)
- Character (a)

The DPRNT command outputs characters and each digit in the value of a variable according to the code set in the settings (ISO).

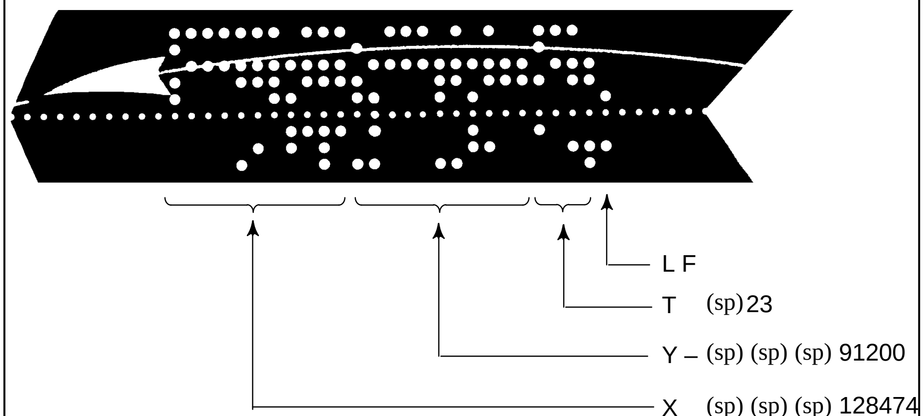
(i) For an explanation of the DPRNT command, see Items (i), (iii), and (iv) for the BPRNT command.

(ii) When outputting a variable, specify # followed by the variable number, then specify the number of digits in the integer part and the number of decimal places enclosed in brackets.
One code is output for each of the specified number of digits, starting with the highest digit. For each digit, a code is output according to the settings (ISO). The decimal point is also output using a code set in the settings (ISO).
Each variable must be a numeric value consisting of up to eight digits. When high–order digits are zeros, these zeros are not output if PRT (bit1 of parameter 6001) is 1. If PRT is 0, a space code is output each time a zero is encountered.
When the number of decimal places is not zero, digits in the decimal part are always output. If the number of decimal places is zero, no decimal point is output.
When PRT (bit 1 of parameter 6001) is 0, a space code is output to indicate a positive number instead of +; if PRT is 1, no code is output.

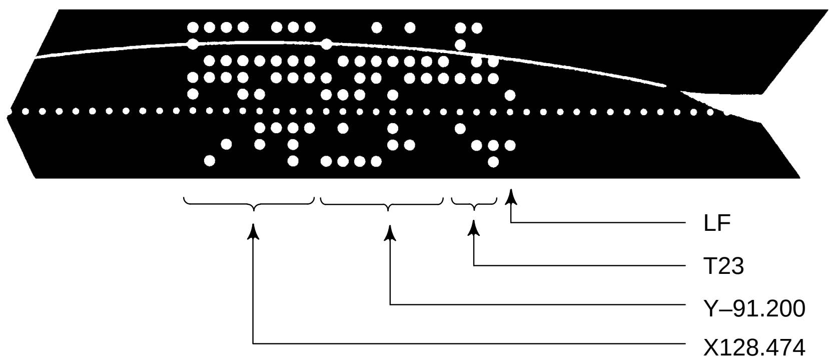
Example)

DPRINT [X#2 [53] Y#5 [53] T#30 [20]]
Variable value
#2=128.47398
#5=–91.2
#30=123.456

(1) Parameter PRT(No.6001#1)=0



(2) Parameter PRT(No.6001#1)=0



- **Close command PCLOS**

PCLOS ;

The PCLOS command releases a connection to an external input/output device. Specify this command when all data output commands have terminated. DC4 control code is output from the controller.

• Required setting

Specify the channel use for parameter 020. According to the specification of this parameter, set data items (such as the baud rate) for the reader/punch interface.

I/O channel 0 : Parameters 101 and 103
I/O channel 1 : Parameters 111 and 113

Specify parameter 102 or 112 so that the reader/punch interface is used as the output device for punching. (Never specify output to the Fanuc Cassette or floppy disks.)

When specifying a DPRNT command to output data, specify whether leading zeros are output as spaces (by setting PRT (bit 1 of parameter 6001) to 1 or 0).

To indicate the end of a line of data in ISO code, specify whether to use only an LF (NCR, of bit 3 of parameter 0103 is 0) or an LF and CR (NCR is 1).

NOTE

1. It is not necessary to always specify the open command (POPEN), data output command (BPRNT, DPRNT), and close command (PCLOS) together. Once an open command is specified at the beginning of a program, it does not need to be specified again except after a close command was specified.
2. Be sure to specify open commands and close commands in pairs. Specify the close command at the end of the program. However, do not specify a close command if no open command has been specified.
3. When a reset operation is performed while commands are being output by a data output command, output is stopped and subsequent data is erased. Therefore, when a reset operation is performed by a code such as M30 at the end of a program that performs data output, specify a close command at the end of the program so that processing such as M30 is not performed until all data is output.
4. Abbreviated macro words enclosed in brackets [ ] remains unchanged. However, note that when the characters in brackets are divided and input several times, the second and subsequent abbreviations are converted and input.
5. O can be specified in brackets [ ]. Note that when the characters in brackets [ ] are divided and input several times, O is omitted in the second and subsequent inputs.

12.11 INTERRUPTION TYPE CUSTOM MACRO

When a program is being executed, another program can be called by inputting an interrupt signal (UINT) from the machine. This function is referred to as an interruption type custom macro function. Program an interrupt command in the following format:

Format

M96 P○○○○ ;	Enables custom macro interrupt
M97 ;	Disables custom macro interrupt

Explanations

Use of the interruption type custom macro function allows the user to call a program during execution of an arbitrary block of another program. This allows programs to be operated to match situations which vary from time to time.

(1) When a tool abnormality is detected, processing to handle the abnormality is started by an external signal.

(2) A sequence of operation operations is interrupted by another operation operation without the cancellation of the current operation.

(3) At regular intervals, information on current operation is read.
Listed above are examples like adaptive control applications of the interruption type custom macro function.

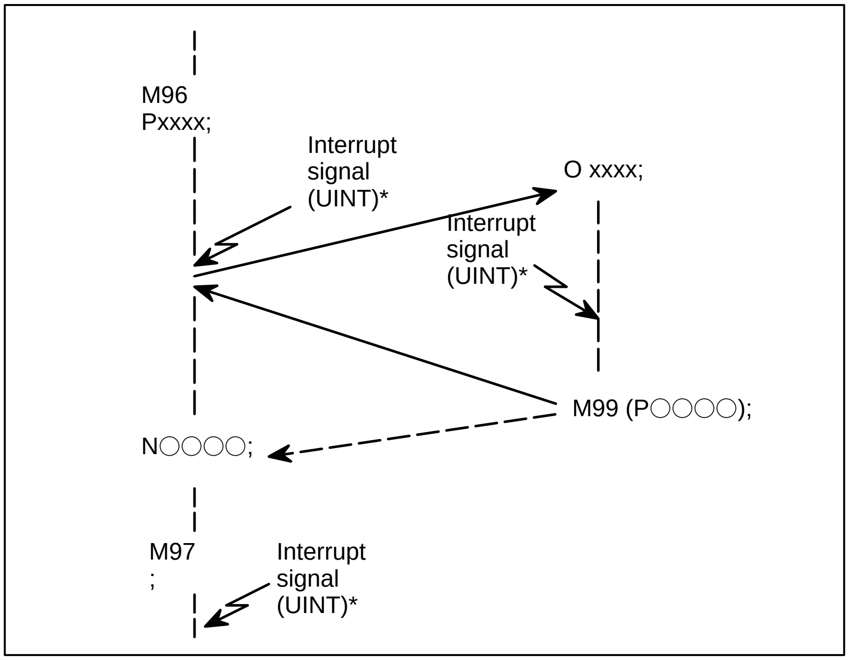


Fig 12.11 Interruption type sustom macro function

When M96Pxxxx is specified in a program, subsequent program operation can be interrupted by an interrupt signal (UINT) input to execute the program specified by Pxxxx.
When the interrupt signal (UINT, marked by * in Fig. NO TAG is input during execution of the interrupt program or after M97 is specified, it is ignored.

12.11.1 Specification Method

Explanations

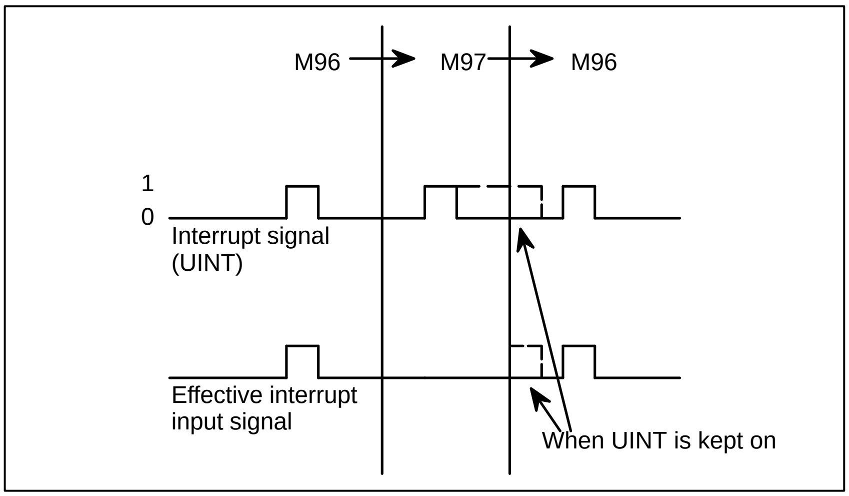
- **Interrupt conditions**

A custom macro interrupt is available only during program execution. It is enabled under the following conditions

– When memory operation or MDI operation is selected
– When STL (start lamp) is on
– When a custom macro interrupt is not currently being processed

- **Specification**

Generally, the custom macro interrupt function is used by specifying M96 to enable the interrupt signal (UINT) and M97 to disable the signal.
Once M96 is specified, a custom macro interrupt can be initiated by the input of the interrupt signal (UINT) until M97 is specified or the controller is reset. After M97 is specified or the NC is reset, no custom macro interrupts are initiated even when the interrupt signal (UINT) is input. The interrupt signal (UINT) is ignored until another M96 command is specified.



The interrupt signal (UINT) becomes valid after M96 is specified. Even when the signal is input in M97 mode, it is ignored. When the signal input in M97 mode is kept on until M96 is specified, a custom macro interrupt is initiated as soon as M96 is specified (only when the status–triggered scheme is employed); when the edge–triggered scheme is employed, the custom macro interrupt is not initiated even when M96 is specified.

NOTE
For the status–triggered and edge–triggered schemes, see Item"Custom macro interrupt signal (UINT)" of Subsec. 12.11.2.

12.11.2 Details of Functions

Explanations

- **Subprogram–type interrupt and macro–type interrupt**

There are two types of custom macro interrupts: Subprogram–type interrupts and macro–type interrupts. The interrupt type used is selected by MSB (bit 5 of parameter 6003).

(a) Subprogram–type interrupt
An interrupt program is called as a subprogram. This means that the levels of local variables remain unchanged before and after the interrupt. This interrupt is not included in the nesting level of subprogram calls.

(b) Macro–type interrupt
An interrupt program is called as a custom macro. This means that the levels of local variables change before and after the interrupt. The interrupt is not included in the nesting level of custom macro calls. When a subprogram call or a custom macro call is performed within the interrupt program, this call is included in the nesting level of subprogram calls or custom macro calls. Arguments cannot be passed from the current program even when the custom macro interrupt is a macro–type interrupt.

- **M codes for custom macro interrupt control**

In general, custom macro interrupts are controlled by M96 and M97. However, these M codes, may already being used for other purposes (such as an M function or macro M code call) by some machine tool builders. For this reason, MPR (bit 4 of parameter 6003) is provided to set M codes for custom macro interrupt control.
When specifying this parameter to use the custom macro interrupt control M codes set by parameters, set parameters 6033 and 6034 as follows:
Set the M code to enable custom macro interrupts in parameter 6033, and set the M code to disable custom macro interrupts in parameter 6034.
When specifying that parameter–set M codes are not used, M96 and M97 are used as the custom macro control M codes regardless of the settings of parameters 6033 and 6034.
The M codes used for custom macro interrupt control are processed internally (they are not output to external units). However, in terms of program compatibility, it is undesirable to use M codes other than M96 and M97 to control custom macro interrupts.

- **Custom macro interrupts and NC statements**

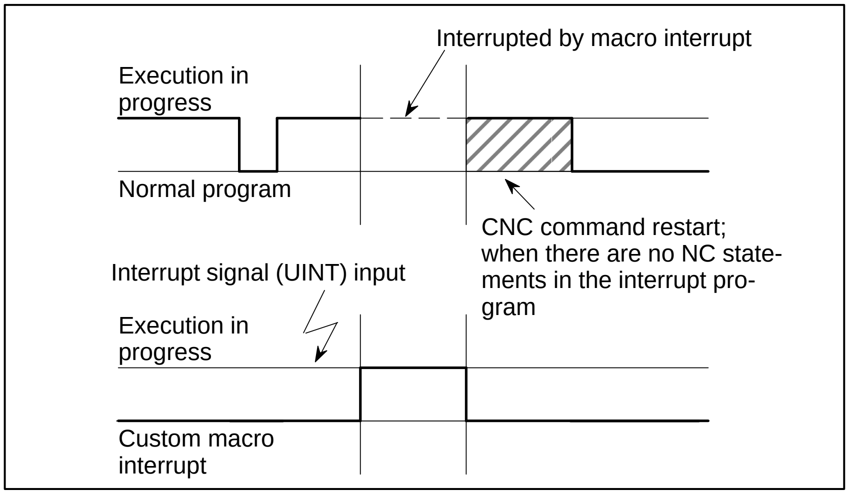
When performing a custom macro interrupt, the user may want to interrupt the NC statement being executed, or the user may not want to perform the interrupt until the execution of the current block is completed. MIN (bit 2 of parameter 6003)is used to select whether to perform interrupts even in the middle of a block or to wait until the end of the block.

- **Type I (when an interrupt is performed even in the middle of a block)**

(i) When the interrupt signal (UINT) is input, any movement or dwell being performed is stopped immediately and the interrupt program is executed.

(ii) If there are NC statements in the interrupt program, the command in the interrupted block is lost and the NC statement in the interrupt program is executed. When control is returned to the interrupted program, the program is restarted from the next block after the interrupted block.

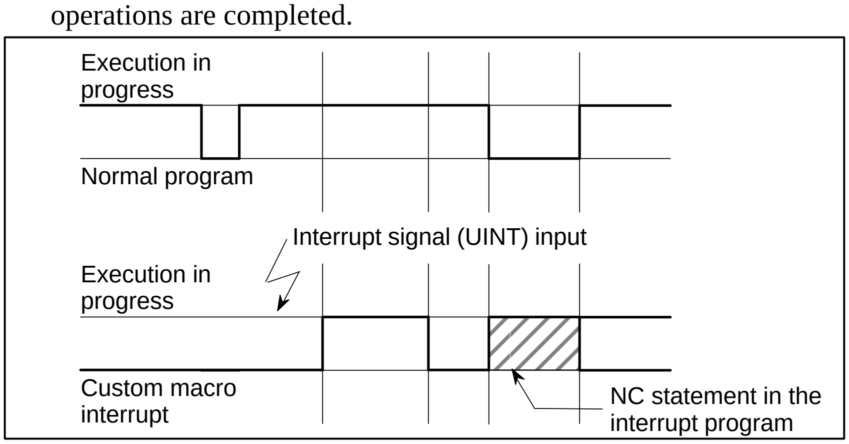
(iii)If there are no NC statements in the interrupt program, control is returned to the interrupted program by M99, then the program is restarted from the command in the interrupted block.



- **Type II (when an interrupt is performed at the end of the block)**

(i)If the block being executed is not a block that consists of several cycle operations such as automatic reference position return (G28), an interrupt is performed as follows:
When an interrupt signal (UINT) is input, macro statements in the interrupt program are executed immediately unless an NC statement is encountered in the interrupt program. NC statements are not executed until the current block is completed.

(ii) If the block being executed consists of several cycle operations, an interrupt is performed as follows:
When the last movement in the cycle operations is started, macro statements in the interrupt program are executed unless an NC statement is encountered. NC statements are executed after all cycle operations are completed.



- **Conditions for enabling and disabling the custom macro interrupt signal**

The interrupt signal becomes valid after execution starts of a block that contains M96 for enabling custom macro interrupts. The signal becomes invalid when execution starts of a block that contains M97.

While an interrupt program is being executed, the interrupt signal becomes invalid. The signal become valid when the execution of the block that immediately follows the interrupted block in the main program is started after control returns from the interrupt program. In type I, if the interrupt program consists of only macro statements, the interrupt signal becomes valid when execution of the interrupted block is started after control returns from the interrupt program.

- **Custom macro interrupt during execution of a block that involves cycle operation**

- **For type I**

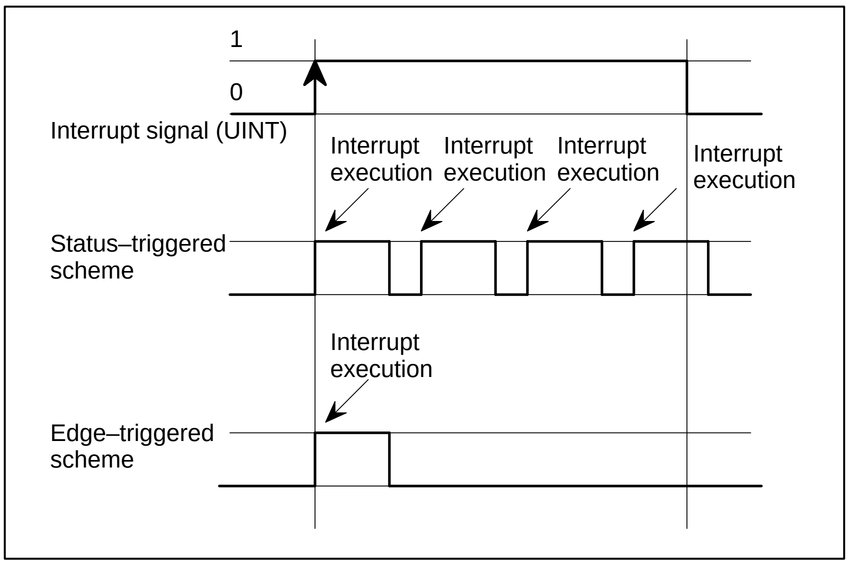
Even when cycle operation is in progress, movement is interrupted, and the interrupt program is executed. If the interrupt program contains no NC statements, the cycle operation is restarted after control is returned to the interrupted program. If there are NC statements, the remaining operations in the interrupted cycle are discarded, and the next block is executed.

- **For type II**

When the last movement of the cycle operation is started, macro statements in the interrupt program are executed unless an NC statement is encountered. NC statements are executed after cycle operation is completed.

- **Custom macro interrupt signal (UINT)**

There are two schemes for custom macro interrupt signal (UINT) input: The status–triggered scheme and edge– triggered scheme. When the status–triggered scheme is used, the signal is valid when it is on. When the edge triggered scheme is used, the signal becomes valid on the rising edge when it switches from off to on status.
One of the two schemes is selected with TSE (bit 3 of parameter 6003). When the status–triggered scheme is selected by this parameter, a custom macro interrupt is generated if the interrupt signal (UINT) is on at the time the signal becomes valid. By keeping the interrupt signal (UINT) on, the interrupt program can be executed repeatedly.
When the edge–triggered scheme is selected, the interrupt signal (UINT) becomes valid only on its rising edge. Therefore, the interrupt program is executed only momentarily (in cases when the program consists of only macro statements). When the status–triggered scheme is inappropriate, or when a custom macro interrupt is to be performed just once for the entire program (in this case, the interrupt signal may be kept on), the edge–triggered scheme is useful.
Except for the specific applications mentioned above, use of either scheme results in the same effects. The time from signal input until a custom macro interrupt is executed does not vary between the two schemes.



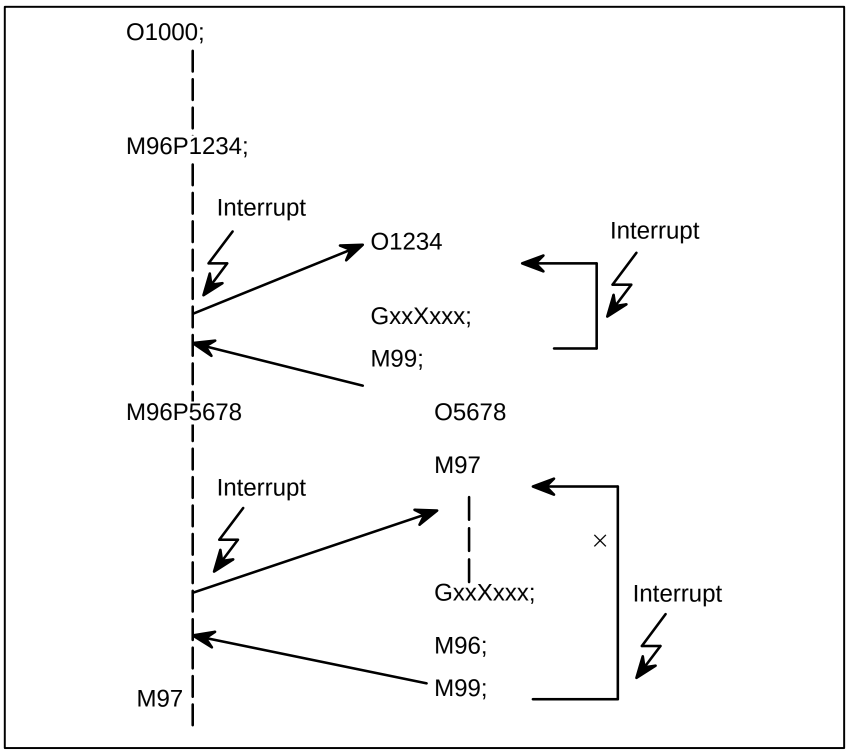
In the above example, an interrupt is executed four times when the status triggered scheme is used; when the edge– triggered scheme is used, the interrupt is executed just once.

- **Return from a custom macro interrupt**

To return control from a custom macro interrupt to the interrupted program, specify M99. A sequence number in the interrupted program can also be specified using address P. If this is specified, the program is searched from the beginning for the specified sequence number. Control is returned to the first sequence number found.

When a custom macro interrupt program is being executed, no interrupts are generated. To enable another interrupt, execute M99. When M99 is specified alone, it is executed before the preceding commands terminate. Therefore, a custom macro interrupt is enabled for the last command of the interrupt program. If this is inconvenient, custom macro interrupts should be controlled by specifying M96 and M97 in the program.

When a custom macro interrupt is being executed, no other custom macro interrupts are generated; when an interrupt is generated, additional interrupts are inhibited automatically. Executing M99 makes it possible for another custom macro interrupt to occur. M99 specified alone in a block is executed before the previous block terminates. In the following example, an interrupt is enabled for the Gxx block of O1234. When the signal is input, O1234 is executed again. O5678 is controlled by M96 and M97. In this case, an interrupt is not enabled for O5678 (enabled after control is returned to O1000).



NOTE

When an M99 block consists only of address O, N, P, L, or M, this block is regarded as belonging to the previous block in the program. Therefore, a single–block stop does not occur for this block. In terms of programming, the following (1) and (2) are basically the same. (The difference is whether G○○ is executed before M99 is recognized.)

(1) G○○ X○○○ ;
M99 ;

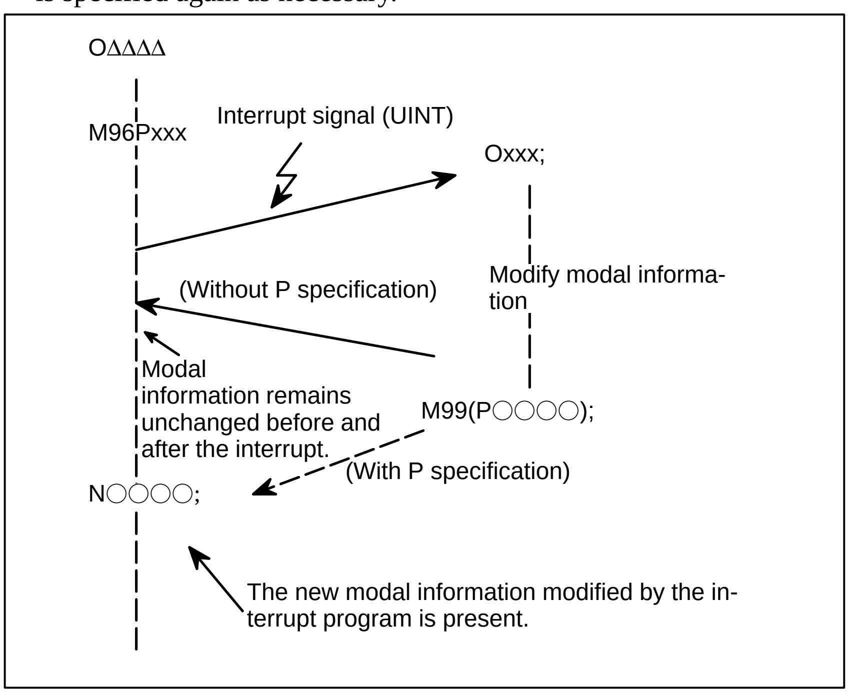
(2)G○○ X○○○ M99 ;

- **Custom macro interrupt and modal information**

A custom macro interrupt is different from a normal program call. It is initiated by an interrupt signal (UINT) during program execution. In general, any modifications of modal information made by the interrupt program should not affect the interrupted program.
For this reason, even when modal information is modified by the interrupt program, the modal information before the interrupt is restored when control is returned to the interrupted program by M99.
When control is returned from the interrupt program to the interrupted program by M99 Pxxxx, modal information can again be controlled by the program. In this case, the new continuous information modified by the interrupt program is passed to the interrupted program. Restoration of the old modal information present before the interrupt is not desirable. This is because after control is returned, some programs may operate differently depending on the modal information present before the interrupt. In this case, the following measures are applicable:

(1)The interrupt program provides modal information to be used after control is returned to the interrupted program.

(2)After control is returned to the interrupted program, modal information is specified again as necessary.



- **Modal information when control is returned by M99**

The modal information present before the interrupt becomes valid. The new modal information modified by the interrupt program is made invalid.

- **Modal information when control is returned by M99 P○○○○**

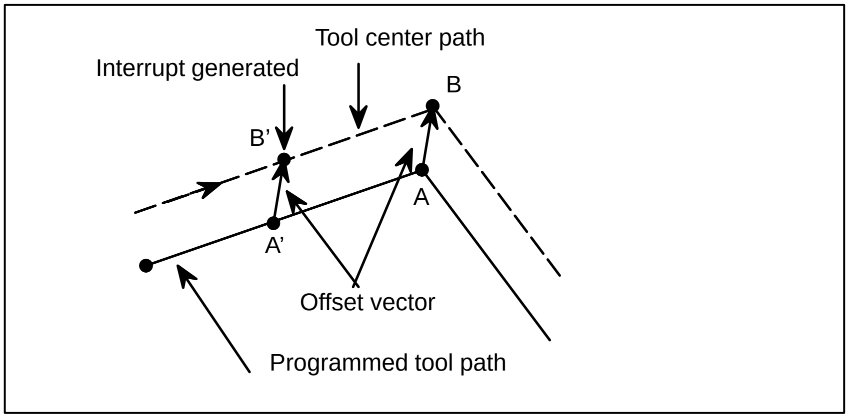
The new modal information modified by the interrupt program remains valid even after control is returned. The old modal information which was valid in the interrupted block can be read using custom macro system variables #4001 to #4120.
Note that when modal information is modified by the interrupt program, system variables #4001 to #4120 are not changed.

- **System variables (position information values) for the interrupt program**

· The coordinates of point A can be read using system variables #5001 and up until the first NC statement is encountered.

· The coordinates of point A' can be read after an NC statement with no move specifications appears.

· The machine coordinates and workpiece coordinates of point B' can be read using system variables #5021 and up and #5041 and up.



- **Custom macro interrupt and custom macro modal call**

When the interrupt signal (UINT) is input and an interrupt program is called, the custom macro modal call is canceled (G67). However, when G66 is specified in the interrupt program, the custom macro modal call becomes valid. When control is returned from the interrupt program by M99, the modal call is restored to the state it was in before the interrupt was generated. When control is returned by M99Pxxxx;, the modal call in the interrupt program remains valid.

13 PATTERN DATA INPUT FUNCTION

This function enables users to perform programming simply by extracting numeric data (pattern data) from a drawing and specifying the numerical values from the CRT/MDI panel.
This eliminates the need for programming using an existing NC language.

With the aid of this function, a machine tool builder can prepare the program of a hole machining cycle (such as a boring cycle or tapping cycle) using the custom macro function, and can store it into the program memory.
This cycle is assigned pattern names, such as BOR1, TAP3, and DRL2.

An operator can select a pattern from the menu of pattern names displayed on the screen.
Data (pattern data) which is to be specified by the operator should be created in advance with variables in a drilling cycle.
The operator can identify these variables using names such as DEPTH, RETURN RELIEF, FEED, MATERIAL or other pattern data names. The operator assigns values (pattern data) to these names.

13.1 DISPLAYING THE PATTERN MENU

Pressing the OFFSET SETTING key and ▷ **[MENU]** is displayed on the following pattern menu screen.

```
MENU : HOLE PATTERN                  O0000 N00000
     1.  BOLT HOLE
     2.  GRID
     3.  LINE ANGLE
     4.  TAPPING
     5.  DRILLING
     6.  BORING
     7.  POCKET
     8.  PECK
     9.  TEST PATRN
    10.  BACK

> _
MDI **** *** ***
[ MACRO ] [  MENU  ] [        ] [        ] [ (OPRT) ]
```

HOLE PATTERN : This is the menu title. An arbitrary character string consisting of up to 12 characters can be specified.

BOLT HOLE : This is the pattern name. An arbitrary character string consisting of up to 10 characters can be specified, including katakana.

The machine tool builder should specify the character strings for the menu title and pattern name using the custom macro, and load the character strings into program memory as a subprogram of program No. 9500.

- **Macro commands specifying the menu title**

Menu title : C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9C_{10} C_{11} C_{12}

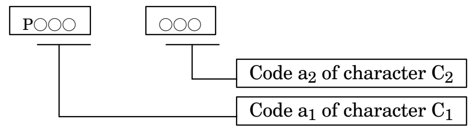
C_1,C_2, ,C_{12} : Characters in the menu title (12 characters)

Macro instruction

G65 H90 P_p Q_q R_r I_i J_j K_k ;

H90 : Specifies the menu title

p : Assume a1 and a2 to be the codes of characters C1 and C2. Then,



q : Assume a_3 and a_4 to be the codes of characters C_3 and C_4. Then, $q=a_3\ 10^3+a_4$

r : Assume a_5 and a_6 to be the codes of characters C_5 and C_6. Then, $r=a_5\ 10^3+a_6$

i : Assume a_7 and a_8 to be the codes of characters C_7 and C_8. Then, $i=a_7\ 10^3+a_8$

j : Assume a_9 and a_{10} to be the codes of characters C_9 and C_{10}. Then, $j=a_9\ 10^3+a_{10}$

k : Assume a_{11} and a_{12} to be the codes of characters C_{11} and C_{12}.Then, $k=a_{11}\ 10^3+a_{12}$

Example) If the title of the menu is”HOLE PATTERN” then the macro instruction is as follows:

G65 H90 P072079 Q076069 R032080
HO LE ␣P
I065084 J084069 K082078;
AT TE RN

For codes corresponding to these characters, refer to the table in 13.3.

- **Macro instruction describing the pattern name**

Pattern name: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 $C_9 C_{10}$
$C_{1,}$ $C_{2,}$,C_{10}: Characters in the pattern name (10 characters)
Macro instruction

G65 H91 P_n Q_q R_r I_i J_j K_k ;

H91 : Specifies the menu title

n : Specifies the menu No. of the pattern name
n=1 to 10

q : Assume a_1 and a_2 to be the codes of characters C_1 and C_2. Then, $q=a_1 \times 10^3+a_2$

r : Assume a_3 and a_4 to be the codes of characters C_3 and C_4. Then, $r=a_3 \times 10^3+a_4$

i : Assume a_5 and a_6 to be the codes of characters C_5 and C_6. Then, $i=a_5 \times 10^3+a_6$

j : Assume a_7 and a_8 to be the codes of characters C_7 and C_8. Then, $j=a_7 \times 10^3+a_8$

k : Assume a_9 and a_{10} to be the codes of characters C_9 and C_{10}. Then, $k=a_9 \times 10^3+a_{10}$

Example) If the pattern name of menu No. 1 is ”BOLT HOLE” then the macro instruction is as follows.

```
G65 H91 P1 Q066079 R076084 I032072 J079076 K069032 ;
              BO      LT     ␣H      OL      E␣
```

- **Pattern No. selection**

To select a pattern from the pattern menu screen, enter the corresponding pattern No. The following is an example.

[1] [INPUT]

The selected pattern No. is assigned to system variable #5900. The custom macro of the selected pattern can be started by starting a fixed program (external program No. search) with an external signal then referring to the system variable #5900 in the program.

> **NOTE**
> If each characters of P, Q, R, I, J, and K are not specified in a macro instruction, two spaces are assigned to each omitted character.

Example

Custom macros for the menu title and hole pattern names.

```
MENU : HOLE PATTERN                O0000 N00000
     1.  BOLT HOLE
     2.  GRID
     3.  LINE ANGLE
     4.  TAPPING
     5.  DRILLING
     6.  BORING
     7.  POCKET
     8.  PECK
     9.  TEST PATRN
    10.  BACK

> _
MDI **** *** ***
[ MACRO ] [ MENU ] [        ] [        ] [ (OPRT) ]
```

O9500 ;

N1G65 H90 P072 079 Q076 069 R032 080 I 065 084 J 084 069 K082 078;	HOLE PATTERN
N2G65 H91 P1 Q066 079 R076 084 I 032 072 J 079 076 K069 032 ;	1.BOLT HOLE
N3G65 H91 P2 Q071 082 R073 068 ;	2.GRID
N4G65 H91 P3 Q076 073 R078 069 I 032 065 J 078071 K076069	3.LINE ANGLE
N5G65 H91 P4 Q084 065 R080 080 I 073 078 J 071 032 ;	4.TAPPING
N6G65 H91 P5 Q068 082 R073 076 I 076 073 J 078 071 ;	5.DRILLING
N7G65 H91 P6 Q066079 R082073 I 078 071 ;	6.BORING
N8G65 H91 P7 Q080 079 R067 075 I 069 084 ;	7.POCKET
N9G65 H91 P8 Q080069 R067075 ;	8.PECK
N10G65 H91 P9 Q084 069 R083 084 I032 080 J065 084 K082 078 ;	9.TEST PATRN
N11G65 H91 P10 Q066 065 R067 075 ;	10.BACK
N12M99 ;	

13.2 PATTERN DATA DISPLAY

When a pattern menu is selected, the necessary pattern data is displayed.

```
VAR. : BOLT HOLE                        O0001 N00000
 NO.  NAME           DATA   COMMENT
 500   TOOL          0.000
 501   STANDARD X    0.000 *BOLT HOLE
 502   STANDARD Y    0.000   CIRCLE*
 503   RADIUS        0.000 SET PATTERN
 504   S. ANGL       0.000 DATA TO VAR.
 505   HOLES NO.     0.000 NO.500–505.
 506                 0.000
 507                 0.000
 ACTUAL POSITION (WORK)
      X      0.000    Y      0.000
 > _  Z      0.000
 MDI **** *** ***
 [ MACRO ] [ MENU ] [      ] [     ] [ (OPRT) ]
```

BOLT HOLE : This is the pattern data title. A character string consisting of up to 12 characters can be set.

TOOL : This is the variable name. A character string consisting of up to 10 characters can be set.

BOLT HOLE CIRCLE : This is a comment statement. A character string can be displayed consisting of up to 8 lines, 12 characters per line.

(It is permissible to use katakana in a character string or line.)
The machine tool builder should program the character strings of pattern data title, pattern name, and variable name using the custom macro, and load them into the program memory as a subprogram whose No. is 9500 plus the pattern No. (O9501 to O9510).

- **Macro instruction specifying the pattern data title (the menu title)**

Menu title : $C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7\ C_8\ C_9C_{10}C_{11}C_{12}$
$C_1, C_2, ..., C_{12}$: Characters in the menu title (12 characters)
Macro instruction

G65 H92 $P_n\ Q_q\ R_r\ I_i\ J_j\ K_k$;

H92 : Specifies the pattern name

p : Assume a_1 and a_2 to be the codes of characters C_1 and C_2. Then, $p=a_1 \times 10^3+a_2$
See 13.3 for character codes.

q : Assume a_3 and a_4 to be the codes of characters C_3 and C_4. Then, $q=a_3 \times 10^3+a_4$

r : Assume a_5 and a_6 to be the codes of characters C_5 and C_6. Then, $r=a_5 \times 10^3+a_6$

i : Assume a_7 and a_8 to be the codes of characters C_7 and C_8. Then, $i=a_7 \times 10^3+a_8$

j : Assume a_9 and a_{10} to be the codes of characters C_9 and C_{10}. Then, $j=a_9 \times 10^3+a_{10}$

k : Assume a_{11} and a_{12} to be the codes of characters C_{11} and C_{12}. Then, $k=a_{11} \times 10^3+a_{12}$

Example) Assume that the pattern data title is ”BOLT HOLE.”The macro instruction is given as follows:

G65 H92 P<u>066079</u> Q<u>076084</u> R<u>032072</u> I<u>079076</u> J<u>069032</u>;
BO LT ␣H OL E

- **Macro instruction specifying the variable name**

Variable name : $C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7\ C_8\ C_9C_{10}$
$C_1, C_2, ..., C_{10}$: Characters in the variable name (10 characters)
Macro instruction

G65 H93 $P_n\ Q_q\ R_r\ I_i\ J_j\ K_k$;

H93 : Specifies the variable name

n : Specifies the menu No. of the variable name
n=1 to 10

q : Assume a_1 and a_2 to be the codes of characters C_1 and C_2. Then, $q=a_1 \times 10^3+a_2$

r : Assume a_3 and a_4 to be the codes of characters C_3 and C_4. Then, $r=a_3 \times 10^3+a_4$

i : Assume a_5 and a_6 to be the codes of characters C_5 and C_6. Then, $i=a_5 \times 10^3+a_6$

j : Assume a_7 and a_8 to be the codes of characters C_7 and C_8. Then, $j=a_7 \times 10^3+a_8$

k : Assume a_9 and a_{10} to be the codes of characters C_9 and C_{10}. Then, $k=a_9 \times 10^3a+a_{10}$

Example) Assume that the variable name of the variable No. 503 is ”RADIUS.” The macro instruction is given as follows:

G65 H93 P503 Q<u>082065</u> R<u>068073</u> I<u>085083</u> ;
RA DI US

> **NOTE**
> Variable names can be assigned to 200 common variables #500 to #699, which are not cleared when the power is turned off.

- **Macro instruction to describe a comment**

One comment line: C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12}
C_1, C_2,..., C_{12} : Character string in one comment line (12 characters)
Macro instruction

G65 H94 P_n Q_q R_r I_i J_j K_k ;

H94 : Specifies the comment

p : Assume a_1 and a_2 to be the codes of characters C_1 and C_2. Then, $p=a_1 \times 10^3+a_2$
See 13.3 for character codes.

q : Assume a_3 and a_4 to be the codes of characters C_3 and C_4. Then, $q=a_3 \times 10^3+a_4$

r : Assume a_5 and a_6 to be the codes of characters C_5 and C_6. Then, $r=a_5 \times 10^3+a_6$

i : Assume a_7 and a_8 to be the codes of characters C_7 and C_8. Then, $i=a_7 \times 10^3+a_8$

j : Assume a_9 and a_{10} to be the codes of characters C_9 and C_{10}. Then, $j=a_9 \times 10^3+a_{10}$

k : Assume a_{11} and a_{12} to be the codes of characters C_{11} and C_{12}. Then, $k=a_{11} \times 10^3+a_{12}$

A comment can be displayed in up to eight lines. The comment consists of the first line to the eighth line in the programmed sequence of G65 H94 for each line.

Example) Assume that the comment is ”BOLT HOLE.” The macro instruction is given as follows:

```
G65 H94 P042066 Q079076 R084032 I072079 J076069;
            *B       OL       T␣       HO       LE
```

Examples

Macro instruction to describe a parameter title , the variable name, and a comment.

```
VAR. : BOLT HOLE                          O0001 N00000
  NO.    NAME             DATA     COMMENT
  500     TOOL            0.000
  501     STANDARD X      0.000     *BOLT HOLE
  502     STANDARD Y      0.000       CIRCLE*
  503     RADIUS          0.000     SET PATTERN
  504     S. ANGL         0.000     DATA TO VAR.
  505     HOLES NO.       0.000     NO.500–505.
  506                     0.000
  507                     0.000

 ACTUAL POSITION (WORK)
    X     0.000        Y     0.000
>_  Z     0.000
MDI **** *** ***
[ MACRO ] [ MENU ] [        ] [         ] [ (OPRT) ]
```

O9501 ;

N1G65 H92 P066 079 Q076 084 R032 072 I 079 076 J069 032 ;	VAR : BOLT HOLE
N2G65 H93 P500 Q084 079 R079076 ;	#500 TOOL
N3G65 H93 P501 Q083 084 R065 078 I068 065 J082 068 K032 088 ;	#501 STANDARD X
N4G65 H93 P502 Q083 084 R065 078 I068 065 J082 068 K032 089 ;	#502 STANDARD Y
N5G65 H93 P503 Q082 065 R068 073 I 085 083 ;	#503 RADIUS
N6G65 H93 P504 Q083 046 R032 065 I 078 071 J 076 032 ;	#504 S.ANGL
N7G65 H93 P505 Q072 079 R076 069 I 083 032 J078 079 K046 032 ;	#505 HOLES NO.
N8G65 H94 ;	Comment
N9G65 H94 P042 066 Q079 076 R084 032 I072 079 J076 069 ;	*BOLT HOLE
N10G65 H94 R032 067 I073 082 J067 076 K069 042 ;	CIRCLE*
N11G65 H94 P083 069 Q084 032 080 065 I084 084 J069 082 K078 032 ;	SET PATTERN
N12G65 H94 P068 065 Q084 065 R032 078 I079 032 J086 065 K082046 ;	DATA TO VAR.
N13G65 H94 P078 079 Q046 053 R048 048 I045 053 J048 053 K046 032 ;	NO.500–505.
N14M99 ;	

13.3 CHARACTERS AND CODES TO BE USED FOR THE PATTERN DATA INPUT FUNCTION

Table.13.3(a)
Characters and codes to be used for the pattern data input function

Character	Code	Comment	Character	Code	Comment
A	065		6	054	
B	066		7	055	
C	067		8	056	
D	068		9	057	
E	069			032	Space
F	070		”	034	Quotation mark
G	071		#	035	Hash sign
H	072		$	036	Dollar sign
I	073		%	037	Percent
J	074		&	038	Ampersand
K	075		’	039	Apostrophe
L	076		(	040	Left parenthesis
M	077		)	041	Right parenthesis
N	078		*	042	Asterisk
O	079		+	043	Plus sign
P	080		,	044	Comma
Q	081		–	045	Minus sign
R	082		.	046	Period
S	083		/	047	Slash
T	084		:	058	Colon
U	085		;	059	Semicolon
V	086		<	060	Left angle bracket
W	087		=	061	Equal sign
X	088		>	062	Right angle bracket
Y	089		?	063	Question mark
Z	090		@	064	”At”mark
0	048		[	091	Left square bracket
1	049		0	092	
2	050		¥	093	Yen sign
3	051		]	094	Right square bracket
4	052		_	095	Underscore
5	053				

Table 13.3 (b) Numbers of subprograms employed in the pattern data input function

Subprogram No.	Function
O9500	Specifies character strings displayed on the pattern data menu.
O9501	Specifies a character string of the pattern data corresponding to pattern No.1
O9502	Specifies a character string of the pattern data corresponding to pattern No.2
O9503	Specifies a character string of the pattern data corresponding to pattern No.3
O9504	Specifies a character string of the pattern data corresponding to pattern No.4
O9505	Specifies a character string of the pattern data corresponding to pattern No.5
O9506	Specifies a character string of the pattern data corresponding to pattern No.6
O9507	Specifies a character string of the pattern data corresponding to pattern No.7
O9508	Specifies a character string of the pattern data corresponding to pattern No.8
O9509	Specifies a character string of the pattern data corresponding to pattern No.9
O9510	Specifies a character string of the pattern data corresponding to pattern No.10

Table. 13.3 (c) Macro instructions used in the pattern data input function

G code	H code	Function
G65	H90	Specifies the menu title.
G65	H91	Specifies the pattern name.
G65	H92	Specifies the pattern data title.
G65	H93	Specifies the variable name.
G65	H94	Specifies the comment.

Table. 13.3 (d) System variables employed in the pattern data input function

System variable	Function
#5900	Pattern No. selected by user.

14 PROGRAMMABLE PARAMETER ENTRY (G10)

The values of parameters can be entered in a program. This function is used for the maximum moving feedrate or time constants are changed to meet changing operation conditions.

Format

Format

```
G10L50; Parameter entry mode setting
N_R_;   For parameters other than the axis type
N_P_R_;         For axis type parameters
 :
 :
G11;    Parameter entry mode cancel
```

Meaning of command

N_:	Parameter No. (4digids)
R_:	Parameter setting value (Leading zeros can be omitted.)
P_:	Axis No. 1 to 8 (Specifying for entering axis type parameters)

Explanations

- **Parameter setting value (R_)**

Do not use a decimal point in a value set in a parameter (R_).
a decimal point cannot be used in a custom macro variable for R_ either.

- **Axis No.(P_)**

Specify an axis number (P_) from 1 to 6 (up to six axes) for an axis type parameter. The control axes are numbered in the order in which they are displayed on the controller display.
For example, specify P2 for the control axis which is displayed second.

WARNING
Do not fail to perform reference point return manually after changing backlash compensation data. Without this, the machine position can deviate from the correct position.

NOTE
Other NC statements cannot be specified while in parameter input mode.

Examples

1. Set bit 2 (SPB) of bit type parameter No. 3404

```
G10L50 ;                Parameter entry mode
N3404 R 00000100 ;      SBP setting
G11 ;                   cancel parameter entry mode
```

2. Change the values for the Z–axis and A–axis in axis type parameter No. 1322 (the coordinates of stored stroke limit 2 in the positive direction for each axis).

```
G10L50 ;                Parameter entry mode
N1322P3R4500 ;          Modify Z axis
N1322P4R12000 ;         Modify A axis
G11 ;                   Cancel parameter entry mode
```

15 AXIS CONTROL FUNCTIONS

15.1 ROTARY AXIS ROLL–OVER

The roll–over function prevents coordinates for the rotation axis from overflowing. The roll–over function is enabled by setting bit 0 of parameter 1008 to 1.

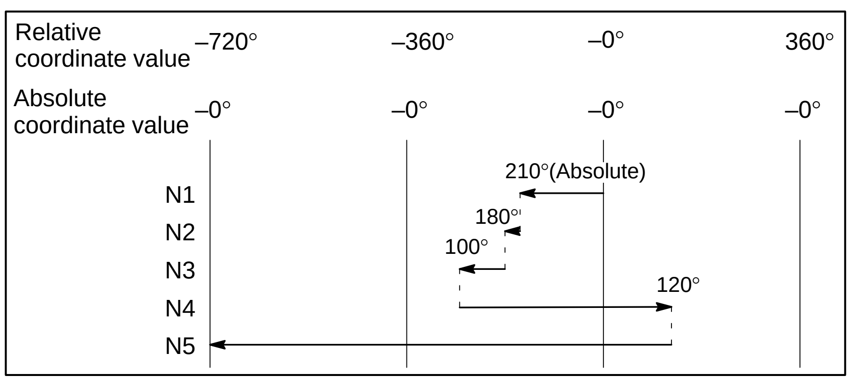
Explanations

For an incremental command, the tool moves the angle specified in the command. For an absolute command, the coordinates after the tool has moved are values set in parameter No. 1260, and rounded by the angle corresponding to one rotation. The tool moves in the direction in which the final coordinates are closest when bit 1 of parameter No. 1008 is set to 0. Displayed values for relative coordinates are also rounded by the angle corresponding to one rotation when bit 2 of parameter No. 1008 is set to 1.

Examples

Assume that axis A is the rotating axis and that the amount of movement per rotation is 360.000 (parameter No. 1260 = 360000). When the following program is executed using the roll–over function of the rotating axis, the axis moves as shown below.

G90 A0 ;	Sequence number	Actual movement value	Absolute coordinate value after movement end
N1 G90 A–150.0 ;	N1	–150	210
N2 G90 A540.0 ;	N2	–30	180
N3 G90 A–620.0 ;	N3	–80	100
N4 G91 A380.0 ;	N4	+380	120
N5 G91 A–840.0 ;	N5	–840	0

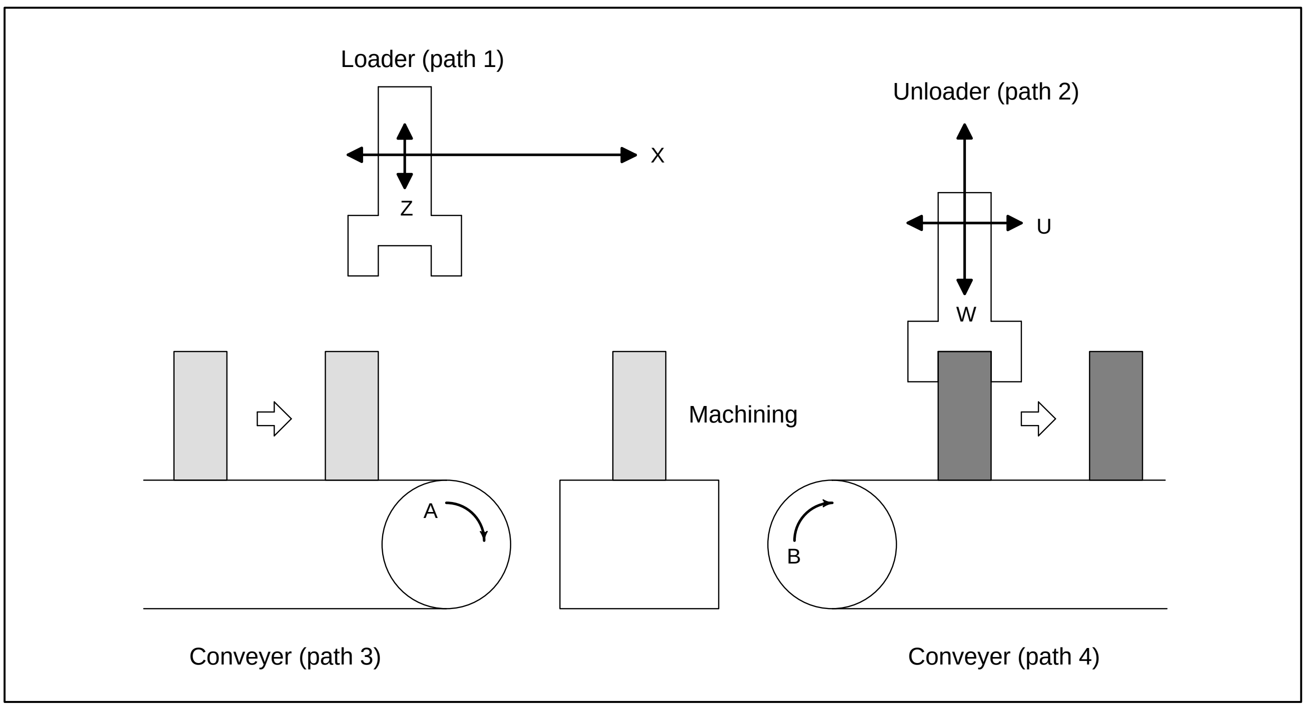


16 MULTIPATH CONTROL

If a machine has several sections that operate independently, they can operate in a group of one or more axes independently of one another under control of one program. A group of axes that can be controlled by this one program is called a path.
With the Power Mate–H, it is possible to specify arbitrarily by program what axes are to be assigned to what path. If a machine has six controllable axes, up to six paths can be specified by assigning one axis to each path. Use of six program commands can make six sections operate simultaneously, but independently.

Examples

In the figure below, the machine operates with four paths, which can be controlled by one Power Mate–H unit.
Path 1: Loader; X–axis and Z–axis
Path 2: Unloader; U–axis and W–axis
Path 3: Conveyer 1; A–axis
Path 4: Conveyer 2; B–axis

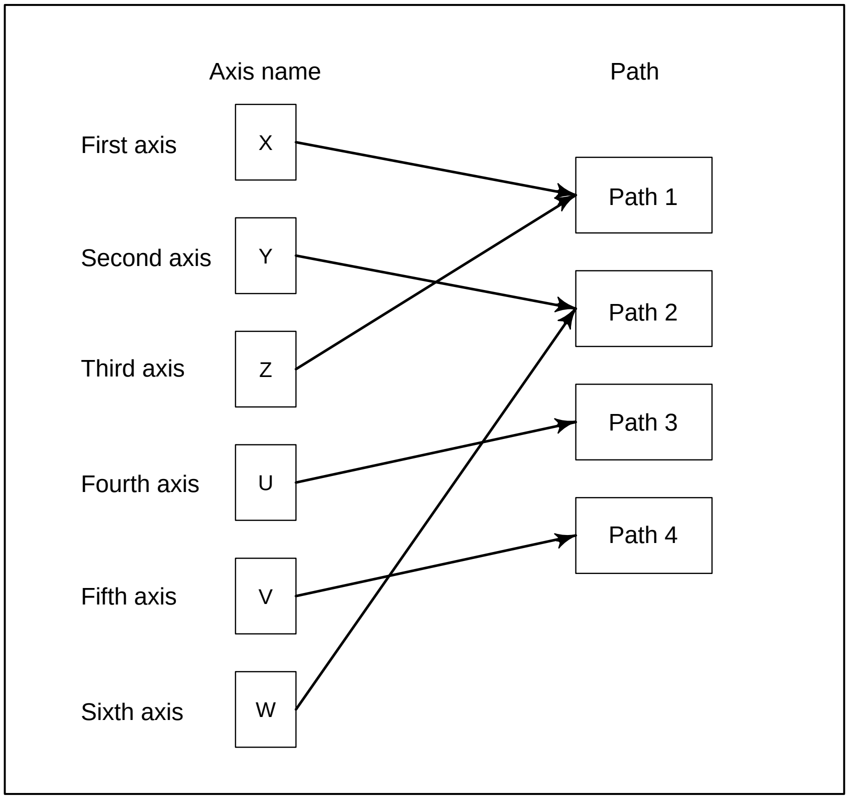


16.1 AXIS NAME AND PATH

Each axis is named X, Y, Z, etc. A path can be defined with program commands by specifying the names of axes that belong to the path. Any combination of axes and paths is possible. Axis–path combinations can be changed during operation.

Examples

Suppose a machine has six axes, X–axis, Y–axis, Z–axis, U–axis, V–axis, and W–axis. This example assigns the X– and Z–axes to path 1, the Y– and W–axes to path 2, the U–axis to path 3, and the V–axis to path 4.



Format

G130 Pn α1 β1 ···; (n=1, 2, 3, 4, 5, 6) : Specifies control over path n.
(α, β, ···) : Specifies axes that are to be assigned to a specific path.

The Power Mate–H is in the one–path mode, in which all axes belong to one path, immediately after power is applied or a reset occurs. The following G code specifies a multipath mode.

A multipath program contains several path–specific programs arranged by path. A G code to exert multipath control must be included at the beginning of each path–specific program. A path–specific program continues until the next G130 code appears.

Number n (up to 6) after the letter P indicates a path number to which the program that follows belongs. For four–path control, numbers 1 to 4 are given in this sequence.

An axis that belongs to a certain path is indicated with its name and number 1 that follows it. One axis cannot belong to more than one path. Meanwhile, each controllable axis must be assigned to some path.

The Power Mate–H is in the one–path mode immediately after power is applied or a reset occurs. To use it in the multipath mode, therefore, it is necessary to specify this G code.

G130 P0; : Multipath control off

This G code terminates the multipath mode and places the Power Mate–H back in the one–path mode. To specify a different combination of paths, it is necessary to return to the one–path mode.

Examples

This example assigns the X– and Y–axes to path 1, the A– and B–axes to path 2, and the U– and V–axes to path 3.

```
O0001 ;       One–path mode
G90 G00 X0 Y0 A0 ;
B0 U0 V0 ;
  .
  .                           Turn on the
G130 P1 XI Y1 ;               multipath        Define X– and Y–axes as path 1.
G00 X100. Y100. ;             control mode
G01 X150. Y150. F100;                          Path 1 program
  .
G130 P2 A1 B1 ;                                Define A– and B–axes as path 2.
G00 A–100. B–100. ;
G01 A–200. B–200. F150 ;                       Path 2 program
  .
G130 P3 U1 V1 ;                                Define U– and V–axes as path 3.
G00 U123. V456. ;
G01 U234. V567. F789 ;                         Path 3 program
  .
G130 P0 ;                       Turn off the multipath control mode.
  .                             One–path mode
  .
M02 ;
```

Explanations

- **Operation during the multipath mode**

The multipath mode can be used only during memory operation. It cannot be specified during other types of operation.

If the G130 code (G code to select the multipath mode) is specified during the one–path mode, it causes the machine to shift to the multipath mode. In the multipath mode, a path–specific program runs independently of, but simultaneously with another path–specific program. If a path–specific program ends execution of a block, it moves to the next block regardless of whether another path–specific program completes a block that it is running.

An inter–path wait function is available, which enables starting the operation of a path in synchronization with that of another path. When this function is specified, a specified block is interrupted and kept in a wait state. When a wait command with the same ID No. is encountered in all specified paths, the wait condition is met and the interrupted execution is resumed.

During the multipath mode, when a certain path–specific program is finished, the path is placed and kept in a wait state until other path–specific programs are finished. When all specified paths complete their programs, the Power Mate–H exits the multipath mode and returns to the one–path mode.

A series of multipath mode operations can be specified between G130P1 and G130P0. This specification can be included any number of times in one program. A combination of axes for each path can be changed at each specification.

NOTE
A in–position check is performed each time the multipath mode function is turned on and off.

NOTE

Functions that can be used during the multipath mode
The following functions can be used during the multipath mode. Specifying any other function results in an alarm being issued.

G00	Rapid traverse
G01	Feed with a speed specified
G04	Dwell
G28	Reference position return
G90	Absolute command
G91	Incremental command
G92	Coordinate system setting
G94	Feed per minute
G95	Feed per rotation
Two–digit M command	

N	Sequence number
F	Feedrate
P	Dwell time

Limitations

The following functions cannot be used during the multipath mode.

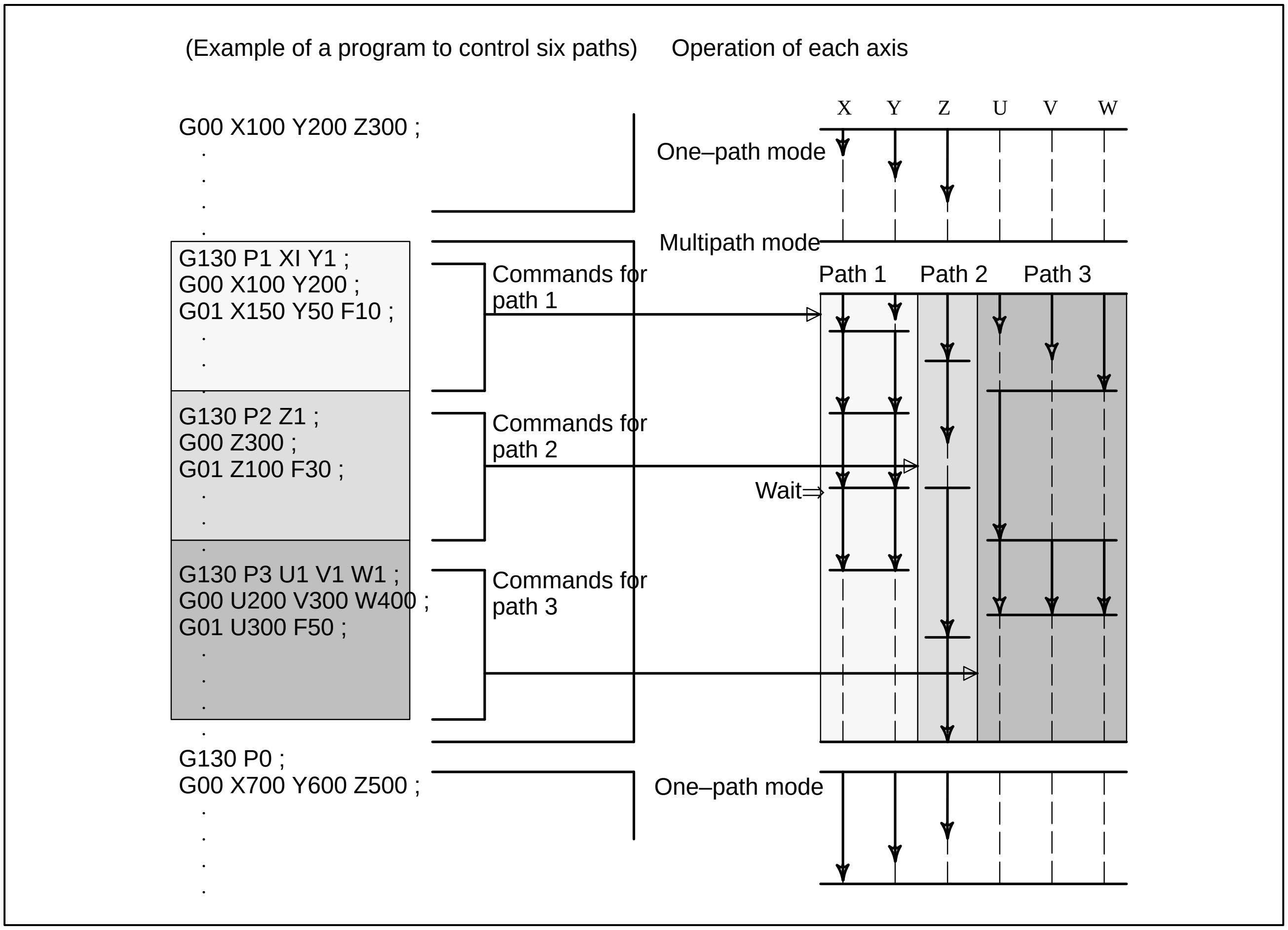
G10 Data setting
G11 Cancel data setting mode
G20 Inch input
G21 Metric input
G27 Reference position return check
G29 Return from reference position
G30 Second and third reference position return
G31 Skip command
G43 Tool length compensation +
G44 Tool length compensation –
G49 Cancel machine length compensation
G65 Macro instruction
G66 Custom macro modal call
G67 Cancel custom macro modal call
G93 Feed at specified rate
M98 Subprogram call
Macro instruction (except reference to variables)
Feedrate switching

NOTE

1. It is impossible to search for and activate part of a multipath program.
2. The M function cannot be used together with other commands. It must be specified in a separate block.

Examples

The following example programs control of six axes in three paths.

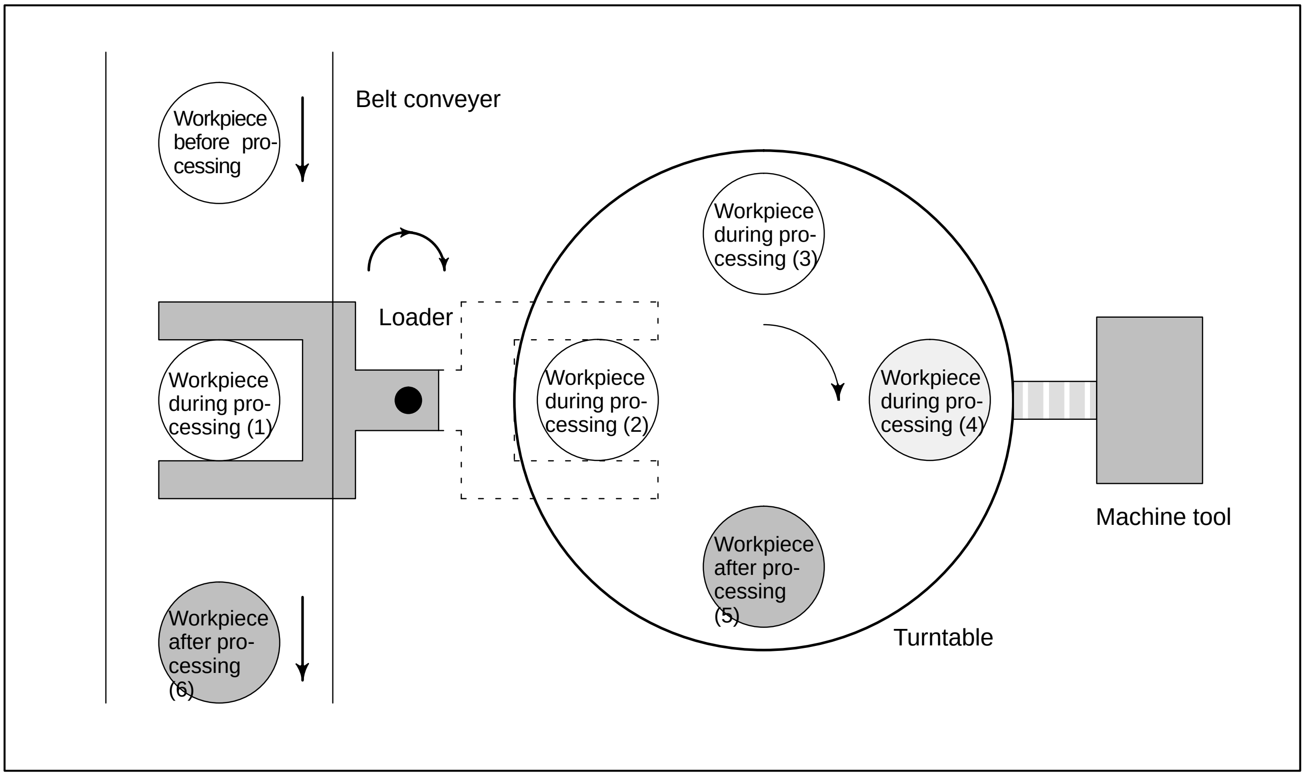


16.2 WAIT FUNCTION

During automatic operation in the multipath mode, programs for different paths run simultaneously, but independently. These programs can be made to wait for others, and therefore it is possible to operate the machine in synchronous manner. A path specified to wait begins to operate when a wait condition in the associated path is satisfied.

Examples

The following example loads a workpiece from a belt conveyer to a machine tool and unloads it from the machine tool. The loader waits until the belt conveyer and table operate to place a workpiece at a specified position.



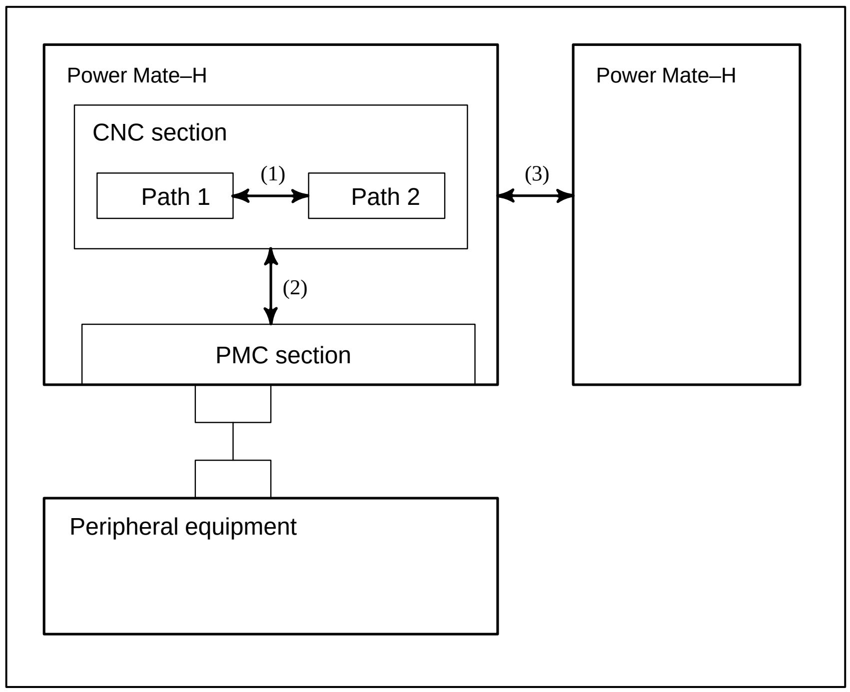
16.2.1 Types of Wait Functions

There are three types of wait functions.

(1) Inter–path wait
This type of wait is used by one path to wait for another within the same Power Mate–H.

(2) Wait for peripheral equipment
This type of wait is used to wait for peripheral equipment according to PMC signal conditions.

(3) Wait for another Power Mate–H
This type of wait is used by one Power Mate–H to wait for another.



Format

The inter–path wait function can be specified using a ”9xx” M code in a part program for each path.

αx M9mm Pp Qq Rr;

αx : A move command can also be specified. If a move command is specified, a value of Q can specify whether to start movement after a specified wait condition is satisfied or immediately when a wait condition is satisfied provided that the wait condition only pertains to the distance yet to be traveled.

M9mm : Two or more wait conditions (events) can be awaited simultaneously. To prevent an incorrect combination of wait–event relationship, a ”9xx” M code is used to specify an ID number. The same M code must be specified for paths that wait for the same condition. The M function is processed internally. It is not output to the PMC.

Pp : Specifies what paths to wait. Usually, two to six paths can be specified. The data of P is given using the code listed below.

(General form**)**

$$P = \sum_{n=1}^{6} Kn * 2^{n-1}$$

(Kn=0:n No wait in path n
1:n Wait in path n)

	Path					
P	6	5	4	3	2	1
01	X	X	X	X	X	O
02	X	X	X	X	O	X
03	X	X	X	X	O	O
04	X	X	X	O	X	X
05	X	X	X	O	X	O
06	X	X	X	O	O	X
07	X	X	X	O	O	O
08	X	X	O	X	X	X
09	X	X	O	X	X	O
10	X	X	O	X	O	X
11	X	X	O	X	O	O
12	X	X	O	O	X	X
13	X	X	O	O	X	O
14	X	X	O	O	O	X
15	X	X	O	O	O	O
16	X	O	X	X	X	X
17	X	O	X	X	X	O
18	X	O	X	X	O	X
19	X	O	X	X	O	O
20	X	O	X	O	X	X

	Path					
P	6	5	4	3	2	1
21	X	O	X	O	X	O
22	X	O	X	O	O	X
23	X	O	X	O	O	O
24	X	O	O	X	X	X
25	X	O	O	X	X	O
26	X	O	O	X	O	X
27	X	O	O	X	O	O
28	X	O	O	O	X	X
29	X	O	O	O	X	O
30	X	O	O	O	O	X
31	X	O	O	O	O	O
32	O	X	X	X	X	X
33	O	X	X	X	X	O
34	O	X	X	X	O	X
35	O	X	X	X	O	O
36	O	X	X	O	X	X
37	O	X	X	O	X	O
38	O	X	X	O	O	X
39	O	X	X	O	O	O
40	O	X	O	X	X	X

	Path					
P	6	5	4	3	2	1
41	O	X	O	X	X	O
42	O	X	O	X	O	X
43	O	X	O	X	O	O
44	O	X	O	O	X	X
45	O	X	O	O	X	O
46	O	X	O	O	O	X
47	O	X	O	O	O	O
48	O	O	X	X	X	X
49	O	O	X	X	X	O
50	O	O	X	X	O	X
51	O	O	X	X	O	O
52	O	O	X	O	X	X
53	O	O	X	O	X	O
54	O	O	X	O	O	X
55	O	O	X	O	O	O
56	O	O	O	X	X	X
57	O	O	O	X	X	O
58	O	O	O	X	O	X
59	O	O	O	X	O	O
60	O	O	O	O	X	X
61	O	O	O	O	X	O
62	O	O	O	O	O	X
63	O	O	O	O	O	O

O : Waiting path
X : Nonwaiting path

Qq: Specifies a wait condition.

Q value	Wait condition
0 or no specification	If a move command is specified, movement begins after the wait condition is satisfied. If no move command is specified, the next block is processed after the wait condition is satisfied.
10n	Movement begins at the same time the block begins. If the distance yet to be traveled by axis n (in the sequence displayed) in the corresponding path becomes less than the distance specified with R, the wait condition is regarded to have been satisfied, and it is reported to other paths.
200	Movement begins at the same time the block begins. If the time specified with R elapses since the beginning of the block, the wait condition is regarded to have been satisfied, and it is reported to other paths.

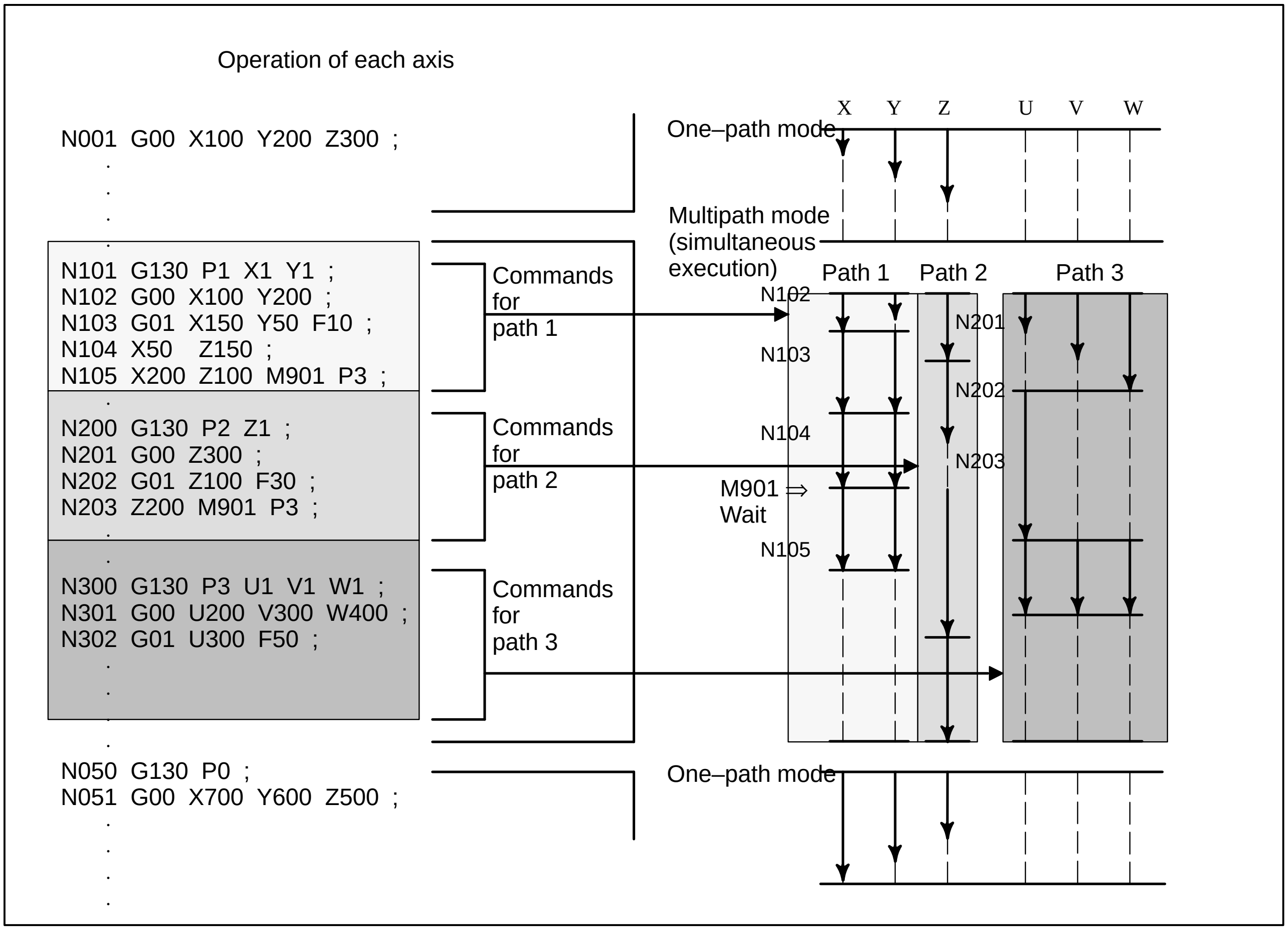
Rr: Specifies a remaining distance or time to be waited for, when a wait condition is specified.

If q = 10n: Specifies the distance yet to be traveled by axis n in the corresponding path. The wait condition is regarded to have been satisfied when this distance is reached.

If q = 200: If the time specified with R elapses since the beginning of the block, the wait condition is regarded to have been satisfied.

Examples

[Example of a wait program]
This example waits for block N105 in path 1 and block N203 in path 2 with M901 used to identify paths 1 and 2.
In the example, block N203 in path 2 is read earlier than block N105 in path 1, but movement does not begin. Instead, it waits until N105 is read. When N105 is read, both paths start operating.



16.3 WAIT BY THE PMC SIGNAL CONDITION

This type of wait function is realized using M801 to M815, which are prepared to facilitate waiting for other machines or peripheral equipment.

When a wait M code is specified, the corresponding signal (WAT1 to WAT4) is output. The operation of a block is delayed until the completion signal (WFN1 to WFN4) corresponding to that block is input.

Format

Command format : N—G—X—F—M8nn Pp Qq Rr ;

Explanations

Use of P, Q, and R also enables an inter–path wait.

M code	Output signal				Completion input signal condition			
	WAT4	WAT3	WAT2	WAT1	WFN4	WFN3	WFN2	WFN1
801	0	0	0	1	0	0	0	1
802	0	0	1	0	0	0	1	0
803	0	0	1	1	0	0	1	1
804	0	1	0	0	0	1	0	0
805	0	1	0	1	0	1	0	1
806	0	1	1	0	0	1	1	0
807	0	1	1	1	0	1	1	1
808	1	0	0	0	1	0	0	0
809	1	0	0	1	1	0	0	1
810	1	0	1	0	1	0	1	0
811	1	0	1	1	1	0	1	1
812	1	1	0	0	1	1	0	0
813	1	1	0	1	1	1	0	1
814	1	1	1	0	1	1	1	0
815	1	1	1	1	1	1	1	1

16.4 SIMULTANEOUS ACTIVATION OF MULTIPLE BLOCKS

Using M821 to M827 enables activating any blocks in up to four Power Mate–H units simultaneously. If a synchronization M code is specified in one unit, the operation is delayed until the corresponding synchronization M code is issued in a specified unit. The M codes are processed automatically in the Power Mate–H. They need not be processed by the PMC. If a move command is specified in the same block in more than one Power Mate–H unit, they start operating simultaneously when the wait condition is satisfied. This simultaneous block activation function cannot be used more than once at the same time. If an attempt is made to do so, an alarm is issued.

Format

Command format**:**N––G––X––F––M82n Pp Qq Rr ;

Explanations

Specifying P, Q, and R also enables an inter–path wait in the Power Mate–H that specifies the simultaneous block activation function.

<table>
<tr><th rowspan="2">Associated unit / Unit of interest</th><th colspan="4">One unit</th><th colspan="3" rowspan="2">Two units</th><th rowspan="2">Three units</th></tr>
<tr><th>#0</th><th>#1</th><th>#2</th><th>#3</th></tr>
<tr><td rowspan="2">#0</td><td rowspan="2">—</td><td rowspan="2">M824</td><td rowspan="2">M822</td><td rowspan="2">M821</td><td>#1,#2</td><td>#1,#3</td><td>#2,#3</td><td>#1,#2,#3</td></tr>
<tr><td>M826</td><td>M825</td><td>M823</td><td>M827</td></tr>
<tr><td rowspan="2">#1</td><td rowspan="2">M821</td><td rowspan="2">—</td><td rowspan="2">M824</td><td rowspan="2">M822</td><td>#0,#2</td><td>#0,#3</td><td>#2,#3</td><td>#0,#2,#3</td></tr>
<tr><td>M825</td><td>M823</td><td>M826</td><td>M827</td></tr>
<tr><td rowspan="2">#2</td><td rowspan="2">M822</td><td rowspan="2">M821</td><td rowspan="2">—</td><td rowspan="2">M824</td><td>#0,#1</td><td>#0,#3</td><td>#1,#3</td><td>#0,#1,#3</td></tr>
<tr><td>M826</td><td>M825</td><td>M823</td><td>M827</td></tr>
<tr><td rowspan="2">#3</td><td rowspan="2">M824</td><td rowspan="2">M822</td><td rowspan="2">M821</td><td rowspan="2">—</td><td>#0,#1</td><td>#0,#2</td><td>#1,#2</td><td>#0,#1,#2</td></tr>
<tr><td>M826</td><td>M825</td><td>M823</td><td>M827</td></tr>
</table>

Examples

(Example 1) To activate a block in #0 and #1 simultaneously, specify:
M824 in #0
M821 in #1

(Example 2) To activate a block in #0, #1, and #2 simultaneously, specify:
M826 in #0
M825 in #1
M823 in #2

III OPERATION

1 GENERAL

1.1 MANUAL OPERATION

Explanations

- **Manual reference position return**

The industrial machine usually has a position used to determine the machine position.
This position is called the reference position, where the attachment is replaced or the coordinate are set. Ordinarily, after the power is turned on, the tool is moved to the reference position.
Manual reference position return is to move the tool to the reference position using switches and pushbuttons located on the operator's panel. (See Section III–3.1)

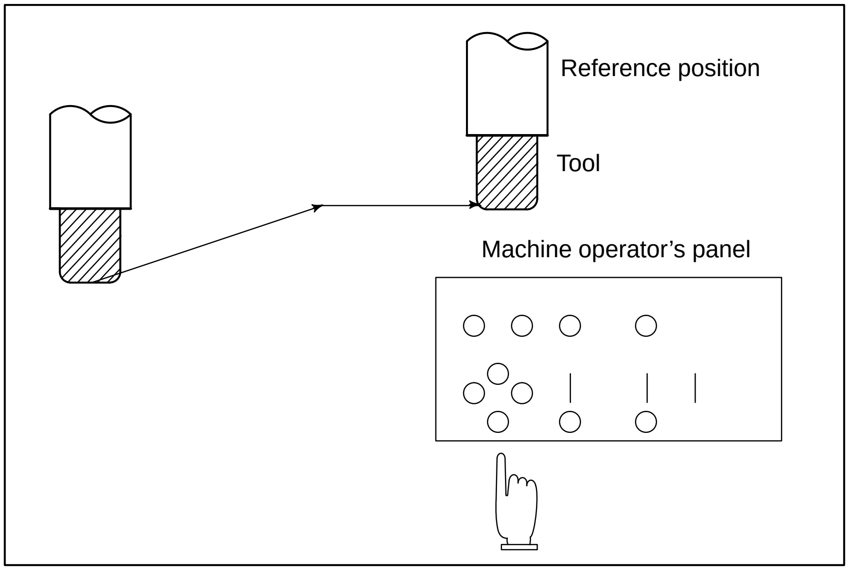


Fig.1.1 (a) Manual reference position return

The tool can be moved to the reference position also with program commands.
This operation is called automatic reference position return (See Section II–6).

- **The tool movement by manual operation**

Using machine operator's panel switches or pushbuttons, the tool can be moved along each axis.

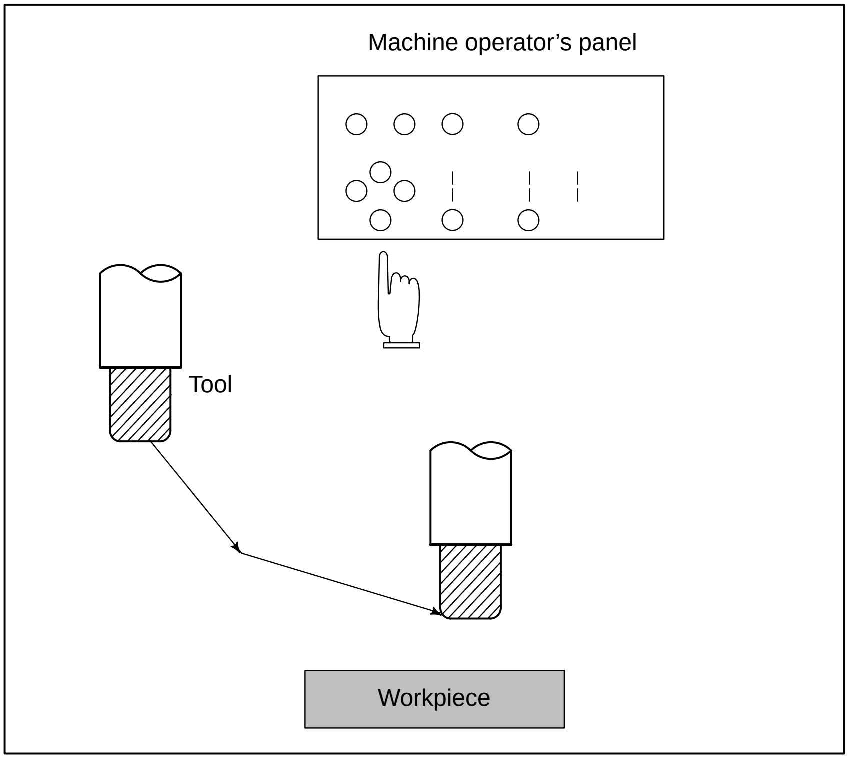


Fig.1.1 (b) The tool movement by manual operation

The tool can be moved in the following ways:

(i) Jog feed (See Section III–3.2)
The tool moves continuously while a pushbutton remains pressed.

(ii) Incremental feed (See Section III–3.3)
The tool moves by the predetermined distance each time a button is pressed.

1.2 TOOL MOVEMENT BY PROGRAMMING – AUTOMATIC OPERATION

Automatic operation is to operate the machine according to the created program. It includes memory and MDI operations. (See Section III–4).

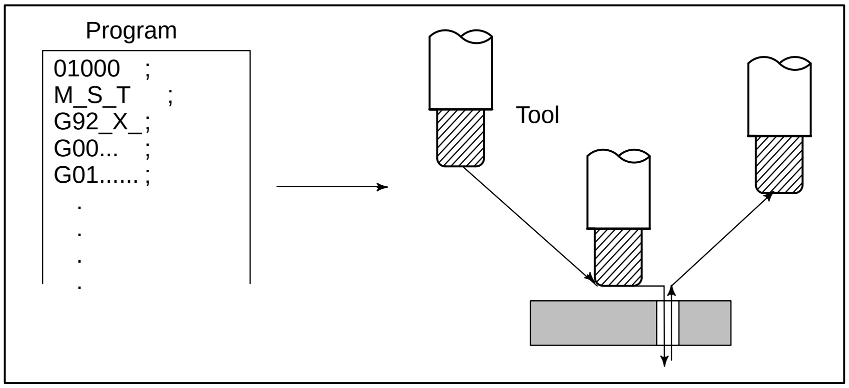


Fig.1.2 (a) Tool Movement by Programming

Explanations

- **Auto operation**

After the program is once registered in memory of controller, the machine can be run according to the program instructions. This operation is called Auto operation.

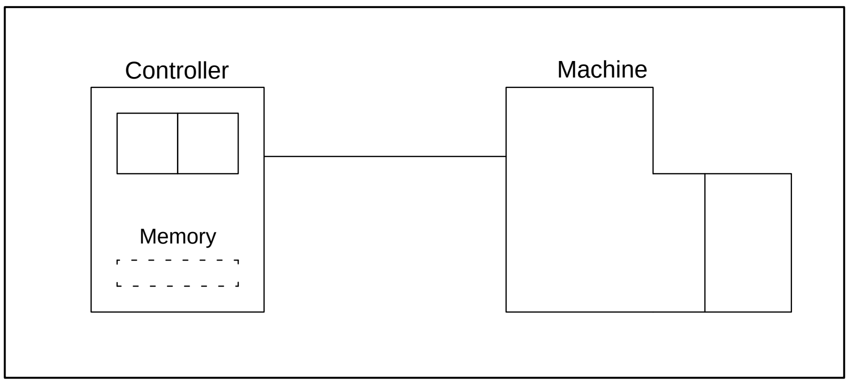


Fig.1.2 (b) Auto Operation

- **MDI operation**

After the program is entered, as an command group, from the MDI keyboard, the machine can be run according to the program. This operation is called MDI operation.

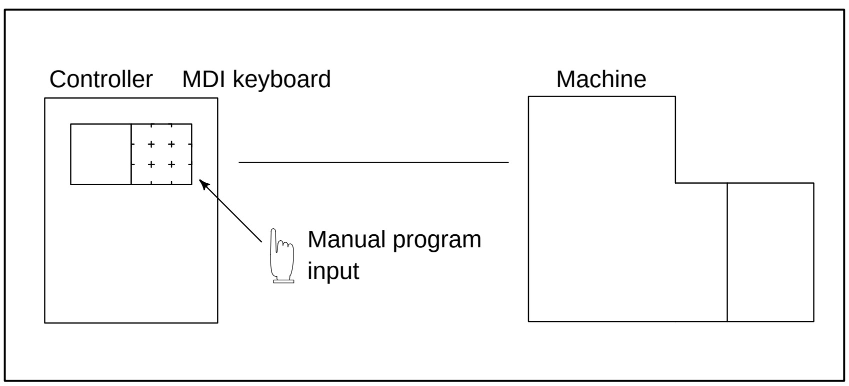


Fig.1.2 (c) MDI operation

1.3 AUTOMATIC OPERATION

Explanations

• Program selection

Select the program used for the workpiece. Ordinarily, one program is prepared for one workpiece. If two or more programs are in memory, select the program to be used, by searching the program number (Section III–9.3).

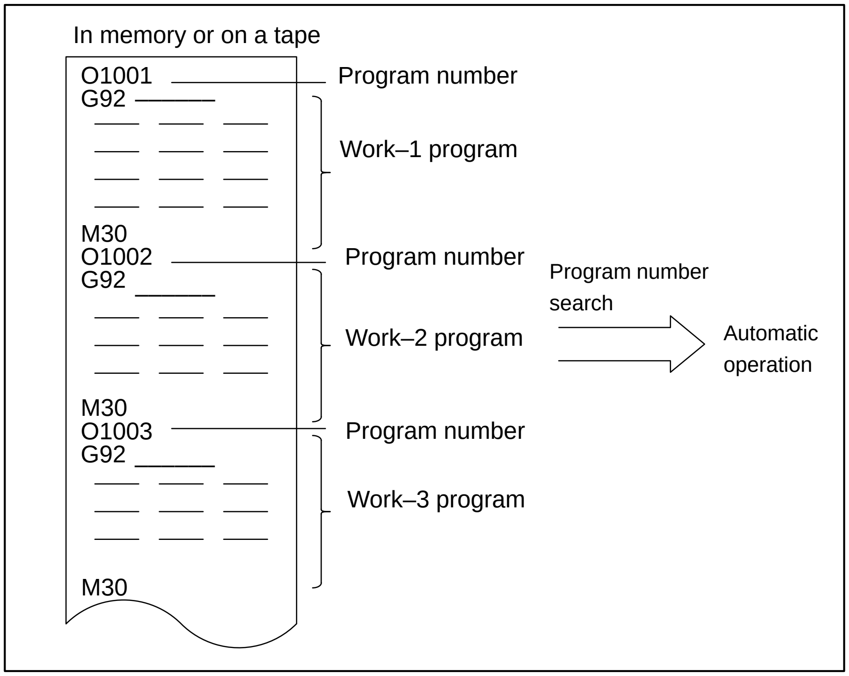


Fig.1.3 (a) Program Selection for Automatic Operation

• Start and stop

Pressing the cycle start pushbutton causes automatic operation to start. By pressing the feed hold or reset pushbutton, automatic operation pauses or stops. By specifying the program stop or program termination command in the program, the running will stop during automatic operation. When one process machining is completed, automatic operation stops. (See Section III–4)

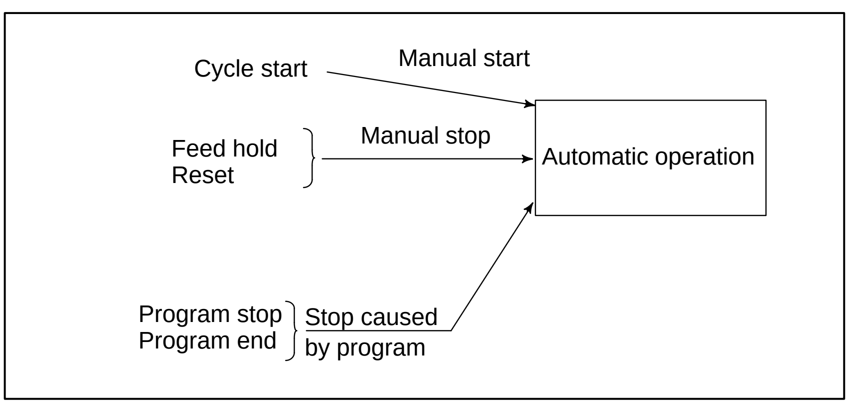


Fig.1.3 (b) Start and Stop for Automatic Operation

1.4 TESTING A PROGRAM

Before operation is started, the automatic running check can be executed. It checks whether the created program can operate the machine as desired. This check can be accomplished by running the machine actually or viewing the position display change (without running the machine) (See Section III–5).

1.4.1 Check by Running the Machine

Explanations

- **Dry run (See Section III–5.4)**

Remove the workpiece, check only movement of the tool. Select the tool movement rate using the dial on the operator's panel.

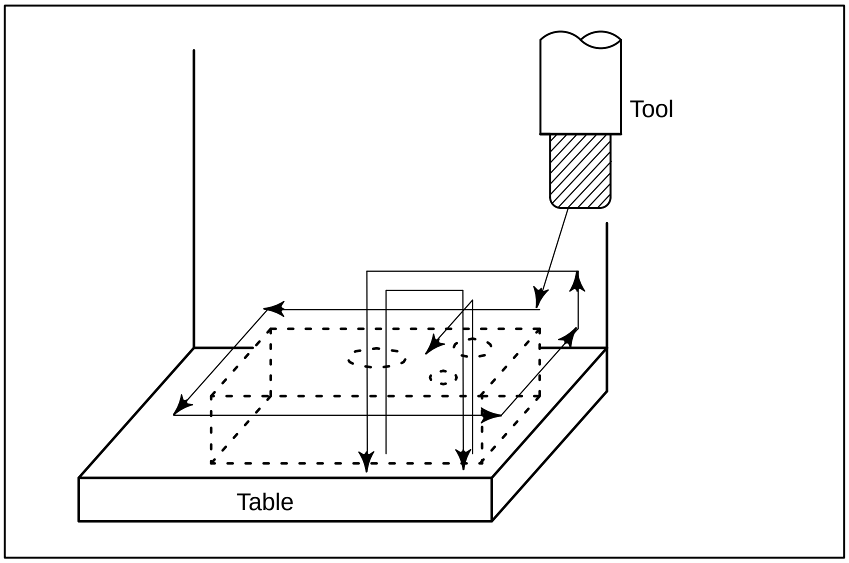


Fig.1.4 (a) Dry run

- **Feedrate override (See Section III–5.2)**

Check the program by changing the rate specified in the program.

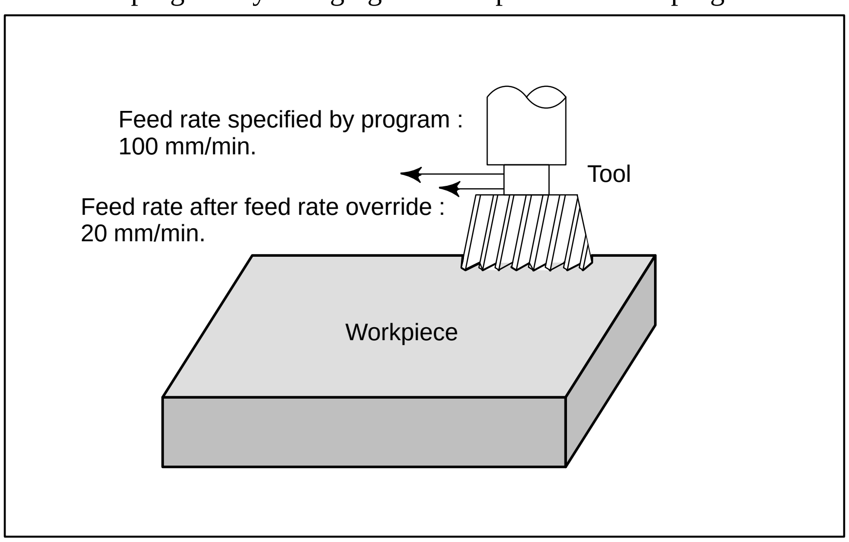


Fig1.4 (b) Feedrate Override

- **Single block (See Section III–5.5)**

When the cycle start pushbutton is pressed, the tool executes one operation then stops. By pressing the cycle start again, the tool executes the next operation then stops. The program is checked in this manner.

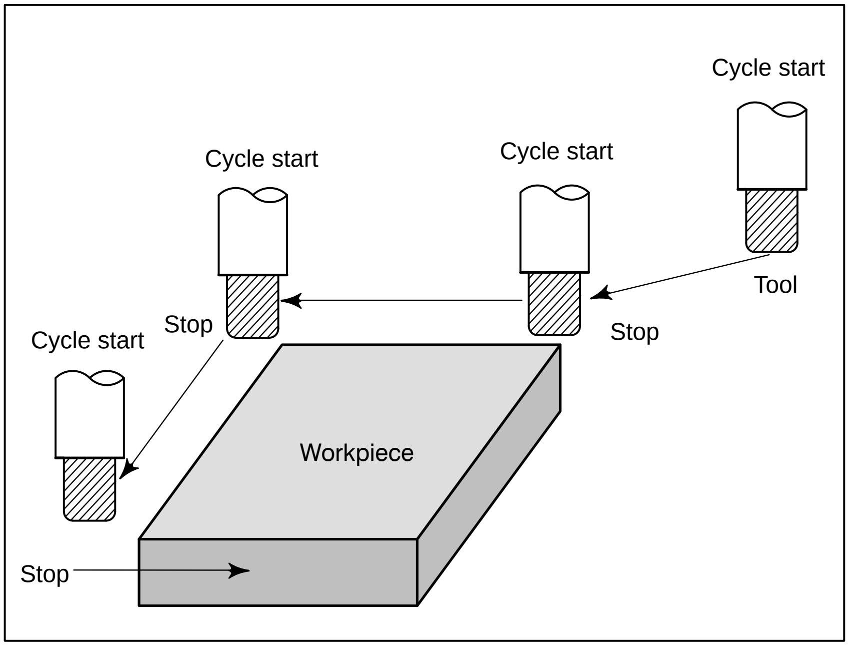


Fig.1.4 (c) Single Block

1.4.2 How to View the Position Display Change without Running the Machine

Explanations

- **Machine lock (See Sections III–5.1)**

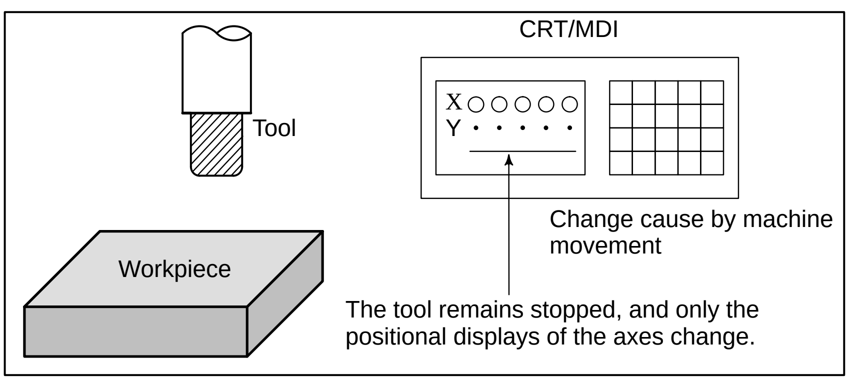


Fig1.4 (d) Machine Lock

- **Auxiliary function lock (See Section III–5.1)**

When automatic running is placed into the auxiliary function lock mode during the machine lock mode, all auxiliary functions are disabled.

1.5 EDITING A PART PROGRAM

After a created program is once registered in memory, it can be corrected or modified from the CRT/MDI panel (See Section III–9).
This operation can be executed using the part program storage/edit function.

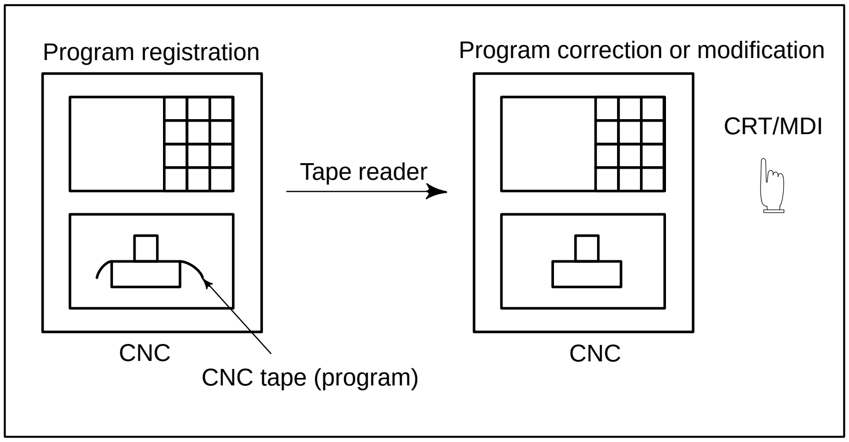


Fig.1.5 (a) Part Program Editing

1.6 DISPLAYING AND SETTING DATA

The operator can display or change a value stored in controller internal memory by key operation on the CRT/MDI screen (See III–11).

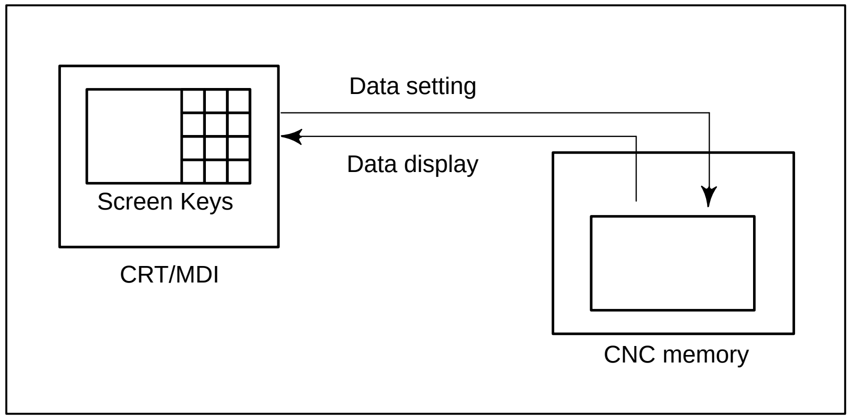


Fig.1.6 (a) Displaying and Setting Data

Explanations

- **Offset value**

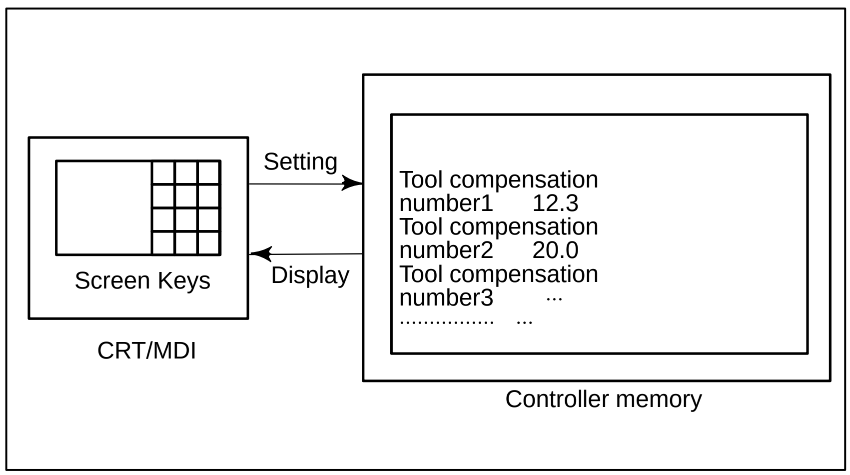


Fig.1.6 (b) Displaying and Setting Offset Values

Machining requires the use of a tool.
The tool has the tool dimension (length). When a workpiece is machined, the tool movement route depends on the tool dimensions.
By setting tool dimension data in controller memory beforehand, automatically generates tool routes that permit any tool to cut the workpiece specified by the program. Tool dimension data is called the offset value (See Section III–11.4.1).

● Displaying and setting operator's setting data

Apart from parameters, there is data that is set by the operator in operation. This data causes machine characteristics to change.
For example, the following data can be set:
·Inch/Metric switching
·Data related to I/O devices
·Mirror image operation on/off
The above data is called setting data (See Section III–11.4.2).

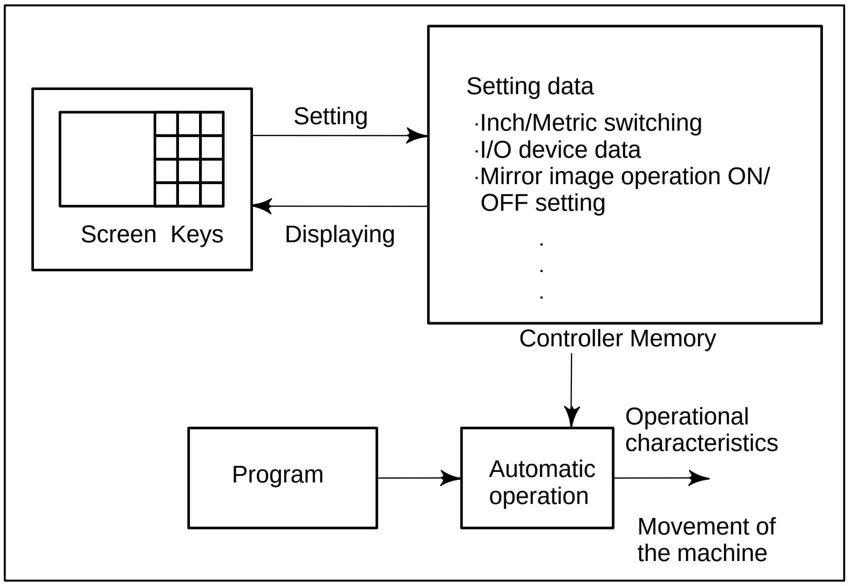


Fig.1.6 (c) Displaying and Setting Operator's setting data

● Displaying and setting parameters

The controller functions have versatility in order to take action in characteristics of various machines.
For example, CNC can specify the following:
·Rapid traverse rate of each axis
·Whether increment system is based on metric system or inch system.
·How to set command multiply/detect multiply (CMR/DMR)
Data to make the above specification is called parameters (See Section III–11.5.1).
Parameters differ depending on machine tool.

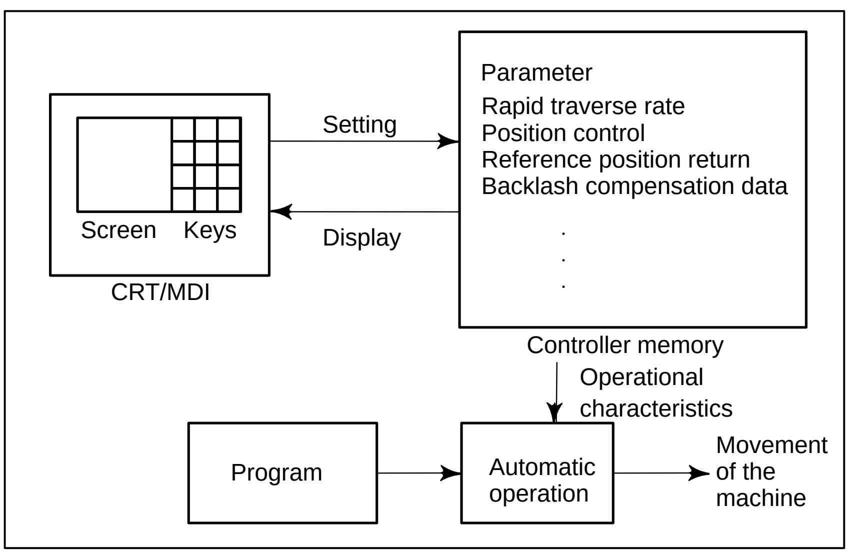


Fig.1.6 (d) Displaying and setting parameters

● Data protection key

A key called the data protection key can be defined. It is used to prevent part programs, offset values, parameters, and setting data from being registered, modified, or deleted erroneously (See Section III–11).

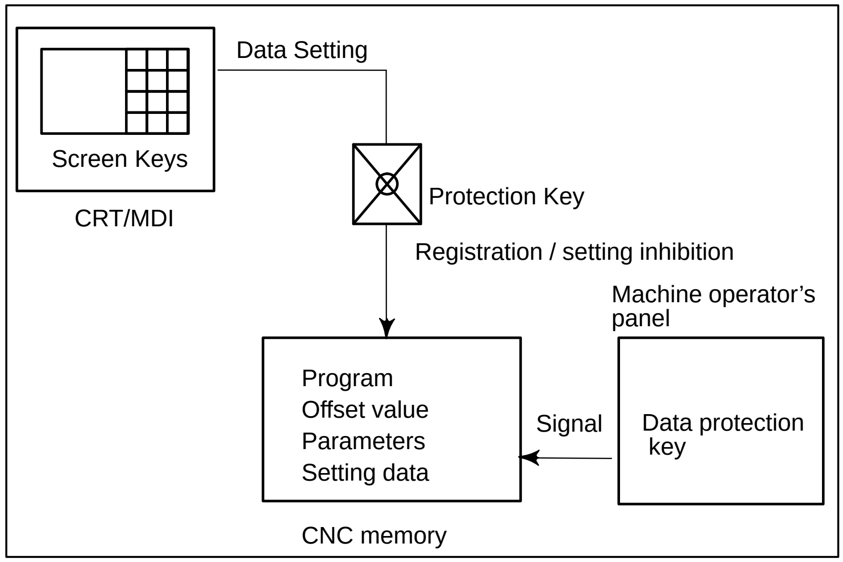
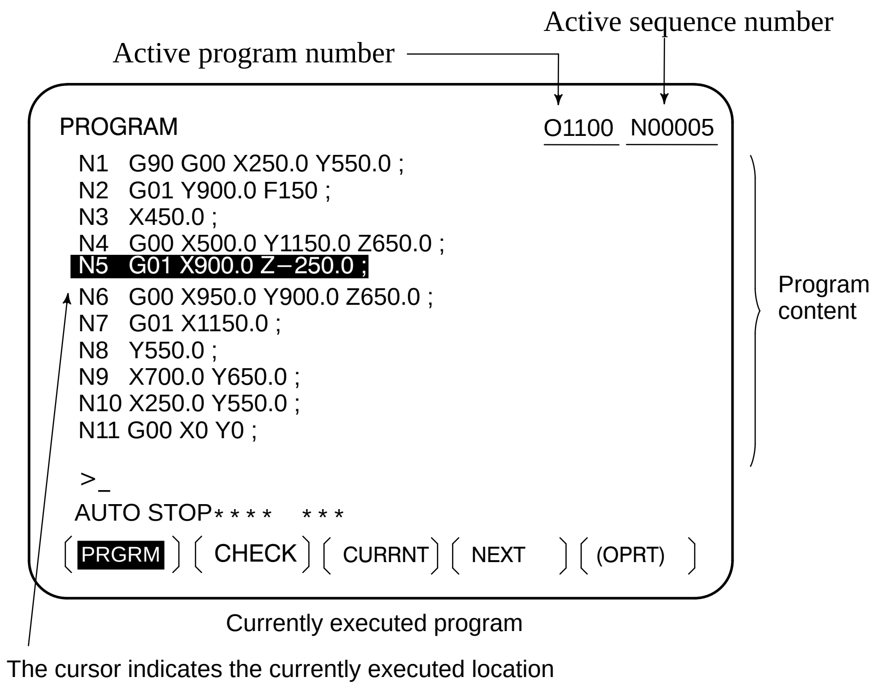


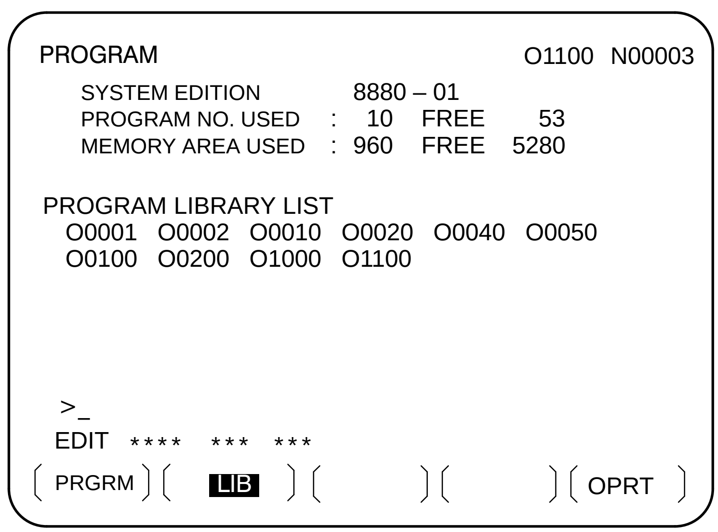
Fig.1.6 (e) Data Protection Key

1.7 DISPLAY

1.7.1 Program Display

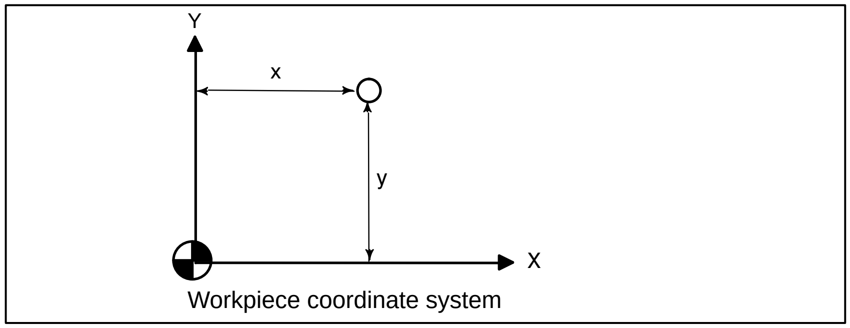
The contents of the currently active program are displayed. In addition, the programs scheduled next and the program list are displayed. (See Section III–11.2.1)





1.7.2 Current Position Display

The current position of the tool is displayed with the coordinate values. The distance from the current position to the target position can also be displayed. (See Section III–11.1 to 11.1.3)




```
 ACTUAL POSITION(WORK)          O1000 N00010

     X        123.456
     Y        363.233
     Z          0.000

               PART COUNT        5
  RUN TIME    0H15M  CYCLE TIME   0H 0M38S
  ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][      ][ (OPRT) ]
```

1.7.3 Alarm Display

When a trouble occurs during operation, error code and alarm message are displayed on CRT screen. See APPENDIX G for the list of error codes and their meanings. (See Section III–7.1)

```
ALARM MESSAGE                       O1000  N00003

 010        IMPROPER G–CODE



 >_
 AUTO STOP****   ***  ALM
( ALARM  )( MSG  )( HISTRY )(       )(       )
```

1.7.4 Parts Count Display, Run Time Display

When this option is selected, two types of run time and number of parts are displayed on the screen. (See Section lll–11.4.3)

```
ACTUAL POSITION(WORK)          O1000 N00010

   X        123.456
   Y        363.233
   Z          0.000

            PART COUNT       5
 RUN TIME    0H15M  CYCLE TIME   0H 0M38S
 ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][       ][ (OPRT) ]
```

1.8 DATA OUTPUT

Programs, offset values, parameters, etc. input in controller memory can be output to paper tape, cassette, or a floppy disk for saving. After once output to a medium, the data can be input into controller memory.

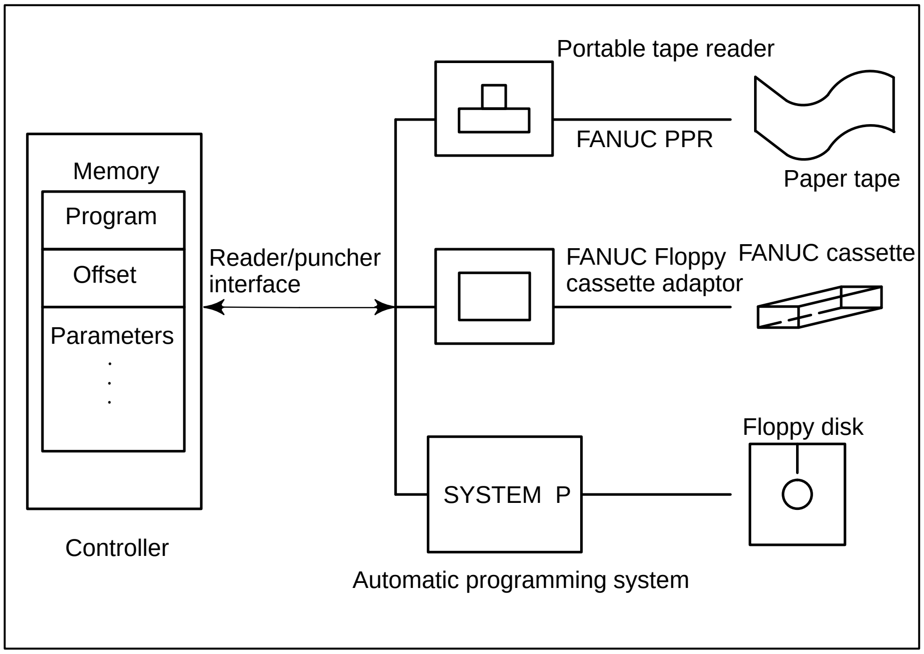


Fig.1.8 (a) Data Output

2 OPERATIONAL DEVICES

The peripheral devices available include the CRT/MDI panel (or DPL/MDI panel) attached to the controller, machine operator's panel and external input/output devices such as floppy cassette and Handy File.

2.1 DISPLAY PANEL

2.1.1 CRT/MDI Panel

Fig. 2.1.1 show the CRT/MDI panel.

9” small monochrome CRT/MDI panel (horizontal type) Fig.2.1.1

External view

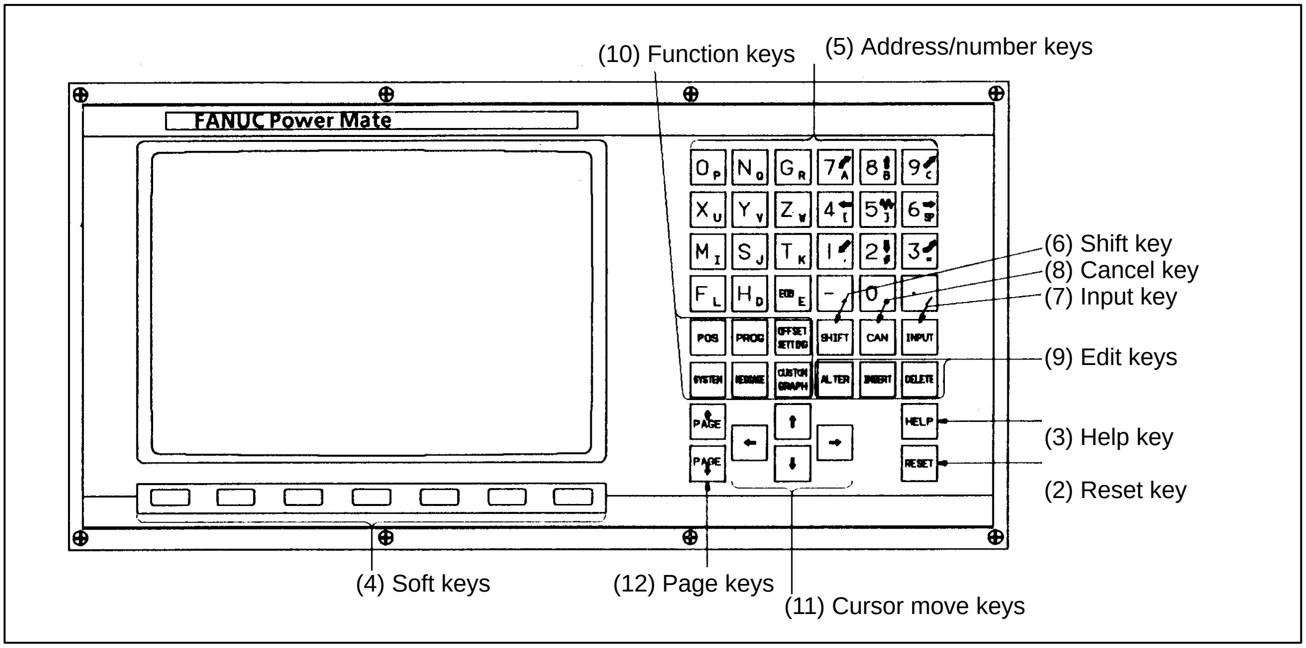


Fig. 2.1.1 CRT/MDI panel

Table2.1.1 Explanation of the MDI keyboard

Number	Name	Explanation
1	Power ON and OFF buttons	Press theses buttons to turn CNC power ON and OFF.
2	RESET key RESET	Press this key to reset the CNC, to cancel an alarm, etc.
3	HELP key HELP	Press this button to use the help function when uncertain about the operation of an MDI key (help function).
4	Soft keys	The soft keys have various functions, according to the Applications. The soft key functions are displayed at the bottom of the CRT screen.

Table2.1.1 Explanation of the MDI keyboard

Number	Name	Explanation
5	Address and numeric keys N Q 4 ← [	Press these keys to input alphabetic, numeric, and other characters.
6	SHIFT key SHIFT	Some keys have two characters on their keytop. Pressing the SHIFT key switches the characters. Special character Ê is displayed on the screen when a character indicated at the bottom right corner on the keytop can be entered.
7	INPUT key INPUT	When an address or a numerical key is pressed, the data is input to the buffer, and it is displayed on the CRT screen. To copy the data in the key input buffer to the offset register, etc., press the INPUT key. This key is equivalent to the [INPUT] key of the soft keys, and either can be pressed to produce the same result.
8	Cancel key CAN	Press this key to delete the last character or symbol input to the key input buffer. When the key input buffer displays >N001X100Z_ and the cancel CAN key is pressed, Z is canceled and >N001X100_ is displayed.
9	Program edit keys ALTER INSERT DELETE	Press these keys when editing the program. ALTER : Alteration INSERT : Insertion DELETE : Deletion
10	Function keys POS PROG …	Press theses keys to switch display screens for each function. See sec. 2.2 for detailas of the function keys.
11	Cursor move keys ← ↑ ↓ →	There are four different cursor move keys. → : This key is used to move the cursor to the right or in the forward direction. The cursor is moved in short units in the forward direction. ← : This key is used to move the cursor to the left or in the reverse direction. The cursor is moved in short units in the reverse direction. ↓ : This key is used to move the cursor in a downward or forward direction. The cursor is moved in large units in the forward direction. ↑ : This key is used to move the cursor in an upward or reverse direction. The cursor is moved in large units in the reverse direction.
11	Page change keys ↑ PAGE PAGE ↓	Two kinds of page change keys are described below. PAGE ↓ : This key is used to changeover the page on the CRT screen in the forward direction. ↑ PAGE : This key is used to changeover the page on the CRT screen in the reverse direction.

2.1.2 DPL/MDI Panel

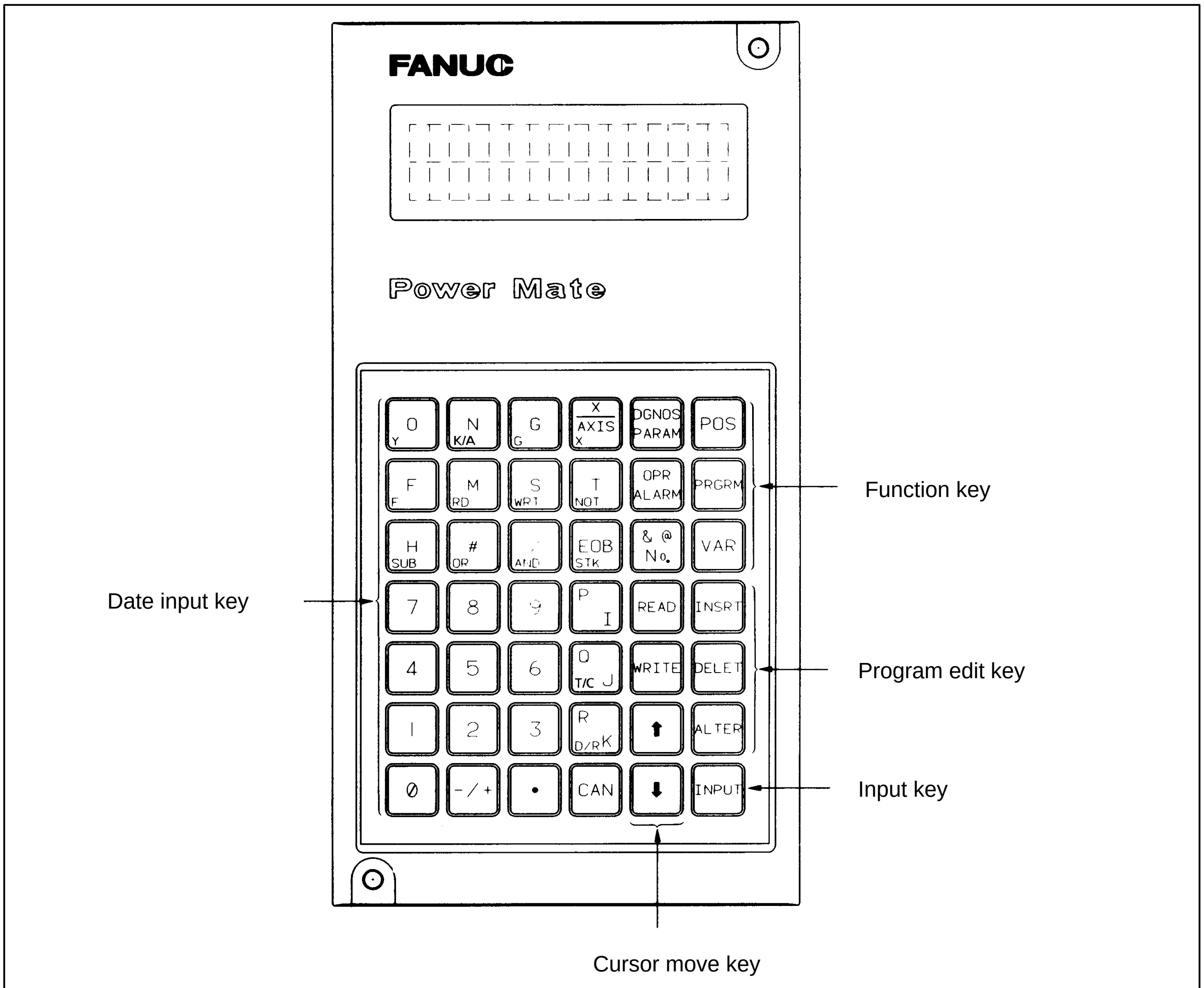


Fig. 2.1.2 DPL/MDI panel

(1) Function keys

Function keys indicate large items like chapters in a document.

<POS>

Indicates the current position.

<PRGRM>

Conducts the following:
In EDIT mode ...edits and displays the program in the memory
In automatic operation ...displays command value.

<VAR>

Used to display offset settings and to set and display macro variables.

<PARAM DGNOS>

Used to set and display parameter, diagnostic, and PMC parameter.

<ALARM>

Display of Alarm number and external message.

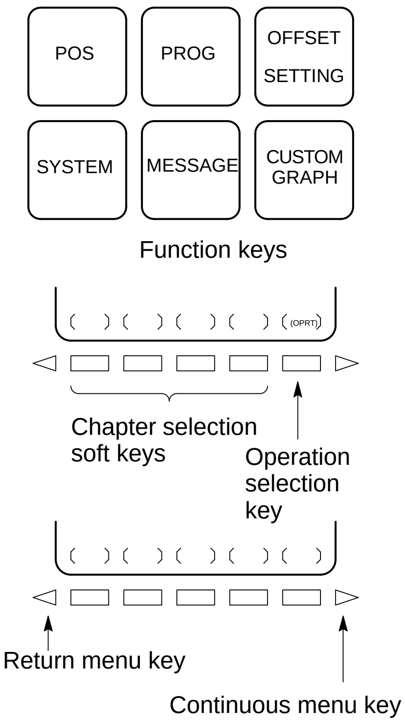
(2) Keyboard functions

Table 2.1.2 MDI Keyboard functions

Key	Functions
Address/numerical key	Press these keys to input alphabetic, numeric, and other characters.
INPUT (INPUT) key	When an address or a numerical key is pressed, the letter or the numeral is input once to the key input buffer, and it is displayed on the DPL. To input the data, press the INPUT key.
Cancel (CAN) key	Press this key to cancel character or sign input to the key input buffer. **(Example)** When the key input buffer displays N0001, N0001 is cancelled with this key. When an alarm is displayed, depressing CAN will reset the alarm message.
Cursor shift keys	There are two kinds of cursor shift key described below. ↓ : This key is used to shift the cursor a short distance in the forward direction. ↑ : This key is used to shift the cursor a short distance in the reverse direction.
READ (READ) key WRITE (WRITE) key	Press this key to actuate I/O device.

2.2 FUNCTION KEYS AND SOFT KEYS

2.2.1 General Screen Operations



1 Press a function key on the CRT/MDI panel. The chapter selection soft keys that belong to the selected function appear.

2 Press one of the chapter selection soft keys. The screen for the selected chapter appears. If the soft key for a target chapter is not displayed, press the continuous menu key (next–menu key).
In some cases, additional chapters can be selected within a chapter.

3 When the target chapter screen is displayed, press the operation selection key to display data to be manipulated.

4 To redisplay the chapter selection soft keys, press the return menu key.

The general screen display procedure is explained above. However, the actual display procedure varies from one screen to another. For details, see the description of individual operations.

2.2.2 Function Keys

Function keys are provided to select the type of screen to be displayed. The following function keys are provided on the CRT/MDI panel:

POS	Press this key to display the **position screen**.
PROG	Press this key to display the **program screen**.
OFFSET SETTING	Press this key to display the **offset/setting screen**.
SYSTEM	Press this key to display the **system screen**.
MESSAGE	Press this key to display the **message screen.**
CUSTOM GRAPH	This key is not usually used but may be used on some machines. When the use of this key is necessary, refer to the relevant manual supplied by the machine tool builder.

2.2.3 Soft Keys

To display a more detailed screen, press a function key followed by a soft key. Soft keys are also used for actual operations.
The following illustrates how soft key displays are changed by pressing each function key.

The symbols in the following figures mean as shown below :

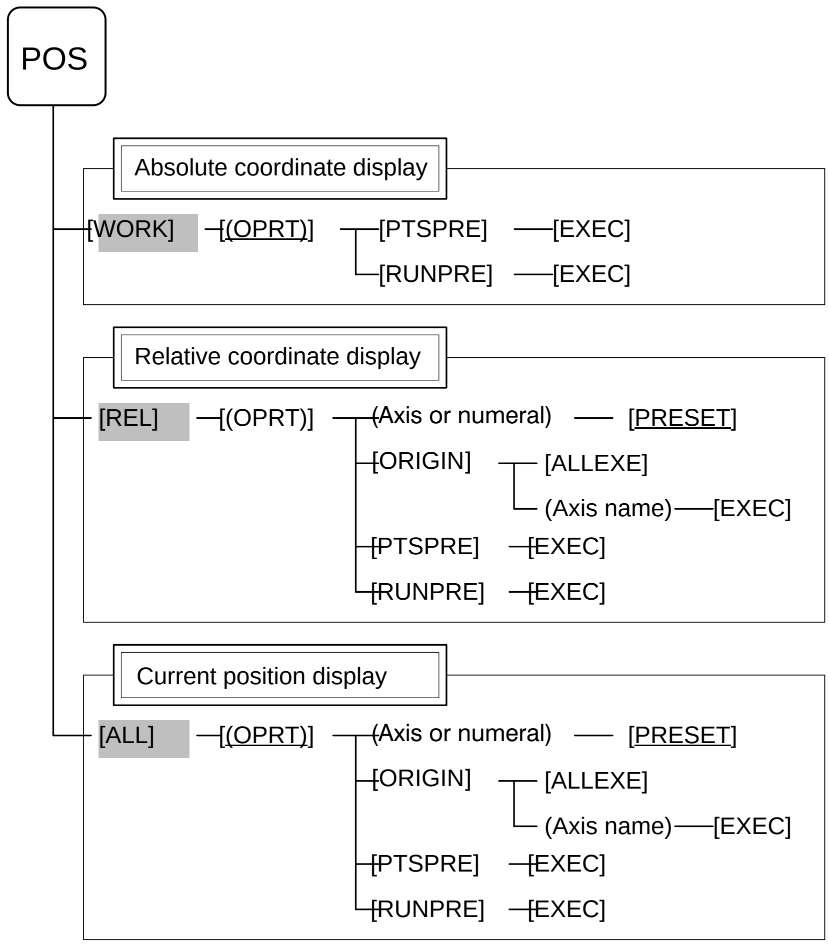
	:	Indicates screens
	:	Indicates a screen that can be displayed by pressing a function key(*1)
[ ]	:	Indicates a soft key(*2)
( )	:	Indicates input from the MDI panel.
[__]	:	Indicates a soft key displayed in green (or highlighted).
	:	Indicates the continuous menu key (rightmost soft key).

*1 Press function keys to switch between screens that are used frequently.

*2 Some soft keys are not displayed depending on the option configuration.

POSITION SCREEN

Soft key transition triggered by the function key POS



PROGRAM SCREEN

Soft key transition triggered by the function key PROG in the AUTO mode

1/2

PROG

Program display screen

[PRGRM] — [(OPRT)] — [BG–EDT] ⇒ See"When the soft key [BG–EDT] is pressed"
(1) — (O number) — [O SRH]
— (N number) — [N SRH]
— [REWIND]

Program check display screen

[CHECK] — [WORK] / [REL] — [(OPRT)] — [BG–EDT] ⇒ See"When the soft key [BG–EDT] is pressed"
— (O number) — [O SRH]
— (N number) — [N SRH]
— [REWIND]

Current block display screen

[CURRNT] — [(OPRT)] — [BG–EDT] ⇒ See"When the soft key [BG–EDT] is pressed"

Next block display screen

[NEXT] — [(OPRT)] — [BG–EDT] ⇒ See"When the soft key [BG–EDT] is pressed"

▷

[FL.SDL] — [PRGRM] ⇒ Return to (1) (Program display)

File directory display screen

— [DIR] — [(OPRT)] — [SELECT] — (File No.) — [F SET]
— [EXEC]

PROGRAM SCREEN

Soft key transition triggered by the function key PROG in the EDIT mode

1/2

PROG

Program display

- [PRGRM] — [(OPRT)]
  - [BG–EDT] ⇨ See"When the soft key [BG–EDT] is pressed"
  - (O number) — [O SRH]
  - (Address) — [SRH↓]
  - (Address) — [SRH↑]
  - [REWIND]
  - ▷
  - [F SRH]
    - [CAN]
    - (N number) — [EXEC]
  - [READ]
    - [CHAIN] (The cursor moves to the end of a program.)
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]
  - [PUNCH]
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]
  - [DELETE]
    - [CAN]
    - (N number) — [EXEC]
  - [EX–EDT]
    - [COPY]
      - [CRSR~] / [~CRSR] / [~BTTM] / [ALL] — (O number) — [EXEC]
    - [MOVE]
      - [CRSR~] / [~CRSR] / [~BTTM] / [ALL] — (O number) — [EXEC]
    - [MERGE]
      - [~CRSR] / [~BTTM] — (O number) — [EXEC]
    - [CHANGE] — (Address) — [BEFORE] — (Address) — [AFTER]
      - [SKIP]
      - [1–EXEC]
      - [EXEC]

(1)(Continued on the next page)

2/2

(1)

Program directory display

- [LIB]
  - [(OPRT)]
    - [BG–EDT] ⇨ See”When the soft key [BG–EDT] is pressed”
    - (O number) — [O SRH] ⇨ Return to the program
    - ▷
    - [READ]
      - [CHAIN]
      - [STOP]
      - [CAN]
      - (O number) — [EXEC]
    - [PUNCH]
      - [STOP]
      - [CAN]
      - (O number) — [EXEC]

▷

Floppy directory display

- [FLOPPY]
  - [PRGRM] ⇨ Return to the program
  - [DIR]
    - [(OPRT)]
      - [F SRH]
        - (Numeral) — [F SET]
        - [CAN]
        - [EXEC]
      - [READ]
        - (Numeral) — [F SET]
        - (Numeral) — [O SET]
        - [STOP]
        - [CAN]
        - [EXEC]
      - [PUNCH]
        - (Numeral) — [F SET]
        - (Numeral) — [O SET]
        - [STOP]
        - [CAN]
        - [EXEC]
      - [DELETE]
        - (Numeral) — [F SET]
        - [CAN]
        - [EXEC]

PROGRAM SCREEN

Soft key transition triggered by the function key PROG in the MDI mode

PROG

- Program display
  - [PRGRM] — [(OPRT)] — [BG–EDT] ⇨ See"When the soft key [BG–EDT] is pressed"
- Program input screen
  - [MDI] — [(OPRT)]
    - [BG–EDT] ⇨ See"When the soft key [BG–EDT] is pressed"
    - [START]
      - [CAN]
      - [EXEC]
    - (Address) — [SRH↓]
    - (Address) — [SRH↑]
    - [REWIND]
- Current block display screen
  - [CURRNT] — [(OPRT)] — [BG–EDT] ⇨ See"When the soft key [BG–EDT] is pressed"
- Next block display screen
  - [NEXT] — [(OPRT)] — [BG–EDT] ⇨ See"When the soft key [BG–EDT] is pressed"

PROGRAM SCREEN

Soft key transition triggered by the function key PROG in the JOG or REF mode

PROG

- Program display
  - [PRGRM] —— [(OPRT)] —— [BG–EDT] ⇒ See”When the soft key [BG–EDT] is pressed”
- Current block display screen
  - [CURRNT] —— [(OPRT)] —— [BG–EDT] ⇒ See”When the soft key [BG–EDT] is pressed”
- Next block display screen
  - [NEXT] —— [(OPRT)] —— [BG–EDT] ⇒ See ”When the soft key [BG–EDT] is pressed”

PROGRAM SCREEN

Soft key transition triggered by the function key PROG in the TJOG or TSTP mode

PROG

- Program display
  - [PRGRM] —— [(OPRT)] —— [BG–EDT] ⇒ See ”When the soft key [BG–EDT] is pressed”
- Program input screen
  - [MDI] —— [(OPRT)] ——
    - [BG–EDT] ⇒ See ”When the soft key [BG–EDT] is pressed”
    - (O number) —— [O SRH] ⇒ Return to the program
    - (Address) —— [SRH↓]
    - (Address) —— [SRH↑]
    - [REWIND]
- Program directory display
  - [LIB] —— [(OPRT)] ——
    - [BG–EDT] ⇒ See ”When the soft key [BG–EDT] is pressed”
    - (O number) —— [O SRH] ⇒ Return to the program

PROGRAM SCREEN

Soft key transition triggered by the function key PROG
(When the soft key [BG–EDT] is pressed in all modes)

1/2

PROG

Program display

- [PRGRM] — [(OPRT)]
  - [BG–END]
  - (O number) — [O SRH]
  - (Address) — [SRH↓]
  - (Address) — [SRH↑]
  - [REWIND]
  - ▷
  - [F SRH]
    - [CAN]
    - (N number) — [EXEC]
  - [READ]
    - [CHAIN] (The cursor moves to the end of a program.)
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]
  - [PUNCH]
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]
  - [DELETE]
    - [CAN]
    - (N number) — [EXEC]
  - [EX–EDT]
    - [COPY]
      - [CRSR~] / [~CRSR] / [~BTTM] / [ALL] — (O number) — [EXEC]
    - [MOVE]
      - [CRSR~] / [~CRSR] / [~BTTM] / [ALL] — (O number) — [EXEC]
    - [MERGE]
      - [~CRSR] / [~BTTM] — (O number) — [EXEC]
    - [CHANGE] — (Address) — [BEFORE] — (Address) — [AFTER]
      - [SKIP]
      - [1–EXEC]
      - [EXEC]

(1)(Continued on the next page)

2/2

(1)

Program directory display

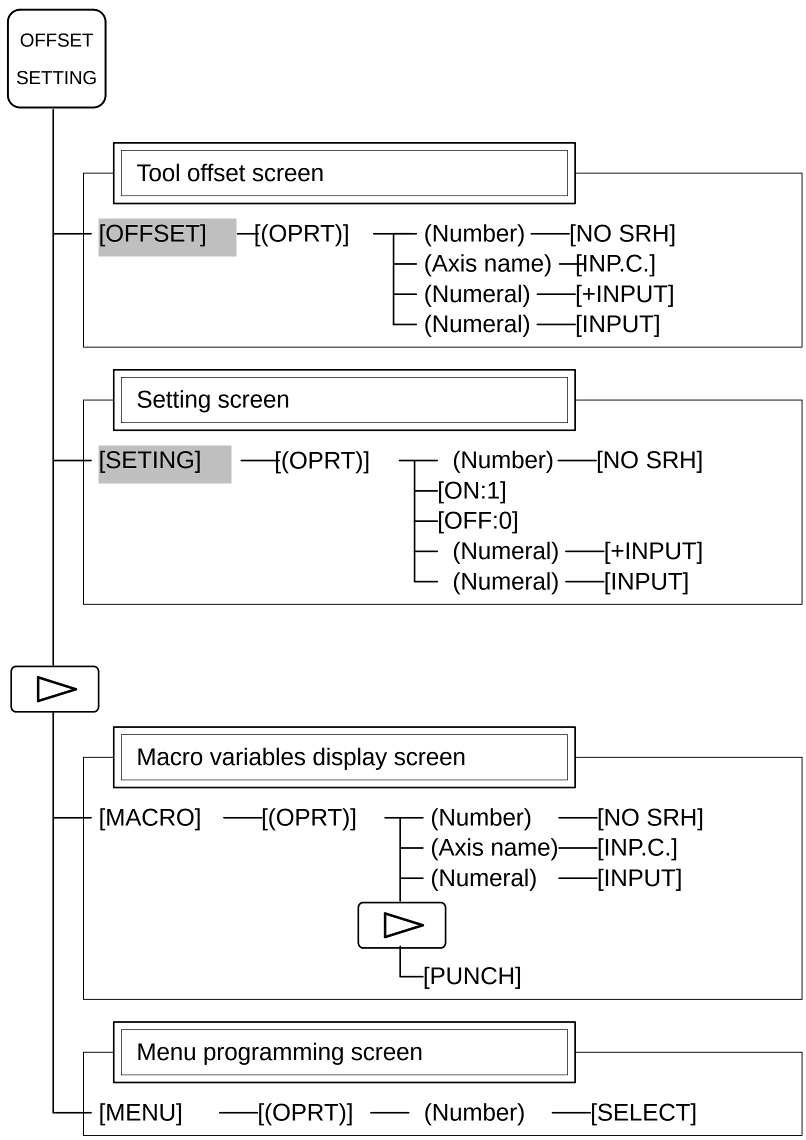
- [LIB] — [(OPRT)]
  - [BG–EDT]
  - (O number) — [O SRH] ⇒ Return to the program
  - [READ]
    - [CHAIN]
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]
  - [PUNCH]
    - [STOP]
    - [CAN]
    - (O number) — [EXEC]

Floppy directory display

- [FLOPPY]
  - [PRGRM] ⇒ Return to the program
  - [DIR] — [(OPRT)]
    - [F SRH]
      - (Numeral) — [F SET]
      - [CAN]
      - [EXEC]
    - [READ]
      - (Numeral) — [F SET]
      - (Numeral) — [O SET]
      - [STOP]
      - [CAN]
      - [EXEC]
    - [PUNCH]
      - (Numeral) — [F SET]
      - (Numeral) — [O SET]
      - [STOP]
      - [CAN]
      - [EXEC]
    - [DELETE]
      - (Numeral) — [F SET]
      - [CAN]
      - [EXEC]

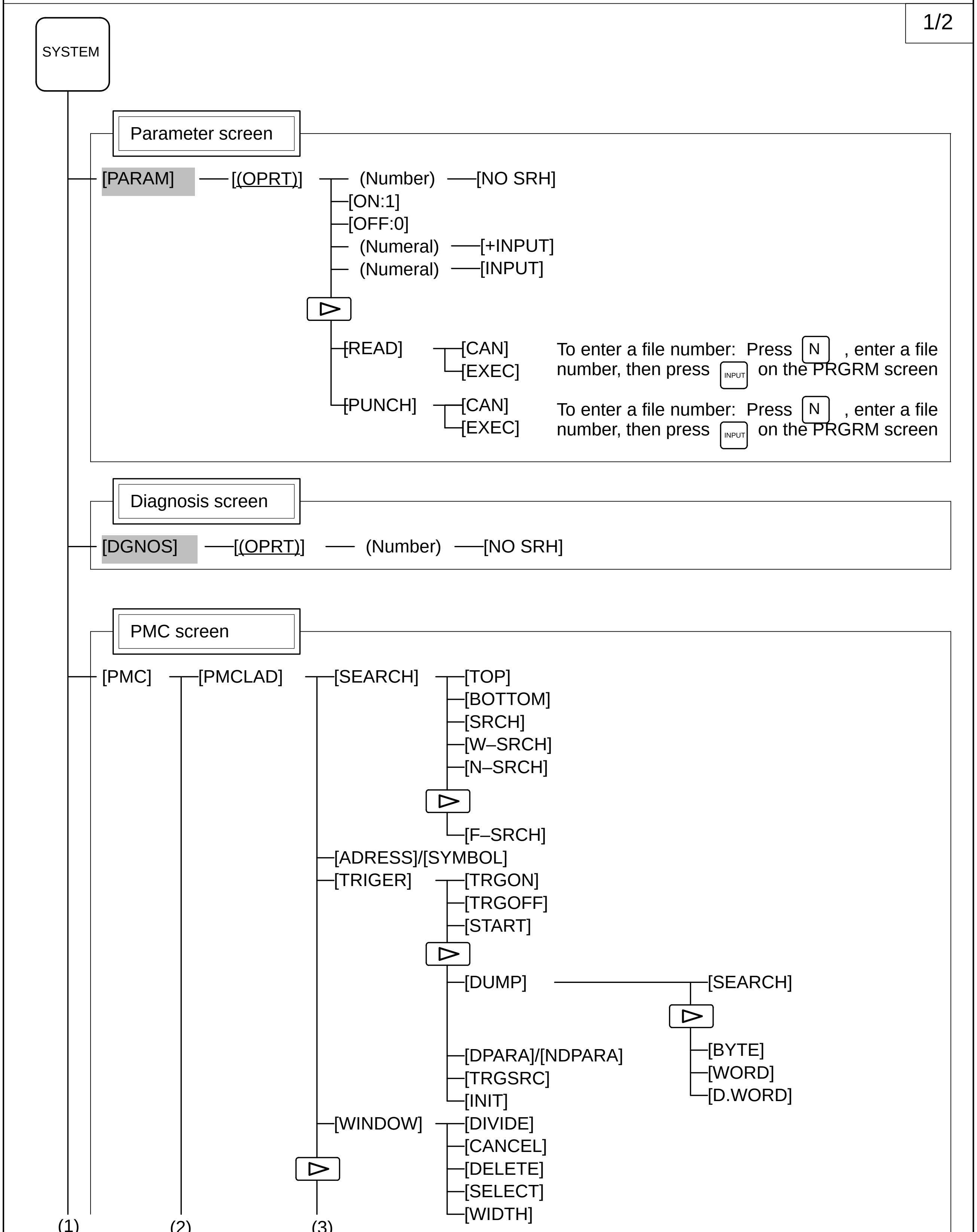
OFFSET/SETTING SCREEN

Soft key transition triggered by the function key OFFSET SETTING

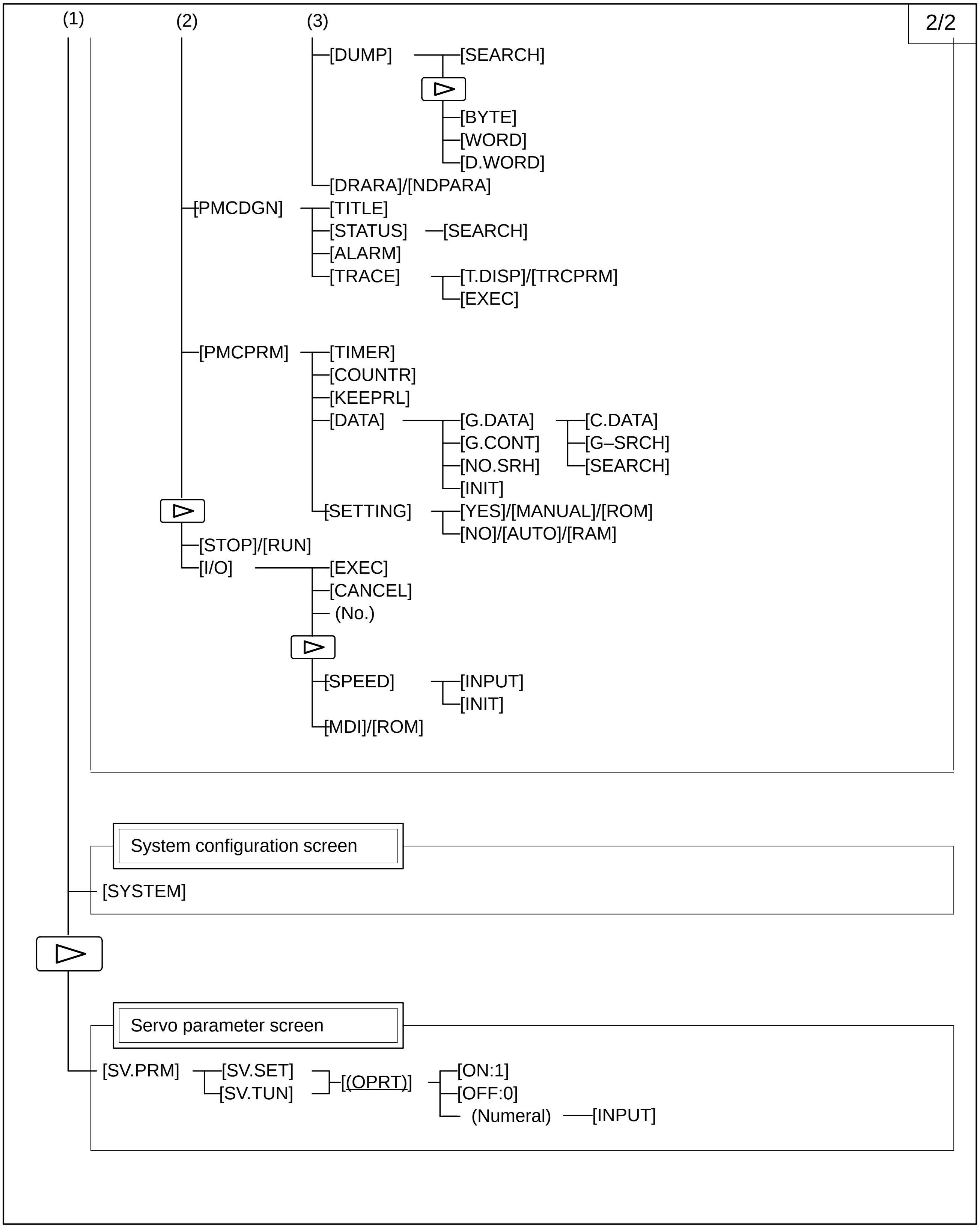


SYSTEM SCREEN Soft key transition triggered by the function key SYSTEM

1/2



(Continued on the next page)


2/2
(1)
(2)
(3)
[DUMP]
[SEARCH]
[BYTE]
[WORD]
[D.WORD]
[DRARA]/[NDPARA]
[PMCDGN]
[TITLE]
[STATUS]
[SEARCH]
[ALARM]
[TRACE]
[T.DISP]/[TRCPRM]
[EXEC]
[PMCPRM]
[TIMER]
[COUNTR]
[KEEPRL]
[DATA]
[G.DATA]
[C.DATA]
[G.CONT]
[G–SRCH]
[NO.SRH]
[SEARCH]
[INIT]
[SETTING]
[YES]/[MANUAL]/[ROM]
[NO]/[AUTO]/[RAM]
[STOP]/[RUN]
[I/O]
[EXEC]
[CANCEL]
(No.)
[SPEED]
[INPUT]
[INIT]
[MDI]/[ROM]
System configuration screen
[SYSTEM]
Servo parameter screen
[SV.PRM]
[SV.SET]
[SV.TUN]
[(OPRT)]
[ON:1]
[OFF:0]
(Numeral)
[INPUT]

MESSAGE SCREEN

Soft key transition triggered by the function key MESSAGE

MESSAGE

- Alarm display screen
  - [ALARM]
- Message display screen
  - [MSG]
- Alarm history screen
  - [HISTRY] —— [(OPRT)] —— [CLEAR]

HELP SCREEN

Soft key transition triggered by the function key HELP

HELP

- Alarm detail screen
  - [1 ALAM] —— [(OPRT)] —— [SELECT]
- Operation method screen
  - [2 OPR] —— [(OPRT)] —— [SELECT]
- Parameter table screen
  - [3 PARA]

2.2.4 Key Input and Input Buffer

When an address and a numerical key are pressed, the character corresponding to that key is input once into the key input buffer. The contents of the key input buffer is displayed at the bottom of the CRT screen.

In order to indicate that it is key input data, a ”>” symbol is displayed immediately in front of it. A ”_” is displayed at the end of the key input data indicating the input position of the next character.

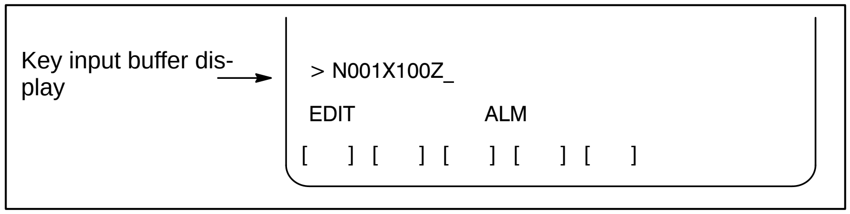


Fig. 2.2.4 Key input buffer display

To input the lower character of the keys that have two characters inscribed on them, first press the SHIFT key and then the key in question.

When the SHIFT key is pressed, ”_” indicating the next character input position changes to ”^”. Now lowercase characters can be entered (shift state).

When a character is input in shift status the shift status is canceled. Furthermore, if the SHIFT key is pressed in shift status, the shift status is canceled.

It is possible to input up to 32 characters at a time in the key input buffer.

Press the CAN key to cancel a character or symbol input in the key input buffer.

(Example)

When the key input buffer displays
>N001X100Z_
and the cancel CAN key is pressed, Z is canceled and
>N001X100_
is displayed.

2.2.5 Warning Messages

After a character or number has been input from the MDI panel, a data check is executed when INPUT key or a soft key is pressed. In the case of incorrect input data or the wrong operation a flashing warning message will be displayed on the status display line.

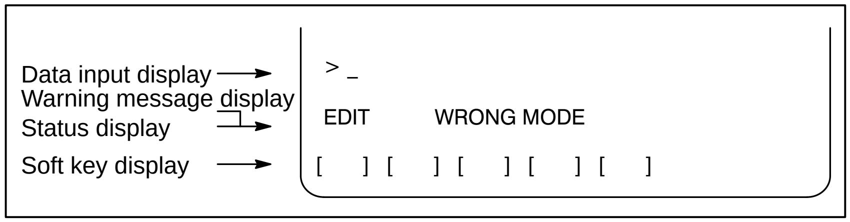


Fig. 2.2.5 Warning message display

Table2.2.5 Warning Messages

Warning message	Content
FORMAT ERROR	The format is incorrect.
WRITE PROTECT	Key input is invalid because of memory protect signal or the parameter is not write enabled.
DATA IS OUT OF RANGE	The value searched exceeds the permitted range.
TOO MANY DIGITS	The input value exceeds the permitted number of digits.
WRONG MODE	Parameter input is not possible in any mode other than MDI mode.
EDIT REJECTED	It is not possible to edit in the current CNC status.

2.3 EXTERNAL I/O DEVICES

Three types of external input/output devices are available. This section outlines each device. For details on these devices, refer to the corresponding manuals listed below.

Table 2.3(a) External I/O device

Device name	Usage	Max. storage capacity	Reference manual
FANUC Handy File	Easy–to–use, multi function input/output device. It is designed for FA equipment and uses floppy disks.	1.4MB (3600m)	B–61834E
FANUC Floppy Cassette	Input/output device. Uses floppy disks.	1MB (2500m)	B–66040E

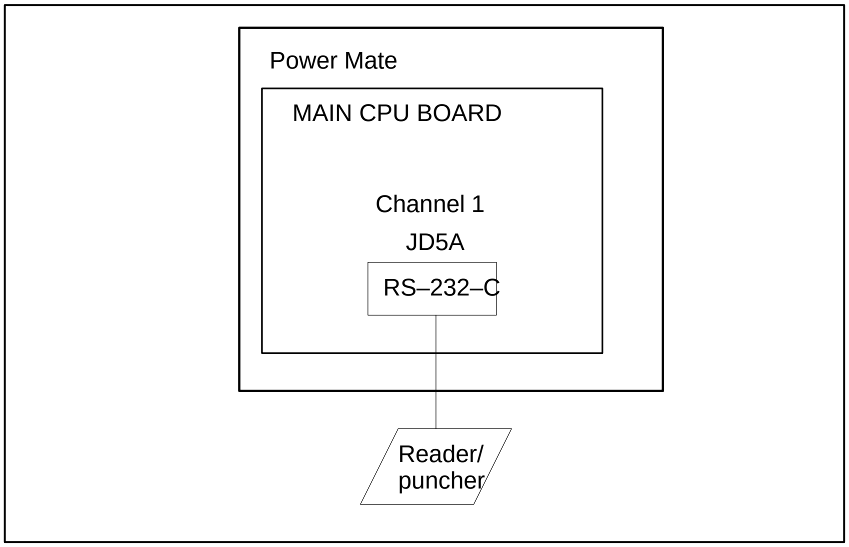
The following data can be input/output to or from external input/output devices:

1. **Programs**
2. **Offset data**
3. **Parameters**
4. **Custom macro common variables**

For how data is input and output, see Chapter 8.

Parameter

Before an external input/output device can be used, parameters must be set as follows.

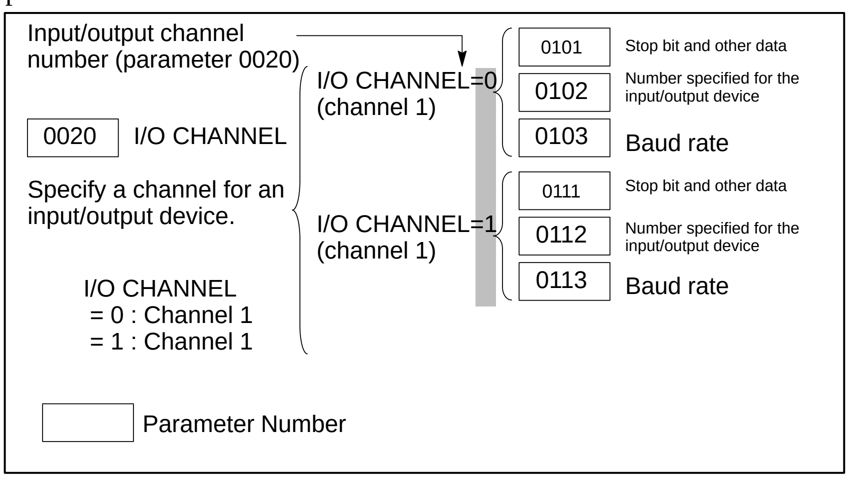


Power Mate has one channels of reader/punch interfaces. The input/output device to be used is specified by setting the channel connected to that device in setting parameter I/O CHANNEL.

The specified data, such as a baud rate and the number of stop bits, of an input/output device connected to a specific channel must be set in parameters for that channel in advance.

For channel 1, two combinations of parameters to specify the input/output device data are provided.

The following shows the interrelation between the reader/punch interface parameters for the channels.

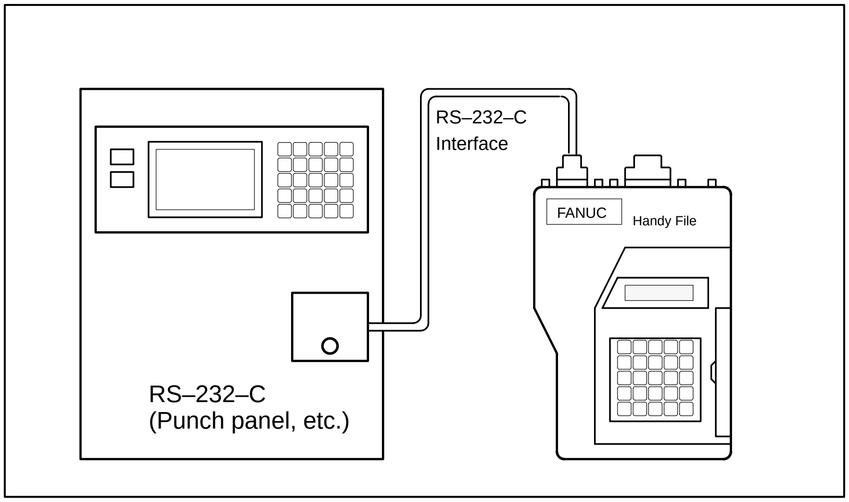


2.3.1 FANUC Handy File

The Handy File is an easy–to–use, multi function floppy disk input/output device designed for FA equipment. By operating the Handy File directly or remotely from a unit connected to the Handy File, programs can be transferred and edited.

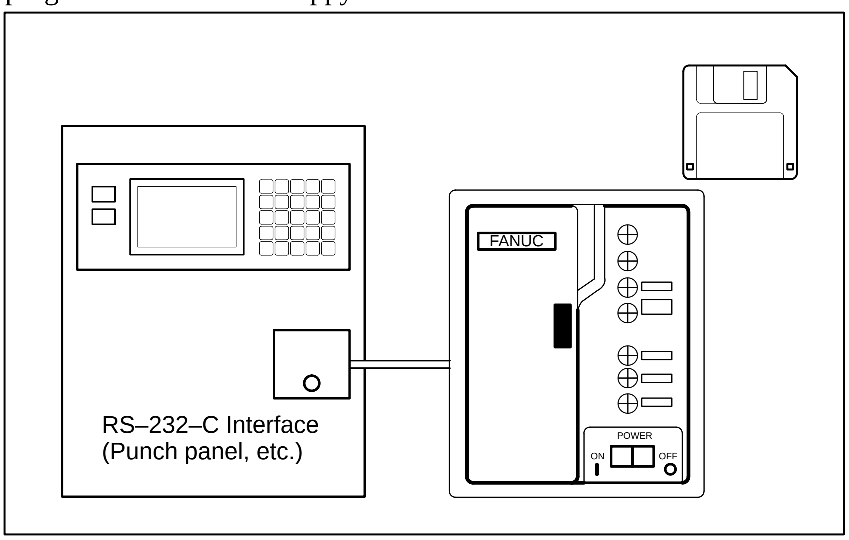
The Handy File uses 3.5–inch floppy disks, which do not have the problems of paper tape (i.e., noisy during input/output, easily broken, and bulky).

One or more programs (up to 1.44M bytes, which is equivalent to the memory capacity of 3600–m paper tape) can be stored on one floppy disk.



2.3.2 FANUC Floppy Cassette

When the Floppy Cassette is connected to the NC, machining programs stored in the NC can be saved on a Floppy Cassette, and machining programs saved in the Floppy Cassette can be transferred to the NC.



2.4 POWER ON/OFF

2.4.1 Turning on the Power

Procedure of turning on the power

Procedure

1 Check that the appearance of the controller machine tool is normal. (For example, check that front door and rear door are closed.)

2 Turn on the power according to the manual issued by the machine tool builder.

3 After the power is turned on, check that the position screen is displayed. If the screen shown in Section 7.1 is displayed, a system failure may have occurred.

```
ACTUAL POSITION(WORK)      O1000 N00010

   X      123.456
   Y      363.233
   Z        0.000

          PART COUNT       5
 RUN TIME    0H15M  CYCLE TIME   0H 0M38S
 ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ] [ REL ] [ ALL ] [    ] [ OPRT ]
```

4 Check that the fan motor is rotating.

> **WARNING**
> When pressing the <POWER ON> key, do not touch any other CRT/MDI panel keys. Until the positional or alarm screen is displayed, do not touch them. Some keys are used for the maintenance or special operation purpose. When they are pressed, unexpected operation may be caused.

2.4.2 Power Disconnection

Procedure for Power disconnection

1 Check that the LED indicating the cycle start is off on the operator's panel.

2 Check that all movable parts of the machine is stopping.

3 If an external input/output device such as the Handy File is connected to the contoroller, turn off the external input/output device.

4 Continue to press the POWER OFF pushbutton for about 5 seconds.

5 Refer to the machine tool builder's manual for turning off the power to the machine.

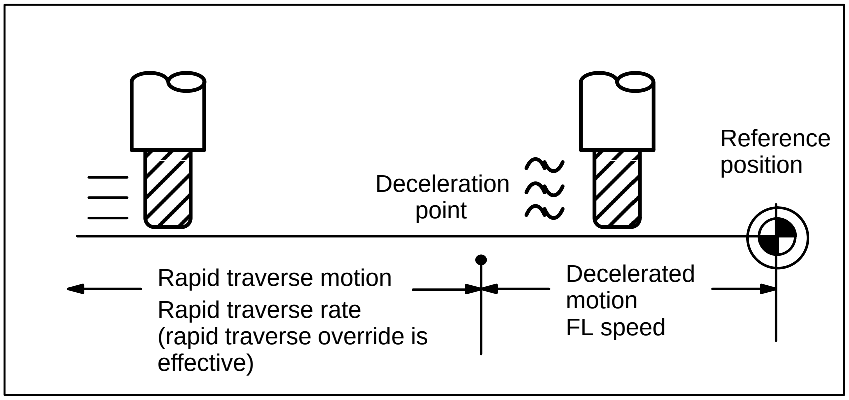
3 MANUAL OPERATION

MANUAL OPERATION are four kinds as follows :

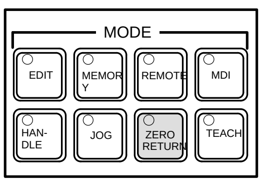
1.Manual reference position return
2.Jog feed
3.Incremental feed
4.Manual absolute on/off

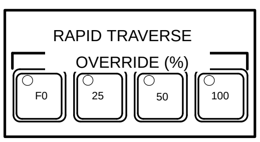
3.1 MANUAL REFERENCE POSITION RETURN

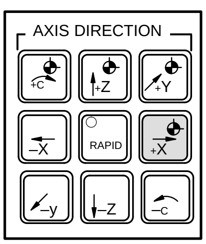
The tool is returned to the reference position as follows :
The tool is moved in the direction specified in parameter ZMI (bit 5 of No. 1006) for each axis with the reference position return switch on the machine operator's panel. The tool moves to the deceleration point at the rapid traverse rate, then moves to the reference position at the FL speed. The rapid traverse rate and FL speed are specified in parameters (No. 1420,1421, and 1425).
Four step rapid traverse override is effective during rapid traverse.
When the tool has returned to the reference position, the reference position return completion LED goes on. The tool generally moves along only a single axis, but can move along three axes simultane ously when specified so in parameter JAX(bit 0 of No.1002).



Procedure for Manual Reference Position Return





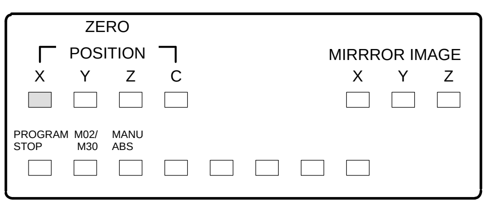


1 Press the reference position return switch, one of the mode selection swithces.

2 To decerease the feedrate, press a rapid traverse override switch. When the tool has returned to the reference position, the reference position return completion LED goes on.

3 Press the feed axis and direction selection switch corresponding to the axis and direction for reference position return. Continue pressing the switch until the tool returns to the reference position. The tool can be moved along three axes simultaneously when specified so in an appropriate parameter setting. The tool moves to the deceleration point at the rapid traverse rate, then moves to the reference position at the FL speed set in a parameter.

4 Perform the same operations for other axes, if necessary.
The above is an example. Refer to the appropriate manual provided by the machine tool builder for the actual operations.



Restrictions

- **Moving the tool again**

Once the REFERENCE POSITION RETURN COMPLETION LED lights at the completion of reference position return, the tool does not move unless the REFERENCE POSITION RETURN switch is turned off.

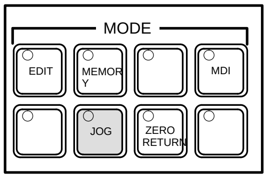
- **Reference position return completion LED**

The REFERENCE POSITION RETURN COMPLETION LED is extinguished by either of the following operations:
– Moving from the reference position.
– Entering an emergency stop state.

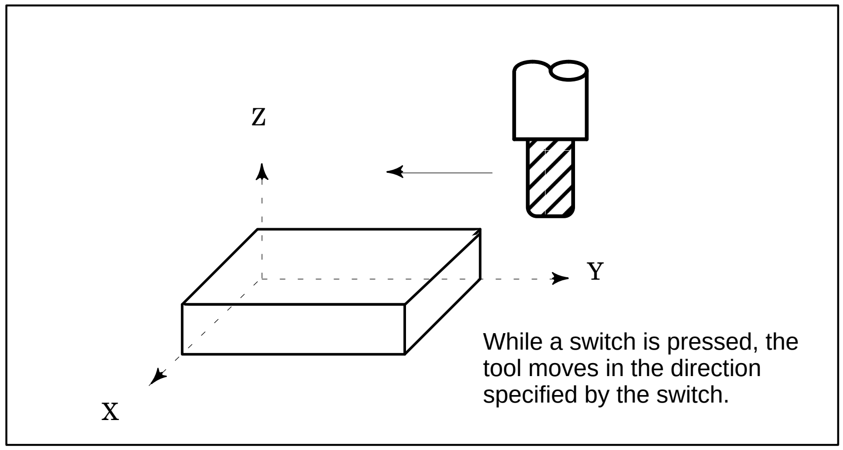
- **The distance to return to reference position**

For the distance (Not in the deceleration condition) to return the tool to the reference position, refer to the manual issued by the machine tool builder.

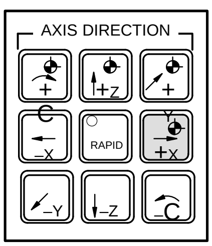
3.2 JOG FEED

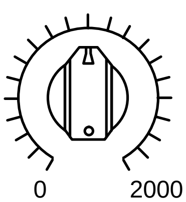


In the jog mode, pressing a feed axis and direction selection switch on the machine operator's panel continuously moves the tool along the selected axis in the selected direction.
The jog feedrate is specified in a parameter (No.1423)
The jog feedrate can be adjusted with the jog feedrate override dial.
Pressing the rapid traverse switch moves the tool at the rapid traverse feedrate regardless of the postiotion of the jog feedrate override dial.
Manual operation is allowed for one axis at a time. 3 axes can be selected at a time by parameter JAX (No.1002#0).

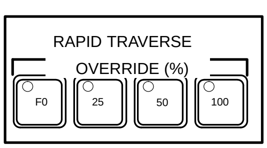


Procedure for Jog Feed





JOG FEED RATE **OVERRIDE**



1 Press the jog switch, one of the mode selection switches.

2 Press the feed axis and direction selection switch corresponding to the axis and direction the tool is to be moved. While the switch is pressed, the tool moves at the feedrate specified in a parameter (No. 1423). The tool stops when the switch is released.

3 The jog feedrate can be adjusted with the jog feedrate override dial.

4 Pressing the rapid traverse switch while pressing a feed axis and direction selection switch moves the tool at the rapid traverse rate while the rapid traverse switch is pressed. Rapid traverse override by the rapid traverse override switches is effective during rapid traverse.

The above is an example. Refer to the appropriate manual provided by the machine tool builder for the actual operations.

Restrictions

- **Acceleration/ deceleration for rapid traverse**

Feedrate, time constant and method of automatic acceleration/ deceleration for manual rapid traverse are the same as G00 in programmed command.

- **Change of modes**

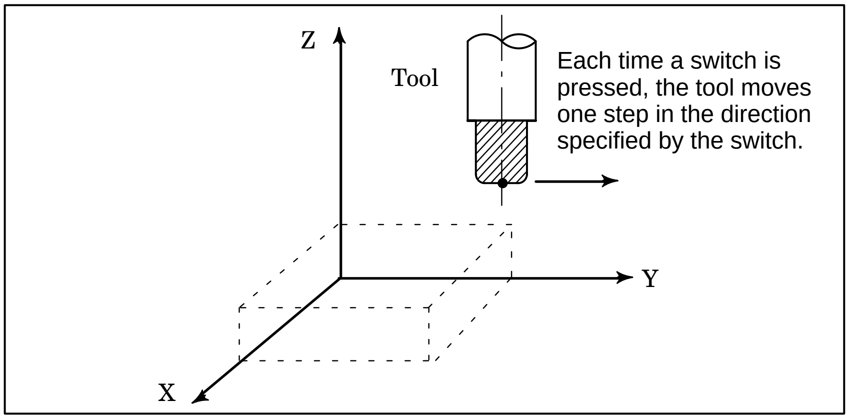
Changing the mode to the jog mode while pressing a feed axis and direction selection switch does not enable jog feed. To enable jog feed, enter the jog mode first, then press a feed axis and direction selection switch.

- **Rapid traverse prior to reference position return**

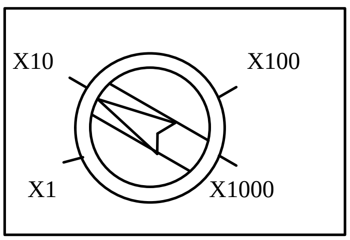
If reference position return is not performed after power–on, pushing RAPID TRAVERSE button does not actuate the rapid traverse but the remains at the JOG feedrate. This function can be disabled by setting parameter RPD (No.1401#01).

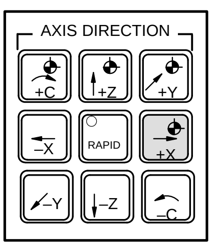
3.3 INCREMENTAL FEED

In the incremental (INC) mode, pressing a feed axis and direction selection switch on the machine operator's panel moves the tool one step along the selected axis in the selected direction. The minimum distance the tool is moved is the least input increment. Each step can be 10, 100, or 1000 times the least input increment.



Procedure for Incremental Feed





1 Press the INC switch, one of the mode selection switches.

2 Select the distance to be moved for each step with the magnification dial.

3 Press the feed axis and direction selection switch corresponding to the axis and direction the tool is to be moved. Each time a switch is pressed, the tool moves one step. The feedrate is the same as the jog feedrate.

4 Pressing the rapid traverse switch while pressing a feed axis and direction selection switch moves the tool at the rapid traverse rate. Rapid traverse override by the rapid traverse override switch is effective during rapid traverse.

The above is an example. Refer to the appropriate manual provided by the machine tool builder for the actual operations.

3.4 MANUAL ABSOLUTE ON AND OFF

Whether the distance the tool is moved by manual operation is added to the coordinates can be selected by turning the manual absolute switch on or off on the machine operator's panel. When the switch is turned on, the distance the tool is moved by manual operation is added to the coordinates. When the switch is turned off, the distance the tool is moved by manual operation is not added to the coordinates.

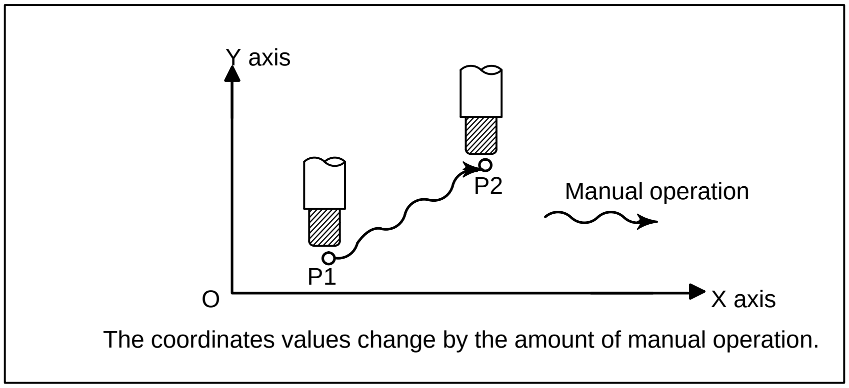


Fig. 3.4(a) Coordinates with the switch ON

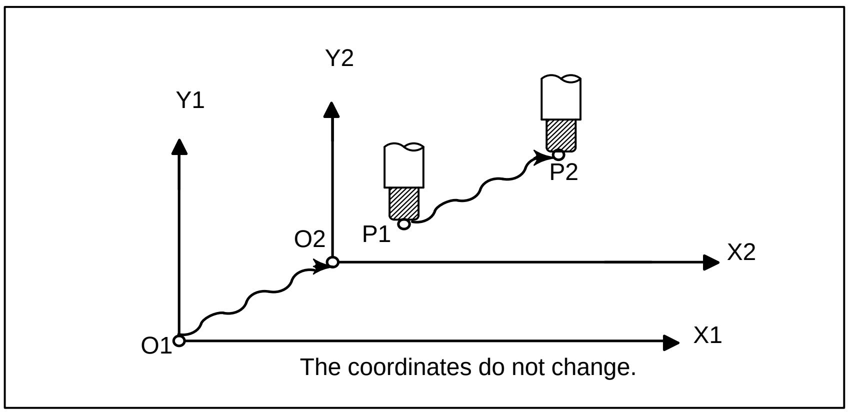


Fig. 3.4(b) Coordinates with the switch OFF

Explanation

The following describes the relation between manual operation and coordinates when the manual absolute switch is turned on or off, using a program example.

```
G01G90  X100.0Y100.0F010 ;        (1)
        X200.0Y150.0     ;        (2)
        X300.0Y200.0     ;        (3)
```

The subsequent figures use the following notation:

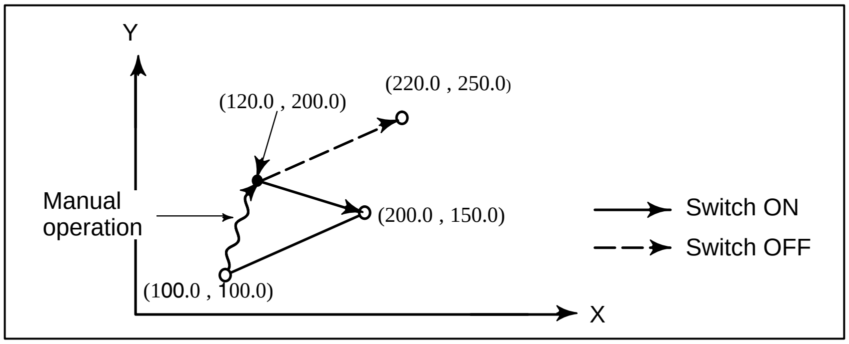
——➤ Movement of the tool when the switch is on

– – ➤ Movement of the tool when the switch is off

The coordinates after manual operation include the distance the tool is moved by the manual operation. When the switch is off, therefore, subtract the distance the tool is moved by the manual operation.

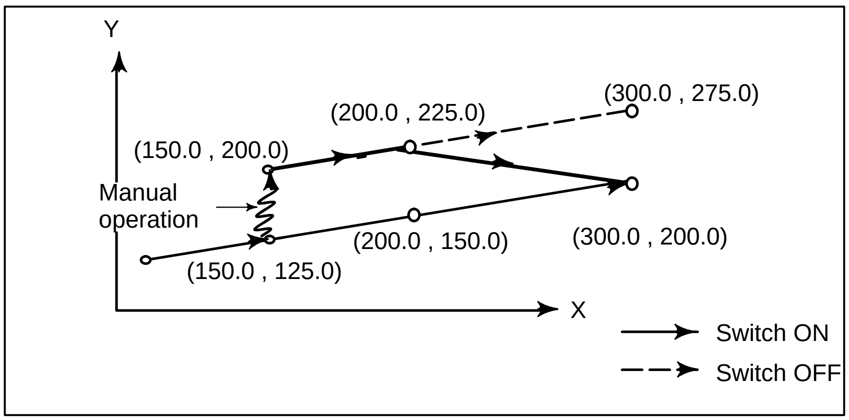
- **Manual operation after the end of block**

Coordinates when block (2) has been executed after manual operation (X–axis +20.0, Y–axis +100.0) at the end of movement of block (1).



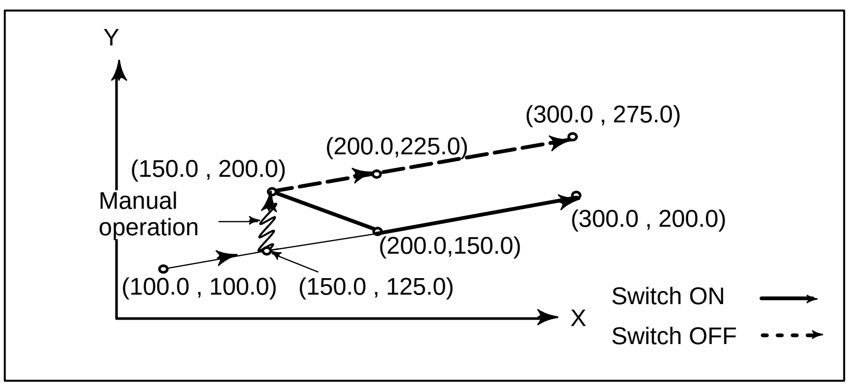
- **Manual operation after a feed hold**

Coordinates when the feed hold button is pressed while block (2) is being executed, manual operation (Y–axis + 75.0) is performed, and the cycle start button is pressed and released



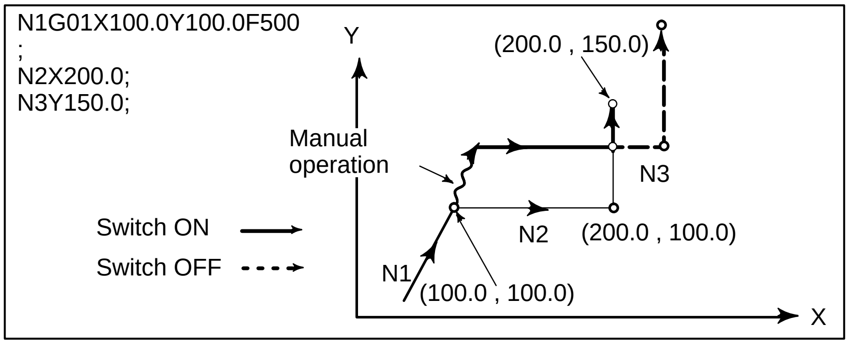
- **When reset after a manual operation following a feed hold**

Coordinates when the feed hold button is pressed while block (2) is being executed, manual operation (Y–axis +75.0) is performed, the control unit is reset with the RESET button, and block (2) is read again



- **When a movement command in the next block is only one axis**

When there is only one axis in the following command, only the commanded axis returns.



- **When the next move block is an incremental**

When the following commands are incremental commands, operation is the same as when the switch is OFF.

4 AUTOMATIC OPERATION

Programmed operation of a machine is referred to as automatic operation. This chapter explains the following types of automatic operation:

·**AUTO OPERATION**

Operation by executing a program registered in CNC memory

·**MDI OPERATION**

Operation by executing a program entered from the MDI panel

·**MIRROR IMAGE**

Function for enabling mirror–image movement along an axis during automatic operation

4.1 AUTO OPERATION

Programs are registered in memory in advance. When one of these programs is selected and the cycle start switch on the machine operator's panel is pressed, automatic operation starts, and the cycle start LED goes on.
When the feed hold switch on the machine operator's panel is pressed during automatic operation, automatic operation is stopped temporarily. When the cycle start switch is pressed again, automatic operation is restarted.
When the reset switch on the CRT/MDI panel is pressed, automatic operation terminates and the reset state is entered.
The following procedure is given as an example. For actual operation, refer to the manual supplied by the machine tool builder.

Procedure for AUTO Operation

Procedure

1 Press the **AUTO** mode selection switch.

2 Select a program from the registered programs. To do this, follow the steps below.

2–1 Press [PROG] to display the program screen.

2–2 Press address [O] .

2–3 Enter a program number using the numeric keys.

2–4 Press the **[O SRH]** soft key for the CRT/MDI.
(Press the cursor <↓> for the DPL/MDI.)

3 Press the cycle start switch on the machine operator's panel. Automatic operation starts, and the cycle start LED goes on. When automatic operation terminates, the cycle start LED goes off.

4 To stop or cancel AUTO operation midway through, follow the steps below.

a. Stopping AUTO operation
Press the feed hold switch on the machine operator's panel. The feed hold LED goes on and the cycle start LED goes off. The machine responds as follows:
(i) When the machine was moving, feed operation decelerates and stops.
(ii) When dwell was being performed, dwell is stopped.
(iii) When M was being executed, the operation is stopped after M is finished.
When the cycle start switch on the machine operator's panel is pressed while the feed hold LED is on, machine operation restarts.

b. Terminating AUTO operation
Press the [RESET] key on the CRT/MDI panel.
Automatic operation is terminated and the reset state is entered. When a reset is applied during movement, movement decelerates then stops.

Explanation

AUTO operation

After AUTO operation is started, the following are executed:

(1) A one–block command is read from the specified program.

(2) The block command is decoded.

(3) The command execution is started.

(4) The command in the next block is read.

(5) Buffering is executed. That is, the command is decoded to allow immediate execution.

(6) Immediately after the preceding block is executed, execution of the next block can be started. This is because buffering has been executed.

(7) Hereafter, AUTO operation can be executed by repeating the steps (4) to (6).

Stopping and terminating AUTO operation

AUTO operation can be stopped using one of two methods: Specify a stop command, or press a key on the machine operator's panel.

– The stop commands include M00 (program stop), M01 (optional stop), and M02 and M30 (program end).

– There are two keys to stop AUTO operation: The feed hold key and reset key.

• Program stop (M00)

AUTO operation is stopped after a block containing M00 is executed. When the program is stopped, all existing modal information remains unchanged as in single block operation. The AUTO operation can be restarted by pressing the cycle start button. Operation may vary depending on the machine tool builder. Refer to the manual supplied by the machine tool builder.

• Optional stop (M01)

Similarly to M00, AUTO operation is stopped after a block containing M01 is executed. This code is only effective when the Optional Stop switch on the machine operator's panel is set to ON. Operation may vary depending on the machine tool builder. Refer to the manual supplied by the machine tool builder.

• Program end (M02, M30)

When M02 or M30 (specified at the end of the main program) is read, AUTO operation is terminated and the reset state is entered.
In some machines, M30 returns control to the top of the program. For details, refer to the manual supplied by the machine tool builder.

• Feed hold

When Feed Hold button on the operator's panel is pressed during AUTO operation, the tool decelerates to a stop at a time.

• Reset

Automatic operation can be stopped and the system can be made to the reset state by using RESET key on the CRT/MDI panel or external reset signal. When reset operation is applied to the system during a tool moving status, the motion is slowed down then stops.

• Optional block skip

When the optional block skip switch on the machine operator's panel is turned on, blocks containing a slash (/) are ignored.

4.2 MDI OPERATION

In the **MDI** mode, a program consisting of up to 10 lines can be created in the same format as normal programs and executed from the MDI panel.
MDI operation is used for simple test operations.
The following procedure is given as an example. For actual operation, refer to the manual supplied by the machine tool builder.

Procedure for MDI Operation

Procedure for CRT/MDI

1 Press the **MDI** mode selection switch.

2 Press the PROG function key on the CRT/MDI panel to select the program screen. The following screen appears:

```
PROGRAM ( MDI )                        0010  00002

 O0000;



 G00  G90  G94
 G17  G22  G21  G49
           H M

   F

 >_
 MDI   ****   ***   ***
[ PRGRM ][ MDI ][ CURRNT ][ NEXT ][ (OPRT) ]
```

Program number O0000 is entered automatically.

3 Prepare a program to be executed by an operation similar to normal program editing. M99 specified in the last block can return control to the beginning of the program after operation ends. Word insertion, modification, deletion, word search, address search, and program search are available for programs created in the MDI mode. For program editing, see Chapter 9.

4 To entirely erase a program created in MDI mode,use one of the following methods:

a. Enter address O , then press the DELETE key on the MDI panel.

b. Alternatively, press the RESET key. In this case, set bit 7 of parameter 3203 to 1 in advance.

5 To execute a program, set the cursor on the head of the program. (Start from an intermediate point is possible.) Push Cycle Start button on the operator's panel. By this action, the prepared program will start. When the program end (M02, M30) or ER(%) is executed, the prepared program will be automatically erased and the operation will end. By command of M99, control returns to the head of the prepared program.

```
PROGRAM ( MDI )                          O0001  N00003
O0000  G00  X100.0  Y200. ;
 M03 ;
 G01  Z120.0  F500 ;
 M93  P9010 ;
 G00  Z0.0 ;
 %

 G00   G90   G94
 G17   G22   G21   G49
             H M

   F
 >_
 MDI    ****   ***   ***
[ PRGRM ][  MDI  ][ CURRNT ][ NEXT ][ (OPRT) ]
```

6 To stop or terminate MDI operation in midway through, follow the steps below.

a. Stopping MDI operation

Press the feed hold switch on the machine operator's panel. The feed hold LED goes on and the cycle start LED goes off. The machine responds as follows:

(i) When the machine was moving, feed operation decelerates and stops.

(ii) When dwell was being performed, dwell is stopped.

(iii) When M was being executed, the operation is stopped after M is finished.

When the cycle start switch on the machine operator's panel is pressed, machine operation restarts.

b. Terminating MDI operation

Press the RESET key on the CRT/MDI panel.

Automatic operation is terminated and the reset state is entered. When a reset is applied during movement, movement decelerates then stops.

Procedure for DPL/MDI

Place the system in MDI mode and select the program screen. The following screen appears.

```
>   O0100
    %
```

The program number, "O0000", is automatically inserted. Create the program to be executed according to the same procedure as in normal program editing. Resetting the system does not erase the newly created program. To erase it, perform the operation described in Note 3.
Continuous-state information must be checked with diagnostic data.

NOTE

1 Registered programs cannot be edited ; that is, registered program cannot be newly registered, deleted, punched, or collated.
2 A program can be created in up to six blocks. If the number of characters in a block is large (about 30 characters or more), the limit may be less than six blocks.
3 To erase all created programs, input : O <DELET>
Alternatively, set parameter MCL (bit 7 of parameter No.3203) to 1. All programs are then erased when the system is reset.

Explanation

The previous explanation of how to execute and stop AUTO operation also applies to MDI operation, except that in MDI operation, M30 does not return control to the beginning of the program (M99 performs this function).

• Erasing the program

Programs prepared in the **MDI** mode will be erased in the following cases:
· In MDI operation, if M02, M30 or ER(%) is executed.
· In **AUTO** mode, if memory operation is performed.
· In **EDIT** mode, if any editing is performed.
· Background editing is performed.

• Restart

After the editing operation during the stop of MDI operation was done, operation starts from the current cursor position.

Restrictions

• Program registration

Programs created in MDI mode cannot be registered.

• Number of lines in a program

A program can have as many lines as can fit on one page of the CRT screen.
A program consisting of up to six lines can be created. When parameter MDL (No. 3107 #7) is set to 0 to specify a mode that suppresses the display of continuous–state information, a program of up to 10 lines can be created.
If the created program exceeds the specified number of lines, % (ER) is deleted (prevents insertion and modification).

• Subprogram nesting

Calls to subprograms (M98) can be specified in a program created in the MDI mode. This means that a program registered in memory can be called and executed during MDI operation. In addition to the main program executed by automatic operation, up to two levels of subprogram nesting are allowed (when the custom macro option is provided, up to four levels are allowed).

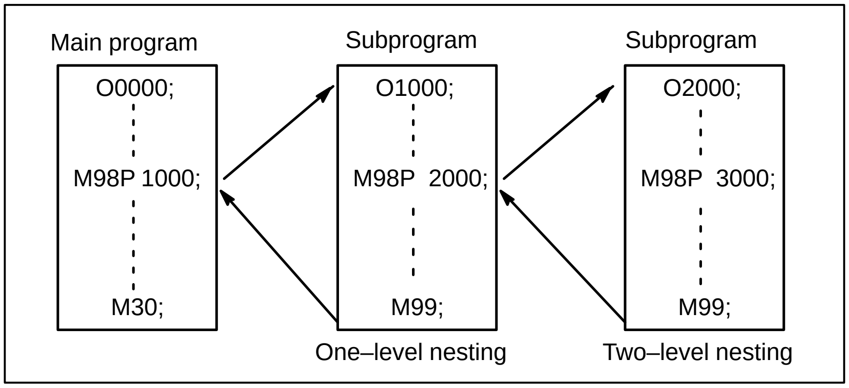


Fig. 4.2 (a) Nesting Level of Subprograms Called from the MDI Program

- **Macro call**

When the custom macro option is provided, macro programs can also be created, called, and executed in the **MDI** mode. However, macro call commands cannot be executed when the mode is changed to **MDI** mode after AUTO operation is stopped during execution of a subprogram.

- **Memory area**

When a program is created in the **MDI** mode, an empty area in program memory is used. If program memory is full, no programs can be created in the **MDI** mode.

4.3 MIRROR IMAGE

During automatic operation, the mirror image function can be used for movement along an axis. To use this function, set the mirror image switch to ON on the machine operator's panel, or set the mirror image setting to ON from the CRT/MDI panel.

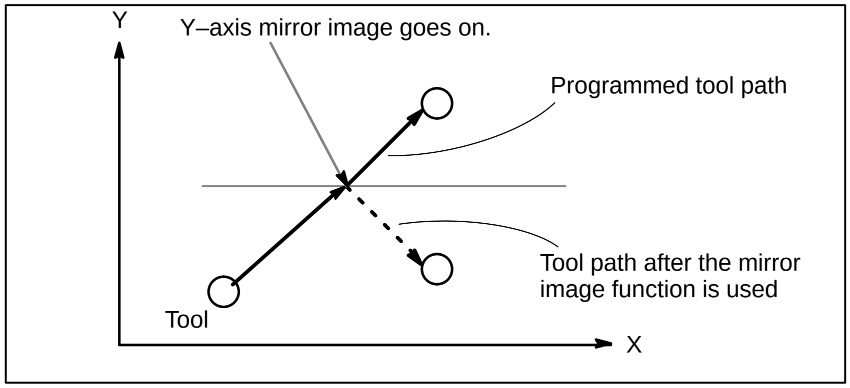


Fig 4.3 (a) Mirror Image

Procedure

The following procedure is given as an example. For actual operation, refer to the manual supplied by the machine tool builder.

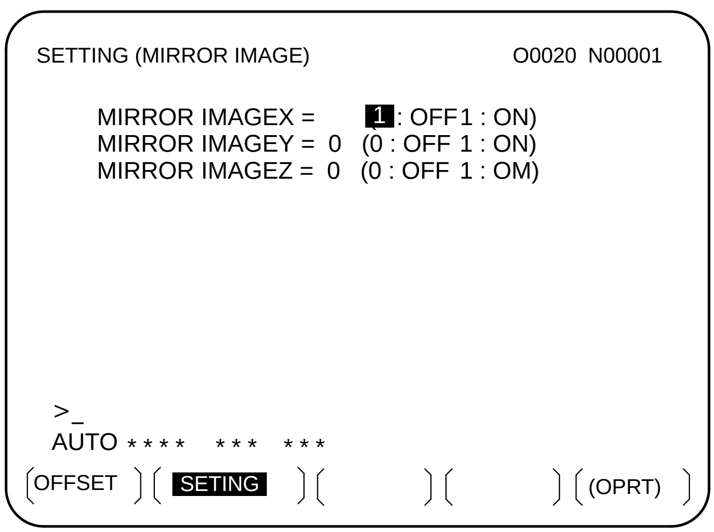
1 Press the single block switch to stop automatic operation. When the mirror image function is used from the begining of operation, this step is omitted.

2 Press the mirror image switch for the target axis on the machine operator's panel.
Alternatively, turn on the mirror image setting by following the steps below:

2–1 Set the **MDI** mode.

2–2 Press the OFFSET SETTING function key.

2–3 Press the **[SETING]** soft key for chapter selection to display the setting screen.



2–4 Move the cursor to the mirror image setting position, then set the target axis to 1.

3 Enter an automatic operation mode (AUTO mode or MDI mode), then press the cycle start button to start automatic operation.

Explanations

- The mirror image function can also be turned on and off by setting bit 0 of parameter 0012 to 1 (on) or 0 (off).
- For the mirror image switches, refer to the manual supplied by the machine tool builder.

Restrictions

The direction of movement during manual operation, the direction of movement from an intermidiate point to the reference position during automatic reference position return, cannot be reversed.

5 TEST OPERATION

The following functions are used to check before actual operation of machine whether the machine operates as specified by the created program.

1. Machine Lock and Auxiliary Function Lock
2. Feedrate Override
3. Rapid Traverse Override
4. Dry Run
5. Single Block

5.1 MACHINE LOCK AND AUXILIARY FUNCTION LOCK

To display the change in the position without moving the tool, use machine lock.
All–axis machine lock, which stops the movement along all axes. In addition, auxiliary function lock, which disables M command, is available for checking a program together with machine lock.

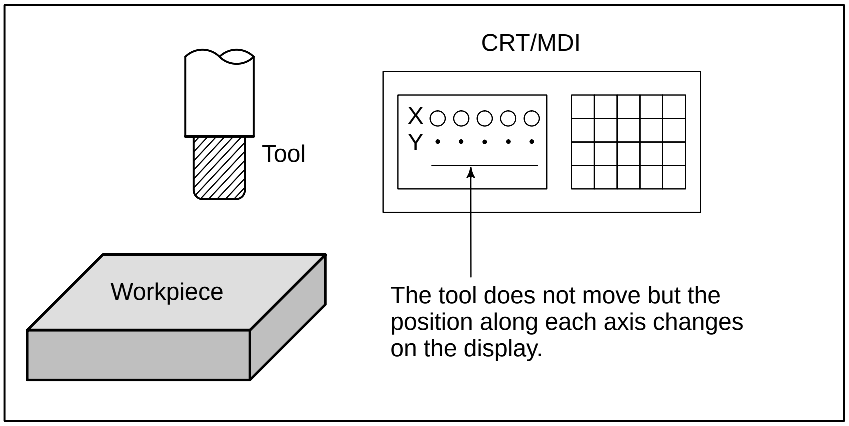


Fig. 5.1 Machine lock

Procedure for Machine Lock and Auxiliary Function Lock

- **Machine Lock**

Press the machine lock switch on the operator's panel. The tool does not move but the position along each axis changes on the display as if the tool were moving.
Refer to the appropriate manual provided by the machine tool builder for machine lock.

- **Auxiliary Function Lock**

Press the auxiliary function lock switch on the operator's panel. M code are disabled and not executed. Refer to the appropriate manual provided by the machine tool builder for auxiliary function lock.

Restrictions

- **M command by only machine lock**

M command are executed in the machine lock state.

- **Reference position return under Machine Lock**

When a G27, G28, or G30 command is issued in the machine lock state, the command is accepted but the tool does not move to the reference position and the reference position return LED does not go on.

- **M codes not locked by auxiliary function lock**

M00, M01, M02, M30, M98, and M99 commands are executed even in the auxiliary function lock state.

5.2 FEEDRATE OVERRIDE

A programmed feedrate can be reduced or increased by a percentage (%) selected by the override dial.This feature is used to check a program.
For example, when a feedrate of 100 mm/min is specified in the program, setting the override dial to 50% moves the tool at 50 mm/min.

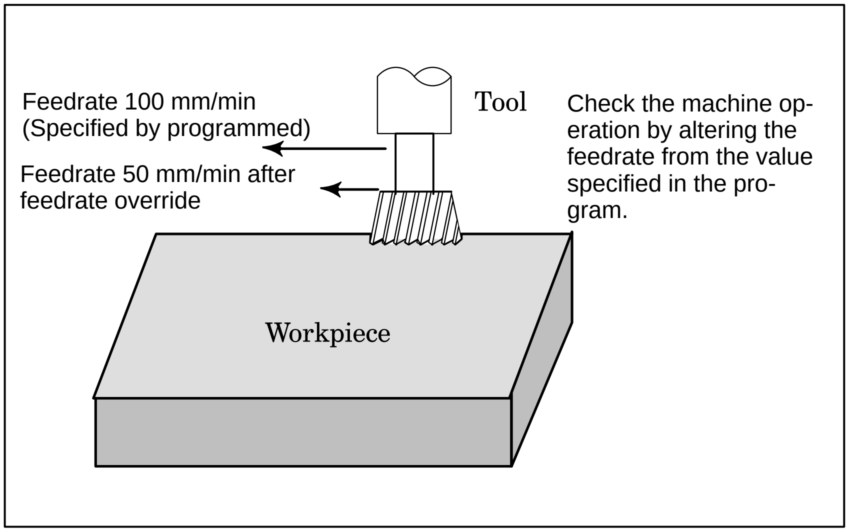
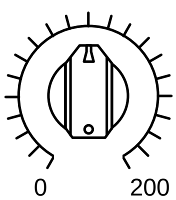


Fig. 5.2 Feedrate override

Procedure for Feedrate Override



JOG FEED RATE OVERRIDE

Set the feedrate override dial to the desired percentage (%) on the machine operator's panel, before or during automatic operation.
On some machines, the same dial is used for the feedrate override dial and jog feedrate dial. Refer to the appropriate manual provided by the machine tool builder for feedrate override.

Restrictions

- **Override Range**

The override that can be specified ranges from 0 to 254%. For individual machines, the range depends on the specifications of the machine tool builder.

5.3 RAPID TRAVERSE OVERRIDE

An override of four steps (F0, 25%, 50%, and 100%) can be applied to the rapid traverse rate. F0 is set by a parameter (No. 1421).

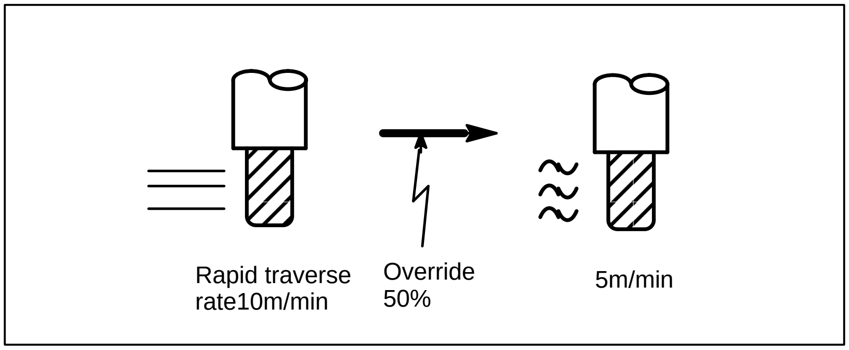
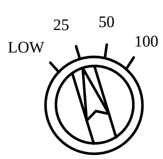


Fig. 5.3 Rapid traverse override

Rapid Traverse Override

Procedure



Rapid traverse override

Select one of the four feedrates with the rapid traverse override switch during rapid traverse. Refer to the appropriate manual provided by the machine tool builder for rapid traverse override.

Explanation

The following types of rapid traverse are available. Rapid traverse override can be applied for each of them.

1) Rapid traverse by G00.
2) Rapid traverse in G27, G28 and G30.
3) Manual rapid traverse.

5.4 DRY RUN

The tool is moved at the feedrate specified by a parameter regardless of the feedrate specified in the program. This function is used for checking the movement of the tool.

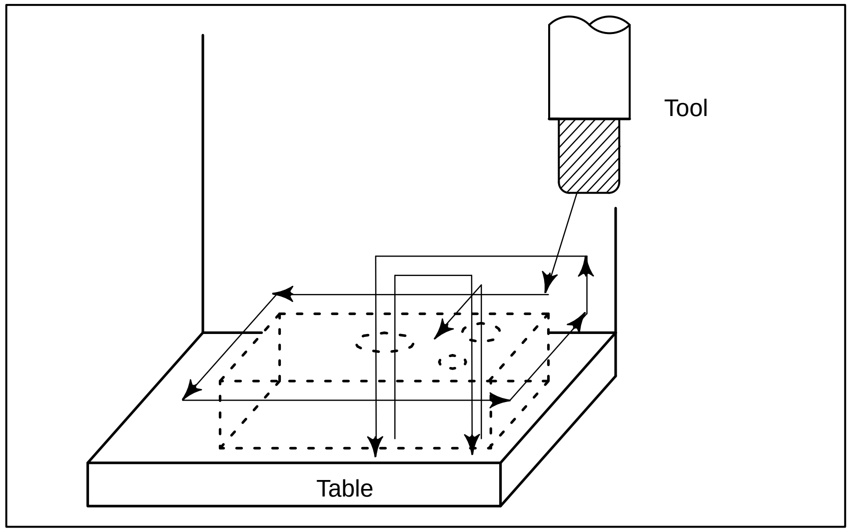


Fig. 5.4 Dry run

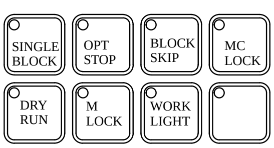
Procedure for Dry Run

Procedure

Press the dry run switch on the machine operator's panel during automatic operation.
The tool moves at the feedrate specified in a parameter. The rapid traverse switch can also be used for changing the feedrate.
Refer to the appropriate manual provided by the machine tool builder for dry run.

Explanation

- **Dry run feedrate**



The dry run feedrate changes as shown in the table below according to the rapid traverse switch and parameters.

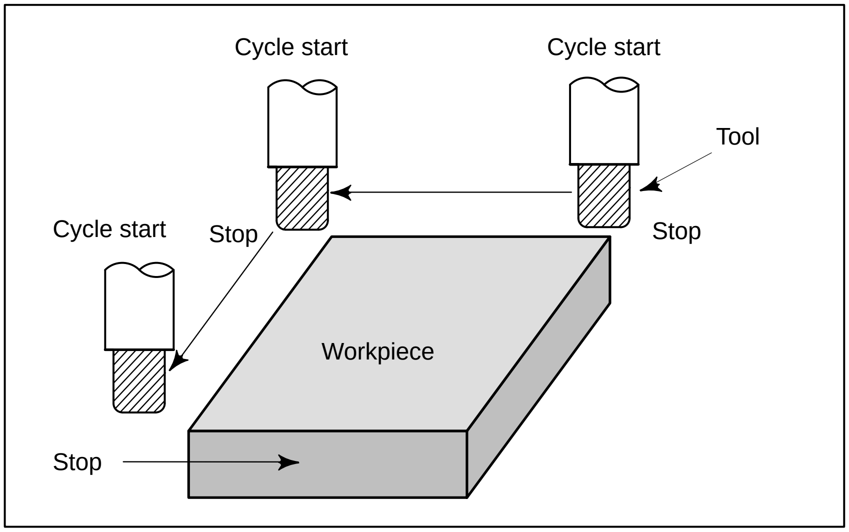
Rapid traverse button	Program command	
	Rapid traverse	Feed
ON	Rapid traverse rate	Dry run feedrate × Max.JV
OFF	Dry run speed × JV,or rapid traverse rate *1)	Dry run feedrate × JV

Max. feedrate of command . . . . Setting by parameter No.1422
Rapid traverse rate . . . . Setting by parameter No.1420
Dry run feedrate . . . . Setting by parameter No.1410
JV: Jog feedrate override

*1:Dry run feedrate x JV when parameter RDR (bit 6 of No. 1401) is 1.
Rapid traverse rate when parameter RDR is 0.

5.5 SINGLE BLOCK

Pressing the single block switch starts the single block mode. When the cycle start button is pressed in the single block mode, the tool stops after a single block in the program is executed. Check the program in the single block mode by executing the program block by block.



5.5 Single block

Procedure for Single block

Procedure

1 Press the single block switch on the machine operator's panel. The execution of the program is stopped after the current block is executed.

2 Press the cycle start button to execute the next block. The tool stops after the block is executed.

Refer to the appropriate manual provided by the machine tool builder for single block execution.

Explanation

- **Reference position return and single block**

If G28 to G30 are issued, the single block function is effective at the intermediate point.

- **Subprogram call and single block**

Single block stop is not performed in a block containing M98P_;. M99; or G65.
However, single block stop is even performed in a block with M98P_ or M99 command, if the block contains an address other than O, N or P.

6 SAFETY FUNCTIONS

To immediately stop the machine for safety, press the Emergency stop button. To prevent the tool from exceeding the stroke ends, Stroke check is available. This chapter describes emergency stop., and stroke check.

6.1 EMERGENCY STOP

If you press Emergency Stop button on the machine operator's panel, the machine movement stops in a moment.

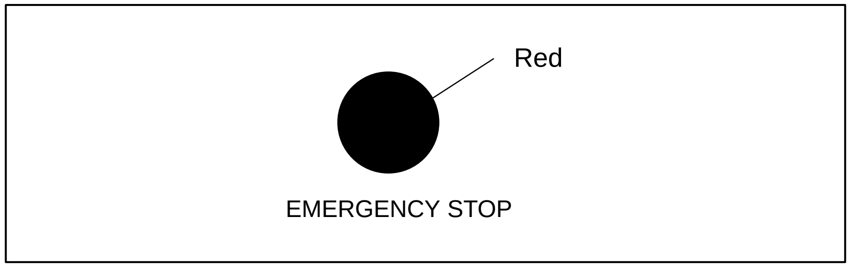


Fig. 6.1 Emergency stop

This button is locked when it is pressed. Although it varies with the machine tool builder, the button can usually be unlocked by twisting it.

Explanation

EMERGENCY STOP interrupts the current to the motor.
Causes of trouble must be removed before the button is released.

6.2 STROKE CHECK

Area which the tool cannot enter can be specified with stored stroke limit 1.

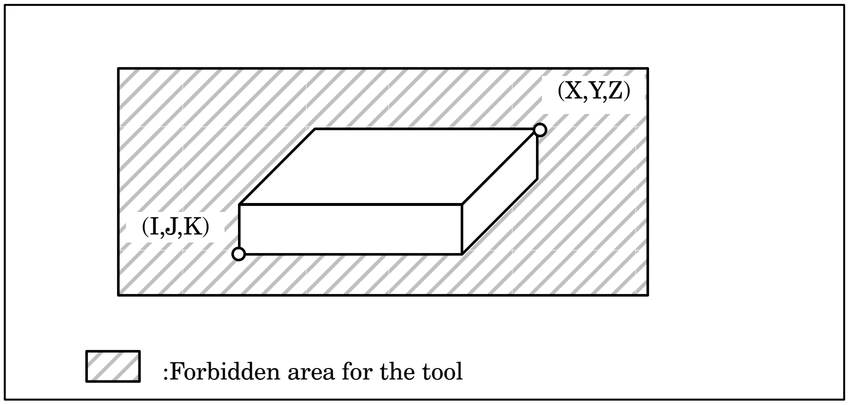


Fig. 6.2(a) Stroke check

When the tool exceeds a stored stroke limit, an alarm is displayed and the tool is decelerated and stopped.
When the tool enters a forbidden area and an alarm is generated, the tool can be moved in the reverse direction from which the tool came.

Explanation

Stored stroke limit

Parameters (Nos. 1320, 1321 or Nos. 1326, 1327) set boundary. Outside the area of the set limits is a forbidden area. The machine tool builder usually sets this area as the maximum stroke.

Unnecessary limits should be set beyond the machine stroke.

- **Effective time for a forbidden area**

Each limit becomes effective after the power is turned on and manual reference position return or automatic reference position return by G28 has been performed.
After the power is turned on, if the reference position is in the forbidden area of each limit, an alarm is generated immediately.

- **Releasing the alarms**

If the tool has already entered a forbidden area when reference position return is performed, an alarm is issued, preventing the tool from being moved out of that area. In such a case, check the set value and, if erroneous, correct it. Then, retry from reference position return.

NOTE
In setting a forbidden area, if the two points to be set are the same, the area is as follows:
· All areas are forbidden areas.

- **Timing for displaying an alarm**

Parameter BFA (bit 7 of No. 1300) selects whether an alarm is displayed immediately before the tool enters the forbidden area or immediately after the tool has entered the forbidden area.

Alarms

Number	Message	Contents
500	OVER TRAVEL: +n	The n–th axis (1–6) exceeded + side stored stroke limit.
501	OVER TRAVEL: –n	The n–th axis (1–6) exceeded – side stored stroke limit.

7 ALARM AND SELF–DIAGNOSIS FUNCTIONS

When an alarm occurs, the corresponding alarm screen appears to indicate the cause of the alarm. The causes of alarms are classified by error codes. Up to 25 previous alarms can be stored and displayed on the screen (alarm history display).
The system may sometimes seem to be at a halt, although no alarm is displayed. In this case, the system may be performing some processing. The state of the system can be checked using the self–diagnostic function.

7.1 ALARM DISPLAY

Explanations

- **Alarm screen (CRT/MDI)**

When an alarm occurs, the alarm screen appears.

```
ALARM MESSAGE                                0000   00000

     100    PARAMETER WRITE ENABLE
     510    OVER TRAVEL   :+1
     520    OVER TRAVEL   :+2
     530    OVER TRAVEL   :+3




  MDI  ****   ***   ***
[ARALM] [ MSG ] [ HISTRY ] [      ] [      ]
```

- **Another method for alarm displays (CRT/MDI)**

In some cases, the alarm screen does not appear, but an ALM is displayed at the bottom of the screen.

```
 PARAMETER (RS232C INTERFACE)          O1000  N00010

0100   ENS                  NCR         CTV
       0     0     0     0    0     0     0     0
0101   NFD                  ASI               SB2
       0     0     0     0    0     0     0     1
0102  DEVICE NUM. (CH0)
0103  BAUDRATE (CH0)          10
0111   NFD                  ASI               SB2
       0     0     0     0    0     0     0     0
0112  DEVICE NUM. (CH1)           0
0113  BAUDRATE (CH1)          0

 >_
  AUTO ****   ***   *** ALM
[NO.SRH] [ ON:1 ] [ OFF:0 ] [+INPUT ] [ INPUT ]
```

In this case, display the alarm screen as follows:

1 Press the function key MESSAGE .

2 Press the chapter selection soft key **[ALARM]**.

- **Reset of the alarm**

Error codes and messages indicate the cause of an alarm. To recover from an alarm, eliminate the cause and press the reset key.

- **Error codes**

The error codes are classified as follows:
No. 000 to 232: Program errors(*)
No. 300 to 308: Absolute pulse coder (APC) alarms
No. 350 and 351: Serial pulse coder (SPC) alarms
No. 400 to 417: Servo alarms
No. 500 to 507: Overtravel alarms
No. 700 to 704: Overheat alarms
No. 900 to 973: System alarms

*For an alarm (No. 000 to 232) that occurs in association with background operation, the indication ”xxxBP/S alarm” is provided (where xxx is an alarm number). Only a BP/S alarm is provided for No. 140.
See the error code list in the appendix G for details of the error codes.

- **Displaying the Alarm screen (DPL/MDI)**

Press the <ALARM> key to toggle back and forth between the alarm screen and the message screen.

Alarm screen		Message screen
ALARM BAL 100 410Y 407Y	←→	HEAD1 STOP

1. Up to four alarms can be displayed at once.
   Alarm numbers for the axis type are followed by an axis name.
2. The state of the battery alarm is displayed on the DPL screen the right.
3. The message screen displays external messages from the PMC.
   (For details, refer to the manual issued by the machine builder.)

7.2 ALARM HISTORY DISPLAY

Up to 25 of the most recent CNC alarms are stored and displayed on the screen.
Display the alarm history as follows:

Procedure for Alarm History Display

1 Press the function key MESSAGE .

2 Press the chapter selection soft key **[HISTRY]**.
The alarm history appears.
The following information items are displayed.
(1) Alarm No.
(2) Alarm message (some contains no message)

3 To delete the recorded information, press the softkey **[(OPRT)]** then the **[DELETE]** key.

```
ALARM HISTORY                                O0100  N00001

(1)010  (2)IMPROPER G–CODE

500   OVER TRAVEL : +1

417 SERVO ALARM : X AXIS DGTL PARAM




  AUTO ****   ***   ***
[ ALARM ][ MSG ][ HISTRY ][        ][ (OPRT) ]
```

(1) Alarm No.
(2) Alarm message (some contains no message)

7.3 CHECKING BY SELF-DIAGNOSTIC SCREEN

The system may sometimes seem to be at a halt, although no alarm has occurred. In this case, the system may be performing some processing. The state of the system can be checked by displaying the self–diagnostic screen.

Procedure for Diagnois

1 Press the function key SYSTEM .

2 Press the chapter select key **[DGNOS]**.

3 The diagnostic screen has more than 1 pages. Select the screen by the following operation.
(1) Change the page by the 1–page change key.
(2) Method by soft key
– Key input the number of the diagnostic data to be displayed.
– Press **[N SRCH]**.

```
DIAGNOSTIC (GENERAL)                          O0000  N0000


000  WAITING FOR FIN SIGNAL                              :0
001  MOTION                                              :0
002  DWELL                                               :0
003  IN–POSITION  CHECK                                  :0
004  FEEDRATE  OVERRIDE  0%                              :0
005  INTERLOCK/START–LOCK                                :0
006                                                      :0

 >_
 EDIT  ****   ***   ***
[ PARAM ][ DGNOS ][ PMC ][ SYSTEM ][ (OPRT) ]
```

Explanations

Diagnostic numbers 000 to 015 indicate states when a command is being specified but appears as if it were not being executed. The table below lists the internal states when 1 is displayed at the right end of each line on the screen.

Table 7.3 (a) Alarm displays when a command is specified but appears as if it were not being executed

No.	Display	Internal status when 1 is displayed
000	WAITING FOR FIN SIGNAL	M function being executed
001	MOTION	Move command in automatic operation being executed
002	DWELL	Dwell being executed
003	IN–POSITION CHECK	In–position check being executed
004	FEEDRATE OVERRIDE 0%	Cutting feed override 0%
005	INTERLOCK/START–LOCK	Interlock ON
010	PUNCHING	Data being output via reader puncher interface
011	READING	Data being input via reader puncher interface
013	JOG FEEDRATE OVERRIDE 0%	Jog override 0%
014	WAITING FOR RESET.ESP.RRW.OFF	Emergency stop, external reset, reset & rewind, or MDI panel reset key on
015	EXTERNAL PROGRAM NUMBER SEARCH	External program number searching

Table 7.3 (b) Alarm displays when an automatic operation is stopped or paused.

No.	Display	Internal status when 1 is displayed
020	CUT SPEED UP/DOWN	Set when emergency stop turns on or when servo alarm occurs
021	RESET BUTTON ON	Set when reset key turns on
022	RESET AND REWIND ON	Reset and rewind turned on
023	EMERGENCY STOP ON	Set when emergency stop turns on
024	RESET ON	Set when external reset, emergency stop, reset, or reset & rewind key turns on
025	STOP MOTION OR DWELL	A flag which stops pulse distribution. It is set in the following cases. (1)External reset turned on. (2)Reset & rewind turned on. (3)Emergency stop turned on. (4)Feed hold turned on. (5)The MDI panel reset key turned on. (6)Switched to the manual mode(JOG/INC). (7)Other alarm occurred. (There is also alarm which is not set.)

The table below shows the signals and states which are enabled when each diagnostic data item is 1. Each combination of the values of the diagnostic data indicates a unique state.

020	CUT SPEED/UP/DOWN	1	0	0	0	1	0	0
021	RESET BUTTON ON	0	0	1	0	0	0	0
022	RESET AND REWIND ON	0	0	0	1	0	0	0
023	EMERGENCY STOP ON	1	0	0	0	0	0	0
024	RESET ON	1	1	1	1	0	0	0
025	STOP MOTION OR DWELL	1	1	1	1	1	1	0

Emergency stop signal input
External reset signal input
MDI reset button turned on
Reset & rewind input
Servo alarm generation
Changed to another mode or feed hold
Single block stop

Diagnostic numbers 030 and 031 indicate TH alarm states.

No.	Display	Meaning of data
030	CHARACTER NUMBER TH DATA	The position of the character which caused TH alarm is displayed by the number of characters from the beginning of the block at TH alarm
031	TH DATA	Read code of character which caused TH alarm

7.4 DISPLAYING AND SETTING PMC DATA IN DIAGNOSIS SCREEN (DPL/MDI)

Displaying PMC data

Procedure

1 Press the <DGNOS/PARAM> key to select the diagnosis screen.

```
> @0001    0
  @0002    1
```

2 Press the key of the PMC address to be displayed.
(Use the bottom left address of the key.)

```
> @0001    0
  D_       0
```

Example: Display the address data for D0100

3 Enter the number of the PMC address to be displayed.

```
> @0001    0
  D0100
```

4 Press the <INPUT> key.

```
> D0100    000000000
  D0100    000001010
```

By pressing the ↓ and ↑ keys, the cursor can be moved within the PMC address being displayed.

Changing the data format

Procedure

1 Pressing the < · > key when PMC data is displayed changes the data format for display/setting.
Each time the < · > key is pressed, the data format changes in the order:

→ 1 byte of flag bits → 1-byte decimal → 2-byte decimal → 4-byte decimal →

> **NOTE**
> The size for the data format currently selected corresponds to the difference between a displayed number and a number displayed below it.

Ending PMC data display

Procedure

Pressing <NO.>, <Number>, and <INPUT> redisplays the diagnosis screen.

Setting PMC data

PMC data can be set from the DPL/MDI when setting parameter DWE is set to 1.

Procedure

1 Select the setting parameter display.

2 Use the cursor keys to position the cursor on DWE.
Press <1> and <INPUT> to set DWE to 1.

> **WARNING**
> When not setting PMC data, set DWE to 0 so that PMC data is not set inadvertently.

3 Select a PMC address and enter a number.

```
> D0100     00000000
  D0101     00001010
```

Example: Enter 100 in decimal in the address data of D0100.

4 Press the < · > key to select a data format.

```
> D0100     0
  D0101     10
```

Example: Select 1-byte decimal.

5 Use the numeric keys to enter a value.

```
> D0100     0
  D0100=    100_
```

6 Press the <INPUT> key.
The data value is input and displayed.

```
> D0100     100
  D0101     10
```

> **NOTE**
> The range of values that can be entered in each data format is as follows:

1 byte of flag bits	:	8 independent bits, each only taking either 0 or 1.
1-byte decimal	:	–128 to 127
2-bytes decimal	:	–32768 to 32767
4-bytes decimal	:	–99999999 to 99999999

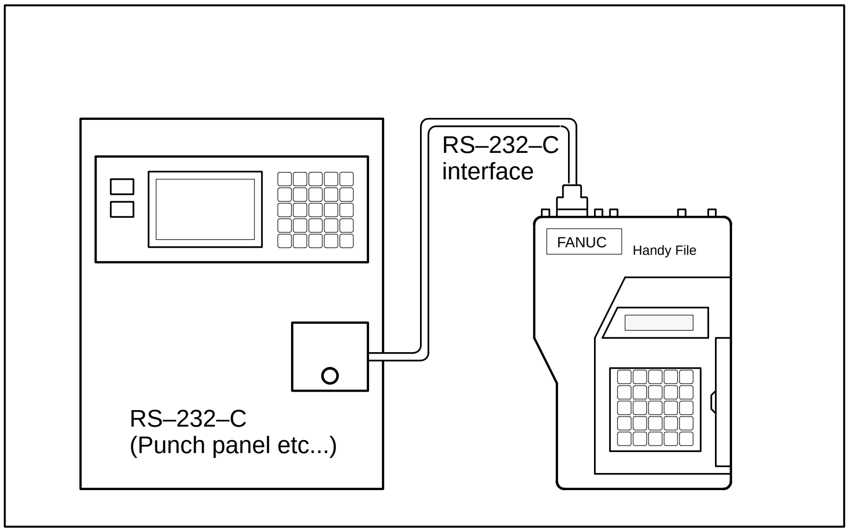
PMC data display/setting areas

The following lists the PMC locations where data can be displayed or set.

X0000 to 0127, X1000 to 1063
Y0000 to 0127, Y1000 to 1063
G0000 to 0255
F0000 to 0255
A0000 to 0024
R0000 to 0999, R9000 to 9117
T0000 to 0079
K0000 to 0019
C0000 to 0079
D0000 to 1859

8 DATA INPUT/OUTPUT

controller data is transferred between the controller and external input/output devices such as the Handy File.
The following types of data can be entered and output :
1.Program
2.Offset data
3.Parameter
4.Custom macro common variable
Before an input/output device can be used, the input/output related parameters must be set.
For how to set parameters, see Chapter 2 **OPERATIONAL DEVICES**.



8.1 FILES

Of the external input/output devices, the FANUC Handy File and FANUC Floppy Cassette use floppy disks as their input/output medium.
In this manual, an input/output medium is generally referred to as a floppy. However, when the description of one input/output medium varies from the description of another, the name of the input/output medium is used. In the text below, a floppy represents a floppy disk.
Unlike an controller tape, a floppy allows the user to freely choose from several types of data stored on one medium on a file–by–file basis.
Input/output is possible with data extending over more than one floppy disk.

Explanations

- **What is a File**

The unit of data, which is input/output between the floppy and the controller by one input/output operation (pressing theVREADWorVPUNCHWkey), is called a HfileI. When inputting controller programs from, or outputting them to the floppy, for example, one or all programs within the controller memory are handled as one file. Files are assigned automatically file numbers 1,2,3,4 and so on, with the lead file as 1.

File 1	File 2	File 3		File n	Blank

- **Request for floppy replacement**

When one file has been entered over two floppies, LEDs on the adaptor flash alternately on completion of data input/output between the first floppy and the controller, prompting floppy replacement. In this case, take the first floppy out of the adaptor and insert a second floppy in its place. Then, data input/output will continue automatically.
Floppy replacement is prompted when the second floppy and later is required during file search–out, data input/output between the controller and the floppy, or file deletion.

Floppy 1

File 1	File 2	File 3		File (k–1)	File k

Floppy 2

Continuation of file k	File (k+1)		File n	Blank

Since floppy replacement is processed by the input/output device, no special operation is required. The controller will interrupt data input/output operation until the next floppy is inserted into the adaptor.
When reset operation is applied to the controller during a request for floppy replacement, the controller is not reset at once, but reset after the floppy has been replaced.

- **Protect switch**

The floppy is provided with the write protect switch. Set the switch to the write enable state. Then, start output operation.

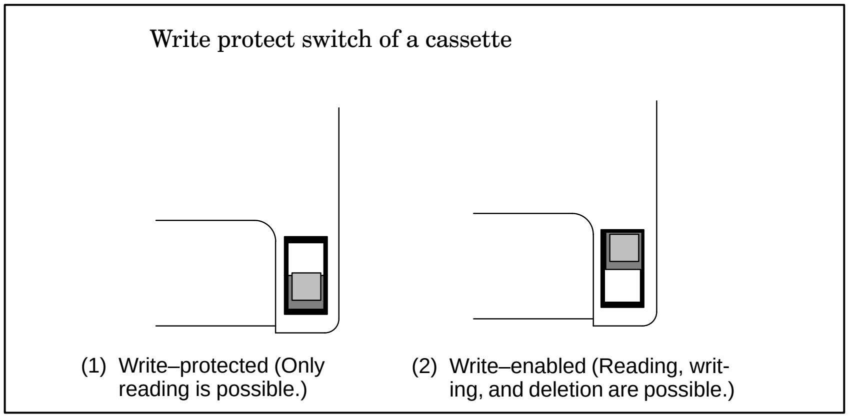


Fig 8.1. Protect swtich

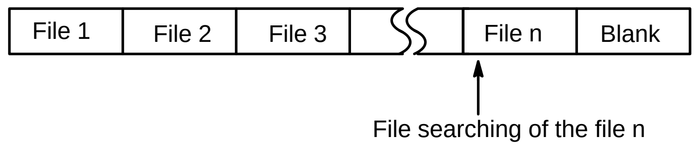
- **Writing memo**

Once written in the cassette data can subsequently be read out by correspondence between the data contents and file numbers. This correspondence cannot be verified, unless the data contents and file numbers are output to the controller and displayed. The data contents can be displayed with display function for directory of floppy disk
(See Section 8.8).
To display the contents,write the file numbers and the contents on the memo column which is the back of floppy.

(Entry example on MEMO)
File 1 controller parameters
File 2 Offset data
File 3 controller program O0100
. .
. .
. .
File (n–1) controller program O0500
File n controller program O0600

8.2 FILE SEARCH

When the program is input from the floppy, the file to be input first must be searched.
For this purpose, proceed as follows:



File heading

Procedure (CRT/MDI)

1 Press the EDIT or AUTO switch on the machine operator's panel.

2 Press function key PROG .

3 Press soft key **[(OPRT)]**

4 Press the rightmost soft key ▷ (next–menu key).

5 Enter address N.

6 Enter the number of the file to search for.
·N0
The beginning of the cassette is searched.
·One of N1 to N9999
Of the file Nos. 1 to 9999, a designated file is searched.
·N–9999
The file next to that accessed just before is searched.
·N–9998
When N–9998 is designated, N–9999 is automatically inserted each time a file is input or output. This condition is reset by the designation of N0,N1 to 9999, or N – 9999 or reset.

7 Press soft keys **[FSRH]** and **[EXEC]**
The specified file is searched for.

Procedure (DPL/MDI)

1 Select EDIT or AUTO mode.

2 Push <PRGRM> key to select the program screen.

3 Key in address N.

4 Key in a file number.

5 Press <READ> key.
The following head searching occurs according to the number specified:

(a) N0
The beginning of the cassette is searched.

(b) One of N1 to N9999
Of the file Nos. 1 to 9999, a designated file is searched.

(c) N–9999
The file next to that accessed just before is searched.

(d) N–9998
When N–9998 is designated, N–9999 in (c) is automatically inserted each time a file is input or output. This condition is reset by the designation of (a), (b) or (c) or reset.

Explanation

- **File search by N–9999**

The same result is obtained both by sequentially searching the files by specifying Nos. N1 to N9999 and by first searching one of N1 to N9999 and then using the N–9999 searching method. The searching time is shorter in the latter case.

Alarm

No.	Description
86	The ready signal (DR) of an input/output device is off. An alarm is not immediately indicated in the controller even when an alarm occurs during head searching (when a file is not found, or the like). An alarm is given when the input/output operation is performed after that. This alarm is also raised when N1 is specified for writing data to an empty floppy. (In this case, specify N0.)

8.3 FILE DELETION

Files stored on a floppy can be deleted file by file as required.

File deletion

Procedure (CRT/MDI)

1 Insert the floppy into the input/output device so that it is ready for writing.

2 Press the EDIT switch on the machine operator's panel.

3 Press function key PROG

4 Press soft key **[(OPRT)]**

5 Press the rightmost soft key ▷ (next–menu key).

6 Enter address N.

7 Enter the number (from 1 to 9999) of the file to delete.

8 Press soft key **[DELETE]**
The file specified in step **7** is deleted.

Procedure (DPL/MDI)

1 Select EDIT mode.

2 Push <PRGRM>.

3 Turn off the protect key.

4 Key in address N.

5 Key in file No. 1 to 9999 to be deleted.

6 Push the <WRTIE>key.

With this operation, the k-th file input in 5) is deleted.

Explanations

- **File number after the file is deleted**

When a file is deleted, the file numbers after the deleted file are each decremented by one. Suppose that a file numbered k was deleted. In this case, files are renumbered as follows:

Before deletion after deletion
1 to (k–1) . . . . . 1 to (k–1)
k . . . . Deleted
(k+1) to n . . . . . . k to (n–1)

- **Protect switch**

Set the write protect switch to the write enable state to delete the files.

8.4 PROGRAM INPUT/OUTPUT

8.4.1 Inputting a Program

This section describes how to load a program into the controller from a floppy or NC tape.

Inputting a program

Procedure (CRT/MDI)

1 Make sure the input device is ready for reading.

2 Press the EDIT switch on the machine operator's panel.

3 When using a floppy, search for the required file according to the procedure in Section **8.2**.

4 Press function key PROG

5 Press soft key **[(OPRT)]**

6 Press the rightmost soft key ▷ (next–menu key).

7 After entering address O, specify a program number to be assigned to the program. When no program number is specified here, the program number used on the floppy or controller tape is assigned.

8 Press soft keys **[READ]** and **[EXEC]**
The program is input and the program number specified in step **7** is assigned to the program.

Procedure (DPL/MDI)

1 Select EDIT mode.

2 Set the NC tape on the tape reader.

3 Press <PROG> to display the program screen.

4 When the controller tape does not have a program number or a program number is to be changed, enter a desired program number. (When the controller tape has a program number and a program number is not changed, this operation is not necessary.)
i) Key in address O.
ii) Key in a desired program number.

5 Press the <READ> key.

Explanations

- **Collation**

If a program is input while the data protect key on the machine operator's panel turns ON, the program loaded into the memory is verified against the contents of the floppy or controller tape.
If a mismatch is found during collation, the collation is terminated with an alarm (P/S No. 79).
If the operation above is performed with the data protection key turns OFF, collation is not performed, but programs are registered in memory.

- **Inputting multiple programs from a NC tape**

When a tape holds multiple programs, the tape is read up to ER (or %).

O1111 – – – M02;	O2222 – – – M30;	O3333 – – – M02;	ER(%)

- **Program numbers on a NC tape**

¢When a program is entered without specifying a program number.
·The O–number of the program on the NC tape is assigned to the program. If the program has no O–number, the N–number in the first block is assigned to the program.
·When the program has neither an O–number nor N–number, the previous program number is incremented by one and the result is assigned to the program.
·When the program does not have an O–number but has a five–digit sequence number at the start of the program, the lower four digits of the sequence number are used as the program number. If the lower four digits are zeros, the previously registered program number is incremented by one and the result is assigned to the program.
¢When a program is entered with a program number
The O–number on the NC tape is ignored and the specified number is assigned to the program. When the program is followed by additional programs, the first additional program is given the program number. Additional program numbers are calculated by adding one to the last program.

- **Program registration in the background**

The method of registration operation is the same as the method of foreground operation. However, this operation registers a program in the background editing area. As with edit operation, the operations described below are required at the end to register a program in foreground program memory.
·For CRT/MDI
[(OPRT)][BG–END]
·For DPL/MDI
<CAN>+<PRGRM>

- **Additional program input**

You can input a program to be appended to the end of a registered program.

Registered program	Input program	Program after input
O1234 ;	O5678 ;	O1234 ;
□□□□□□□ ;	○○○○○○○ ;	□□□□□□□ ;
□□□□□;	○○○○○ ;	□□□□□;
□□□□;	○○○○ ;	□□□□;
□□□;	○○○ ;	□□□;
%	%	%
		O5678 ;
		○○○○○○○ ;
		○○○○○ ;
		○○○○ ;
		○○○ ;
		%

In the above example, all lines of program O5678 are appended to the end of program O1234. In this case, program number O5678 is not registered. When inputting a program to be appended to a registered program, press the **[READ]** soft key without specifying a program number in step 8 (CRT/MDI). Then, press the **[CHAIN]** and **[EXEC]** soft keys.

· In entire program input, all lines of a program are appended, except for its O number.

- When canceling additional input mode, press the reset key or the [CAN] or [STOP] soft key.
- Pressing the [CHAIN] soft key positions the cursor to the end of the registered program. Once a program has been input, the cursor is positioned to the start of the new program.
- Additional input is possible only when a program has already been registered.

Alarm

No.	Description
70	The size of memory is not sufficient to store the input programs
73	An attempt was made to store a program with an existing program number.
79	The verification operation found a mismatch between a program loaded into memory and the contents of the program on the floppy or NC tape.

8.4.2 Outputting a Program

A program stored in the memory of the controller unit is output to a floppy or controller tape.

Outputting a program

Procedure (CRT/MDI)

1 Make sure the output device is ready for output.

2 To output to an controller tape, specify the punch code system (ISO or EIA) using a parameter.

3 Press the EDIT switch on the machine operator's panel.

4 Press function key PROG .

5 Press soft key **[(OPRT)]**.

6 Press the rightmost soft key ▷ (next–menu key).

7 Enter address O.

8 Enter a program number. If –9999 is entered, all programs stored in memory are output.
To output multiple programs at one time, enter a range as follows :
OΔΔΔΔ,O□□□□
Programs No.ΔΔΔΔ to No.□□□□ are output.

9 Press soft keys **[PUNCH]** and **[EXEC]**
The specified program or programs are output.

Procedure (DPL/MDI)

A program registered in memory can be punched using the procedure below.

1 Set the output device ready for punch operation.

2 Set a setting data punch code (ISO or EIA).

3 Select EDIT mode.

4 Press **<PRGRM>** to display the program screen.

5 Key in address O.

6 Key in a desired program number.
Entering –9999 causes all programs in memory to be output.

7 The number of input program is punched with pushing **<WRITE>**.

Explanations (Output to a floppy)

- **File output location**

When output is conducted to the floppy, the program is output as the new file after the files existing in the floppy. New files are to be written from the beginning with making the old files invalid, use the above output operation after the N0 head searching.

- **An alarm while a program is output**

When P/S alarm 86 occurs during program output, the floppy is restored to the condition before the output.

- **Outputting a program after file heading**

When program output is conducted after N1 to N9999 head searching, the new file is output as the designated n–th position. In this case, 1 to n–1 files are effective, but the files after the old n–th one are deleted. If an alarm occurs during output, only the 1 to n–1 files are restored.

- **Efficient use of memory**

To efficiently use the memory in the cassette or card, output the program by setting parameter NFD (No. 0101#7,No. 0111#7) to 1. This parameter makes the feed is not output, utilizing the memory efficiently.

- **On the memo record**

Head searching with a file No. is necessary when a file output from the controller to the floppy is again input to the controller memory or compared with the content of the controller memory. Therefore, immediately after a file is output from the controller to the floppy, record the file No. on the memo.

- **Punching programs in the background**

Punch operation can be performed in the same way as in the foreground. This function alone can punch out a program selected for foreground operation.

- **Procedure (CRT/MDI)**

<O> (Program No.) [**PUNCH**] [**EXEC**]: Punches out a specified program.
<O> H–9999I [**PUNCH**] [**EXEC**]: Punches out all programs.

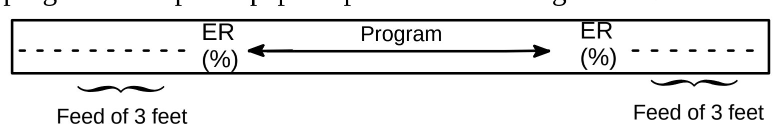
- **Procedure (DPL/MDI)**

<O> (Program No.) **<WRITE>**: Punches out a specified program.
<O> H–9999I **<WRITE>**: Punches out all programs.

Explanations (Output to an controller tape)

- **Format**

A program is output to paper tape in the following format:



If three–feet feeding is too long, press the CAN key during feed punching to cancel the subsequent feed punching.

- **TV check**

A space code for TV check is automatically punched.

- **ISO code**

When a program is punched in ISO code, two CR codes are punched after an LF code.

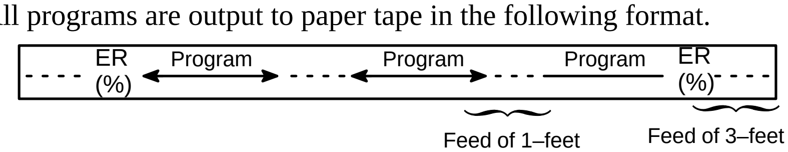
- - - - - - - - - - - - - - LF CR CR

- **Stopping the punch**

Press the RESET key to stop punch operation.

- **Punching all programs**

All programs are output to paper tape in the following format.



The sequence of the programs punched is undefined.

8.5 OFFSET DATA INPUT AND OUTPUT

8.5.1 Inputting Offset Data

Offset data is loaded into the memory of the controller from a floppy or NC tape. The input format is the same as for offset value output. See section 8.5.2.
When an offset value is loaded which has the same offset number as an offset number already registered in the memory, the loaded offset data replaces existing data.

Inputting offset data

Procedure (CRT/MDI)

1 Make sure the input device is ready for reading

2 Press the EDIT switch on the machine operator's panel.

3 When using a floppy, search for the required file according to the procedure in Section 8.2.

4 Press function key OFFSET SETTING .

5 Press soft keys **[(OPRT)]**.

6 Press rightmost soft key ▷ (next menu key).

7 Press soft keys **[READ]** and **[EXEC]**.

8 The input offset data will be displayed on the screen after completion of input operation.

Procedure (DPL/MDI)

1 Select the EDIT mode.

2 Display the data display screen by pressing **<VAR>** key.

3 Perform the same operation as for program input.

4 The input offset data will be displayed on the screen after completion of input operation.

8.5.2 Outputting Offset Data

All offset data is output in a output format from the memory of the controller to a floppy or NC tape.

Outputting offset data

Procedure (CRT/MDI)

1 Make sure the output device is ready for output.

2 Specify the punch code system (ISO or EIA) using a parameter.

3 Press the EDIT switch on the machine operator's panel.

4 Press function key OFFSET SETTING .

5 Press soft key**[(OPRT)]**.

6 Press the rightmost soft key ▷ (next–menu key)

7 Press soft keys **[PUNCH]**and**[EXEC]**.
Offset data is output in the output format described below.

Procedure (DPL/MDI)

1 Select the EDIT mode.

2 Select the offset data display screen by pressing **<VAR>** key.

3 Press the **<WRITE>** key.

4 Specify file heading when required.
For which file the offset date is output to, refer to 8.4.2.

5 While offset, is being output, the display appears as below.

```
>#0100=
        WRITE
```

6 In order to stop output of data from a tape before it has finished, turn on external reset signal ERS (bit 7 of G008).
Once data output from a tape has been stopped, it cannot be restarted.

Explanations

● Output format

Output format is as follows:

Format

G10 L11 P_R_;
where P_: Offset No.
R_: Tool compensation amount

● Output file name

When the floppy disk directory display function is used, the name of the output file is OFFSET.

8.6 INPUTTING AND OUTPUTTING PARAMETERS

8.6.1 Inputting Parameters

Parameters are loaded into the memory of the controller unit from a floppy or NC tape. The input format is the same as the output format. See Section **8.6.2**. When a parameter is loaded which has the same data number as a parameter already registered in the memory, the loaded parameter replaces the existing parameter.

Inputting parameters

Procedure (CRT/MDI)

1. Make sure the input device is ready for reading.
2. When using a floppy, search for the required file according to the procedure in Section **8.2**.
3. Press the EMERGENCY STOP button on the machine operator's panel.
4. Press function key OFFSET SETTING .
5. Press the soft key **[SETING]** for chapter selection.
6. Enter 1 in response to the prompt for writing parameters (PWE). Alarm P/S100 (indicating that parameters can be written) appears.
7. Press soft key SYSTEM .
8. Press chapter selection soft key **[PARAM]**.
9. Press soft key **[(OPRT)]**.
10. Press the rightmost soft key ▷ (next–menu key).
11. Press soft keys **[READ]** and **[EXEC]**.
    Parameters are read into memory. Upon completion of input, the "INPUT" indicator at the lower–right corner of the screen disappears.
12. Press function key OFFSET SETTING .
13. Press soft key **[SETING]** for chapter selection.
14. Enter 0 in response to the prompt for writing parameters.
15. Turn the power to the NC back on.
16. Release the EMERGENCY STOP button on the machine operator's panel.

Procedure (DPL/MDI)

1 Press the EMERGENCY STOP button on the machine side.

2 The parameter screen is selected by pressing the <PARAM> key.

3 Set PWE on the setting screen to 1. Alarm PS100 is displayed at this time.

4 Perform the same operation as for program input.

5 NC parameters are input to the memory by this operation. Normally, alarm PS000 will activate after completion of parameter reading. Normally, P/S alarm 000 is generated after parameters have finished being read in.

6 Set PWE on the setting parameter to 0.

7 Turn on the NC power again if PS alarm activates.

8 Release the emergency stop button of machine side.

8.6.2 Outputting Parameters

All parameters are output in the defined format from the memory of the controller to a floppy or NC tape.

Outputting parameters

Procedure (CRT/MDI)

1 Make sure the output device is ready for output.

2 Specify the punch code system (ISO or EIA) using a parameter.

3 Press the EDIT switch on the machine operator’s panel.

4 Press function key SYSTEM .

5 Press chapter selection soft key **[PARAM]**.

6 Press soft key **[(OPRT)]**.

7 Press rightmost soft key (next–menu key).

8 Press soft keys **[PUNCH]** and **[EXEC]**.
All parameters are output in the defined format.

Procedure (DPL/MDI)

1 Select the EDIT mode.

2 Select the parameter display screen by <PARAM> key.

3 Press the <WRITE> key.

4 Execute file heading when required.
For which file the parameter is output to, refer to 8.4.2.

5 While parameter, is being output, the display appears as below.

```
>&0100      00000000
            WRITE
```

6 In order to stop output of data from a tape before it has finished, turn on external reset signal ERS (bit 7 of G008).
Once data output from a tape has been stopped, it cannot be restarted.

Explanations

- **Output format**

Output format is as follows:
NP ;
NA1P A2P AnP ;
NP ;

N:Parameter No.
A:Axis No.(n is the number of control axis)
P:Parameter setting value .

- **Output file name**

When the floppy disk directory display function is used, the name of the output file is PARAMETER.

8.7 INPUTTING/OUTPUTTING CUSTOM MACRO COMMON VARIABLES

8.7.1 Inputting Custom Macro Common Variables

The value of a custom macro common variable (#500 to #699) is loaded into the memory of the controller from a floppy or NC tape. The same format used to output custom macro common variables is used for input. See Section **8.7.2.** For a custom macro common variable to be valid, the input data must be executed by pressing the cycle start button after data is input. When the value of a common variable is loaded into memory, this value replaces the value of the same common variable already existing (if any) in memory.

Inputting custom macro common variables

Procedure (CRT/MDI)

1 Input the program according to the procedure in Section **8.4.1**.

2 Select the AUTO mode by the machine operator's panel upon completing input.

3 Press the cycle start button to execute the loaded program.

4 Display the macro vriable screen to check whether the values of the common variables have been set correctly.

<u>**Display of the macro variable screen**</u>

·Press function key OFFSET SETTING .

· Press the rightmost soft key (next–menu key).

·Press soft key [**MACRO**].

·Select a variable with the page keys or numeric keys and soft key **[NO.SRH]**.

Procedure (DPL/MDI)

1 Select EDIT mode.

2 Perform the same operation as for program input and read in the custom macro statements like a program.

3 After reading is finished, select AUTO mode. By executing the program that was read in, the values of the common variables will be stored in memory.

Explanations

- **Common variables**

The common variables (#500 to #699) can be input and output.
Common variables #100 to 199 cannot be input or output.

8.7.2 Outputting Custom Macro Common Variable

Custom macro common variables (#500 to #699) stored in the memory of the controller can be output in the defined format to a floppy or NC tape.

Outputting custom macro common variable

Procedure (CRT/MDI)

1 Make sure the output device is ready for output.

2 Specify the punch code system (ISO or EIA) using a parameter.

3 Press the EDIT switch on the machine operator's panel.

4 Press function key OFFSET SETTING .

5 Press the rightmost soft key ▷ (next–menu key), then press soft key **[MACRO]**.

6 Press soft key **[(OPRT)]**.

7 Press the rightmost soft key ▷ (next–menu key).

8 Press soft keys **[PUNCH]** and **[EXEC]**.
Common variables are output in the defined format.

Procedure (DPL/MDI)

1 Select the EDIT mode.

2 Select the tool offset data display screen by pressing **<VAR>** key.

3 Press the **<WRITE>** key.

4 Specify file heading when required.
For which file the offset date is output to, refer to 8.4.2.

5 While common variable is being output, the display appears as below.

```
>#0100 =
               WRITE
```

6 In order to stop output of data from a tape before it has finished, turn on external reset signal ERS (bit 7 of G008).
Once data output from a tape has been stopped, it cannot be restarted.

Explanations

- **Output format**

The output format is as follows:

```
%
;
#500=[25283*65536+65536]/134217728 ......... (1)
#501=#0; .................................... (2)
#502=0; ..................................... (3)
#503= ....................................... ;
............................................. ;
............................................. ;
#699= ....................................... ;
M02;
%
```

(1) The precision of a variable is maintained by outputting the value of the variable as <expression>.

(2) Undefined variable

(3) When the value of a variable is 0

- **Output file name**

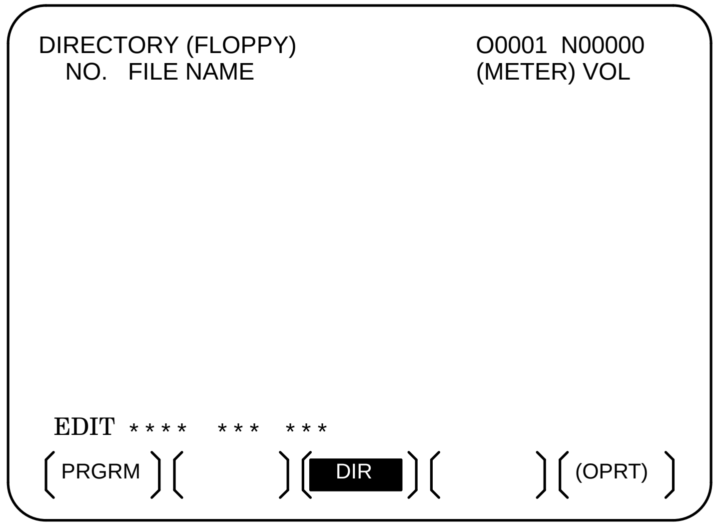
When the floppy disk directory display function is used, the name of the output file is V**MACRO VAR**W.

- **Common variable**

The common variables (#500 to #699) can be input and output. Common variables #100 to 199 cannot be input or output.

8.8 DISPLAYING DIRECTORY OF FLOPPY DISK

On the floppy directory display screen, a directory of the FANUC Handy File or FANUC Floppy Cassette, can be displayed. In addition, those files can be loaded, output, and deleted.



8.8.1 Displaying the Directory

Displaying the directory of floppy disk files

Procedure 1 (CRT/MDI)

Use the following procedure to display a directory of all the files stored in a floppy:

1 Press the EDIT switch on the machine operator's panel.

2 Press function key PROG .

3 Press the rightmost soft key ▷ (next–menu key).

4 Press soft key **[FLOPPY]**.

5 Press page key ↑PAGE or PAGE↓.

6 The screen below appears.

```
DIRECTORY (FLOPPY)                      O0001  N00000
 NO. FILE NAME                          (METER) VOL
 0001  PARAMETER                            58.5
 0002  O0001                                 1.9
 0003  O0002                                 1.9
 0004  O0010                                 1.3
 0005  O0040                                 1.3
 0006  O0050                                 1.9
 0007  O0100                                 1.9
 0008  O1000                                 1.9
 0009  O9500                                 1.6


 EDIT ****   ***   ***
( F SRH )( READ )( PUNCH )( DELETE )(        )
```

Fig.8.8.1 (a)

7 Press a page key again to display another page of the directory.

Procedure 2 (CRT/MDI)

Use the following procedure to display a directory of files starting with a specified file number :

1 Press the EDIT switch on the machine operator’s panel.

2 Press function key PROG .

3 Press the rightmost soft key ▷ (next–menu key).

4 Press soft key **[FLOPPY]**.

5 Press soft key **[(OPRT)].**

6 Press soft key **[F SRH].**

7 Enter a file number.

8 Press soft keys**[F SET]**and**[EXEC]**.

9 Press a page key to display another page of the directory.

10 Press soft key **[CAN]** to return to the soft key display shown in the screen of Fig 8.8.1(a).

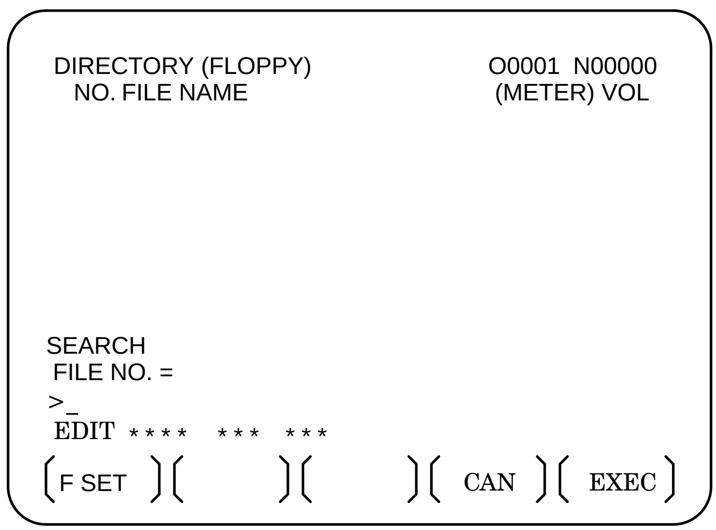


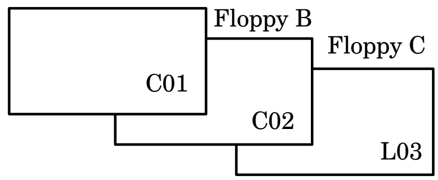
Fig.8.8.1 (b)

Explanations

- **Screen fields and their meanings**

NO :Displays the file number
FILE NAME :Displays the file name.
(METER) :Converts and prints out the file capacity to paper tape length.You can also produce H (FEET)I by setting the INPUT UNIT to INCH of the setting data.
VOL.: When the file is multi–volume, that state is displayed.

(Ex.) Floppy A



C(number)means . . . . . . CONTINUE
L(number)means . . . . . . LAST
number . . . . number of floppies

8.8.2 Reading Files

The contents of the specified file number are read to the memory of controller.

Reading files

Procedure (CRT/MDI)

1 Press the EDIT switch on the machine operator's panel.

2 Press function key PROG .

3 Press the rightmost soft key ▷ (next–menu key).

4 Press soft key **[FLOPPY]**.

5 Press soft key **[(OPRT)]**.

6 Press soft key **[READ]**.

```
DIRECTORY (FLOPPY)                          O0001  N00000
  NO. FILE NAME                             (METER) VOL





READ
 FILE NO. =                          PROGRAM NO. =
>_
EDIT  * * * *   * * *   * * *
( F SET  )( O SET )( STOP  )(  CAN  )(  EXEC  )
```

7 Enter a file number.

8 Press soft key **[F SET]**.

9 To modify the program number, enter the program number, then press soft key **[O SET]**.

10 Press soft key **[EXEC]**. The file number indicated in the lower–left corner of the screen is automatically incremented by one.

11 Press soft key **[CAN]** to return to the soft key display shown in the screen of Fig. **8.8.1**.**(a)**.

8.8.3 Outputting Programs

Any program in the memory of the controller unit can be output to a floppy as a file.

Outputting programs

Procedure(CRT/MDI)

1 Press the EDIT switch on the machine operator's panel.

2 Press function key PROG .

3 Press the rightmost soft key ▷ (next–menu key).

4 Press soft key **[FLOPPY].**

5 Press soft key **[(OPRT)].**

6 Press soft key **[PUNCH].**

```
DIRECTORY (FLOPPY)                        O0002  N01000
  NO. FILE NAME                           (METER) VOL




PUNCH
 FILE NO. =                            PROGRAM NO. =
>_
EDIT **** *** ***
[ F SET ][ O SET ][ STOP ][ CAN ][ EXEC ]
```

7 Enter a program number. To write all programs into a single file, enter –9999 in the program number field. In this case, the file name **VALL.PROGRAMW** is registered.

8 Press soft key **[O SET]**.

9 Press soft key **[EXEC]**. The program or programs specified in step **7** are written after the last file on the floppy. To output the program after deleting files starting with an existing file number, key in the file number, then press soft key **[F SET]** followed by soft key **[EXEC]**.

10 Press soft key **[CAN]** to return to the soft key display shown in the screen ofFig.**8.8.1(a)**.

8.8.4 Deleting Files

The file with the specified file number is deleted.

Deleting files

Procedure (CRT/MDI)

1 Press the EDIT switch on the machine operator's panel.

2 Press function key PROG .

3 Press the rightmost soft key ▷ (next–menu key).

4 Press soft key **[FLOPPY].**

5 Press soft key **[(OPRT)].**

6 Press soft key **[DELETE].**

```
DIRECTORY (FLOPPY)                         O0001  N00000
  NO. FILE NAME                             (METER) VOL



DELETE
 FILE NO. =        NAME=
>_
EDIT * * * *    * * *    * * *
( F SET )( F NAME )(        )( CAN )( EXEC )
```

7 Specify the file to be deleted.
When specifying the file with a file number, type the number and press soft key **NF SET].**When specifying the file with a file name, type the name and press soft key **NF NAMEO**.

8 Press soft key **[EXEC]**.
The file specified in the file number field is deleted. When a file is deleted, the file numbers after the deleted file are each decremented by one.

9 Press soft key **[CAN]** to return to the soft key display shown in the screen of Fig. **8.8.1(a)**.

Restrictions

- **Inputting file numbers and program numbers with keys**

If **[F SET]** or **[O SET]** is pressed without key inputting file number and program number, file number or program number shows blank. When 0 is entered for file numbers or program numbers, 1 is displayed.

- **I/O devices**

To use channel 0 ,set a device number in parameter 102.
Set the I/O device number to parameter No. 0112 when cannel 1 is used.

- **Significant digits**

For the numeral input in the data input area with FILE NO. and PROGRAM NO., only lower 4 digits become valid.

- **Collation**

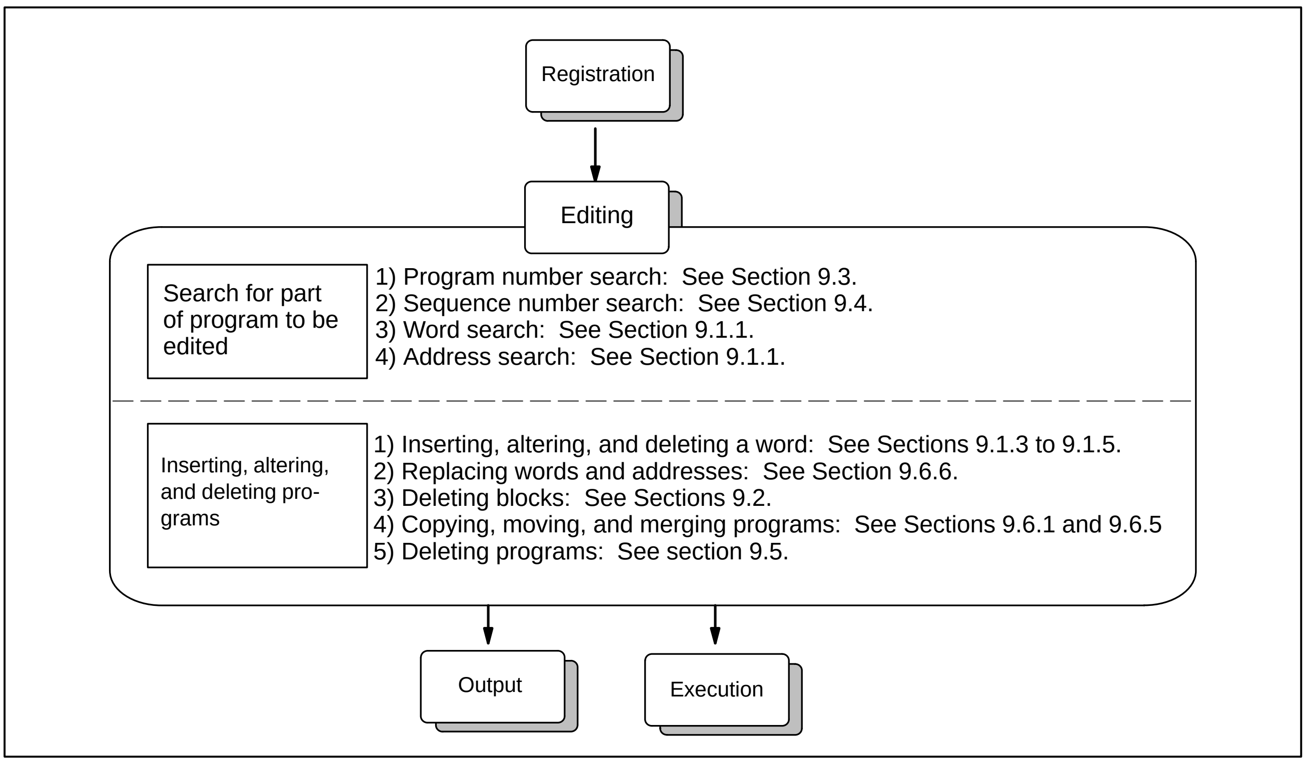
When the data protection key on the machine operator's panel is ON, no programs are read from the floppy. They are verified against the contents of the memory of the controller instead.

Alarm

No.	Contents
71	An invalid file number or program number was entered. (Specified program number is not found.)
79	Verification operation found a mismatch between a program loaded into memory and the contents of the floppy
86	The dataset–ready signal (DR) for the input/output device is turned off. (The no file error or duplicate file error occurred on the input/output device because an invalid file number, program number, or file name was entered.

9 EDITING PROGRAMS

This chapter describes how to edit programs registered in the controller. Editing includes the insertion, modification, deletion, and replacement of words. Editing also includes deletion of the entire program and automatic insertion of sequence numbers. The extended part program editing function can copy, move, and merge programs. This chapter also describes program number search, sequence number search, word search, and address search, which are performed before editing the program.



9.1 INSERTING , ALTERING AND DELETING A WORD

This section outlines the procedure for inserting, modifying, and deleting a word in a program registered in memory.

Procedure for inserting, altering and deleting a word

1 Select **EDIT** mode.

2 Press PROG when CRT/MDI is used ; press **<PRGRM>** button when DPL/MDI is used.

3 Select a program to be edited.
If a program to be edited is selected, perform the operation 4.
If a program to be edited is not selected, search for the program number.

4 Search for a word to be modified.
·Scan method
·Word search method

5 Perform an operation such as altering, inserting, or deleting a word.

Explanation

- **Concept of word and editing unit**

A word is an address followed by a number. With a custom macro, the concept of word is ambiguous.
So the editing unit is considered here.
The editing unit is a unit subject to alteration or deletion in one operation.
In one scan operation, the cursor indicates the start of an editing unit.
An insertion is made after an editing unit.
Definition of editing unit
(i) Program portion from an address to immediately before the next address
(ii) An address is an alphabet, **IF**, **WHILE**, **GOTO**, **END**, **DO=**,or **; (EOB)**.
According to this definition, a word is an editing unit.
The word "word," when used in the description of editing, means an editing unit according to the precise definition.

> **WARNING**
> The user cannot continue program execution after altering, inserting, or deleting data of the program by suspending machining in progress by means of an operation such as a single block stop or feed hold operation during program execution. If such a modification is made, the program may not be executed exactly according to the contents of the program displayed on the screen after machining is resumed. So, when the contents of memory are to be modified by part program editing, be sure to enter the reset state or reset the system upon completion of editing before executing the program.

9.1.1 Word Search

A word can be searched for by merely moving the cursor through the text (scanning), by word search, or by address search.

Procedure for scanning a program

Procedure for CRT/MDI

1 Press the cursor key [→]

The cursor moves forward word by word on the screen; the cursor is displayed at a selected word.

2 Press the cursor key [←]

The cursor moves backward word by word on the screen; the cursor is displayed at a selected word.

Example) When Z1250.0 is scanned

```
Program                                O0050  N01234
 O0050  ;
 N01234  X100.0 Z1250.0  ;
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

3 Holding down the cursor key [→] or [←] scans words continuously.

4 The first word of the next block is searched for when the cursor key [↓] is pressed.

5 The first word of the previous block is searched for when the cursor key [↑] is pressed.

6 Holding down the cursor key [↓] or [↑] moves the cursor to the head of a block continuously.

7 Pressing the page key [PAGE ↓] displays the next page and searches for the first word of the page.

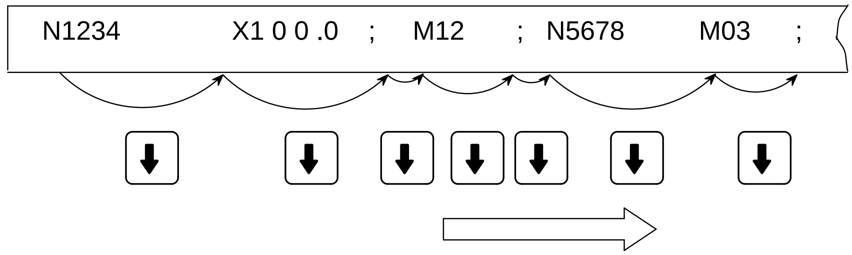
8 Pressing the page key [↑ PAGE] displays the previous page and searches for the first word of the page.

9 Holding down the page key [PAGE ↓] or [↑ PAGE] displays one page after another.

Procedure for DPL/MDI

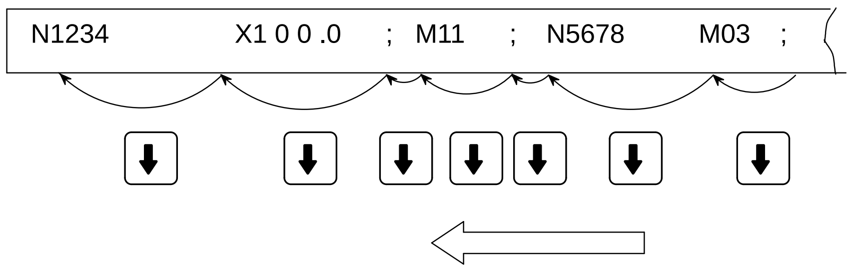
Scan is used per 1 word.

(a) Press the cursor<↓> key



The cursor moves forward word by word on the screen; the cursor is displayed below at the address character of a selected word.

(b) Press the cursor<↑> button



The cursor moves backward word by word on the screen; the cursor is displayed below at the address character of a selected word.

(c) Keep pushing the cursor<↓> or the cursor<↑> key to make a continuous search.

Procedure for searching a word

Example) Searching for M12

```
PROGRAM                              O0050  N01234
 O0050  ;
 N01234   X100.0  Z1250.0  ;  ←
 H12  ; ←
 N56789  M03  ;
 M02  ;
 %
```

N01234 is being searched for/ scanned currently.
M12 is searched for.

Procedure for CRT/MDI

1 Key in address M .

2 Key in 1 2 .

·M12 cannot be searched for if only M1 is keyed in.
·M09 cannot be searched for by keying in only M9.
To search for M09, be sure to key in M09.

3 Press the **[SRH↓]** key starts search operation.
Upon completion of search operation, the cursor is displayed at M12.
Pressing the **[SRH↑]** key rather than the **[SRH↓]** key performs search operation in the reverse direction.

Procedure for DPL/MDI

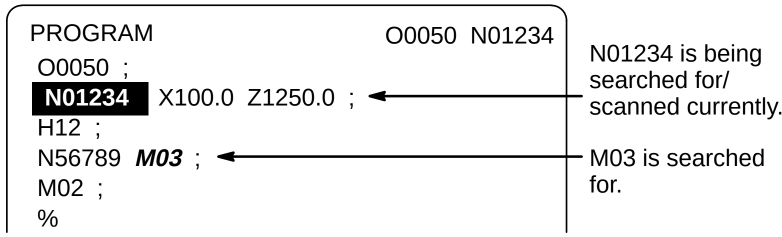
1 Key in address M .

2 Key in 1 2 .

3 Press cursor<↓> key and the search begins. When the search ends, M12 is displayed at the cursor.
Pressing cursor<↑> key searches in the backword direction.

Procedure for searching an address

Example) Searching for M03



Procedure for CRT/MDI

1 Key in address [M] .

2 Press the **[SRH↓]** key.
Upon completion of search operation, the cursor is displayed at M03. Pressing the **[SRH↑]** key rather than the **[SRH↓]** key performs search operation in the reverse direction.

Procedure for DPL/MDI

1 Key in address [M] .

2 Press cursor<↓> key and the search begins. When the search ends, M is displayed at the cursor.
Pressing cursor<↑> key searches in the backword direction.

Alarm

Alarm number	Description
71	The word or address being searched for was not found.

9.1.2 Heading a Program

The cursor can be jumped to the top of a program. This function is called heading the program pointer. This section describes the three methods for heading the program pointer.

Procedure for Heading a Program

Procedure for CRT/MDI

Method 1

1 Press RESET when the program screen is selected in EDIT mode. When the cursor has returned to the start of the program, the contents of the program are displayed from its start on the screen.

Method 2

Search for the program number.

1 Press address O ,when a program screen is selected in the **AUTO** or **EDIT** mode.

2 Input a program number.

3 Press the soft key **[O SRH]**.

Method 3

1 Select **[AUTO]** or **[EDIT]** mode.

2 Press PROG .

3 Press the **[(OPRT)]** key.

4 Press the **[REWIND]** key.

Procedure for DPL/MDI

Method 1

1 Perform program number search.

Method 2

1 Select **AUTO** or **EDIT** mode.

2 Press PROG button.

3 Press address O .

4 Press cursor<↑>.

9.1.3 Inserting a Word

Procedure for inserting a word

1 Search for or scan the word immediately before a word to be inserted.

2 Key in an address to be inserted.

3 Key in data.

4 Press the INSERT key.

Example of Inserting M15

Procedure for CRT/MDI

1 Search for or scan Z1250.

```
Program                                O0050  N01234
 O0050  ;
 N01234  X100.0  Z1250.0  ;   <------  Z1250.0 is searched for/ scanned.
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

2 Key in M 1 5 .

3 Press the INSERT key.

```
Program                                O0050  N01234
 O0050  ;
 N01234  X100.0  Z1250.0  M15  ;   <------  M15 is inserted.
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

Procedure for DPL/MDI

N1234 X100.0 ; M12 ; N5678 M03 ;

To be searched for

M15 to be inserted

1. Search for or scan the word immediately before the insertion location.
2. Key in M (an address to be inserted.)
3. Key in data.
4. Press <INSRT> key

N1234 X100.0 M15 ; M12 ; N5678 M03 ;

After M15 is inserted

> **NOTE**
> The last word of a block can be inserted as follows:
> Key in Z100 <EOB> (instead of Z100 <INSRT> <EOB> <INSERT>. The result is the same.)

9.1.4 Altering a Word

Procedure for altering a word

1 Search for or scan a word to be altered.

2 Key in an address to be inserted.

3 Key in data.

4 Press the ALTER key.

Example of changing M13 to M15

Procedure for CRT/MDI

1 Search for or scan M13.

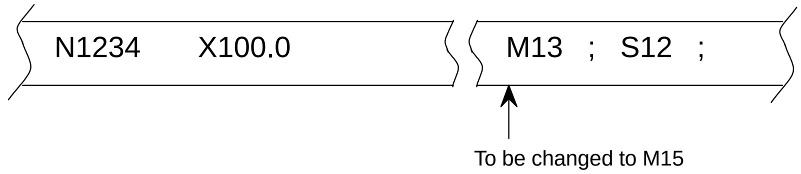
```
Program                                  O0050  N01234
 O0050 ;
 N01234  X100.0  Z1250.0 M13 ;   ←—— M13 is searched
 M12 ;                                for/scanned.
 N56789  M03 ;
 M02 ;
 %
```

2 Key in M 1 5 .

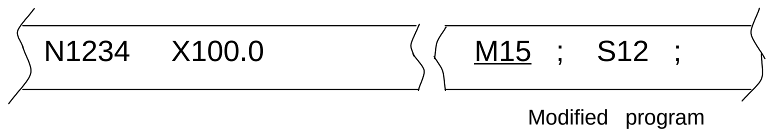
3 Press the ALTER key.

```
Program                                  O0050  N01234
 O0050 ;
 N1234  X100.0  Z1250.0 M15 ;   ←—— M13 is changed to
 M12 ;                               M15.
 N5678  M03 ;
 M02 ;
 %
```

Procedure for DPL/MDI



1. Search for/scan the word to be changed.
2. Key in the address to be modified. In the above example, key in address M.
3. Key in data.
4. Press <ALTER> key.
   <M> <1> <5> <ALTER>



9.1.5 Deleting a Word

Procedure for deleting a word

1 Search for or scan a word to be deleted.

2 Press the DELETE key.

Example of deleting X100.0

Procedure for CRT/MDI

1 Search for or scan X100.0.

```
Program                                  O0050  N01234
 O0050  ;
 N01234  X100.0  Z1250.0  M15  ;
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

X100.0 is searched for/ scanned.

2 Press the DELETE key.

```
Program                                  O0050  N01234
 O0050  ;
 N01234  Z1250.0  M15  ;
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

X100.0 is deleted.

Procedure for DPL/MDI

```
N1234   X100.0   Y1250   M15  ;   M12  ;
```

Y1250 to be deleted

1 Search for/scan the word to be deleted.

2 Press the <DELET> button.

```
N1234  X100.0  M15  ;        M12  ;
```

Program after deletion

9.2 DELETING BLOCKS

A block or blocks can be deleted in a program.

9.2.1 Deleting a Block

The procedure below deletes a block up to its EOB code; the cursor advances to the address of the next word.

Procedure for deleting a block

1 Search for or scan address N for a block to be deleted.

2 Key in EOB .

3 Press the DELETE .

Example of deleting a block of No.1234

Procedure for CRT/MDI

1 Search for or scan N01234.

```
Program                              O0050  N01234
 O0050  ;
 N01234   Z1250.0  M15  ;
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

N01234 is searched for/ scanned.

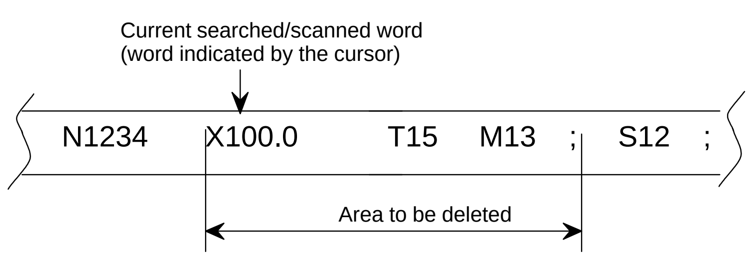
2 Key in EOB .

3 Press the DELETE key.

```
Program                              O0050  N01234
 O0050  ;
 M12  ;
 N56789  M03  ;
 M02  ;
 %
```

Block containing N01234 has been deleted.

Procedure for DPL/MDI



Pressing the <EOB> and the <DELET> key deletes up to an EOB and causes the cursor to move to below the address character of the next word.

9.2.2 Deleting Multiple Blocks

The blocks from the currently displayed word to the block with a specified sequence number can be deleted.

Procedure for deleting multiple blocks

1 Search for or scan a word in the first block of a portion to be deleted.

2 Key in address N .

3 Key in the sequence number for the last block of the portion to be deleted.

4 Press the DELETE key.

Example of deleting blocks from a block containing N01234 to a block containing N56789

Procedure for CRT/MDI

1 Search for or scan N01234.

```
Program                              O0050  N01234
 O0050 ;
 N01234  Z1250.0  M15 ;
 M12 ;
 N56789  M03 ;
 M02 ;
 %
```

N01234 is searched for/ scanned.

2 Key in N 5 6 7 8 9 .

```
Program                              O0050  N01234
 O0050 ;
 N01234  Z1250.0  M15 ;
 M12 ;
 N56789  M03 ;
 M02 ;
 %
```

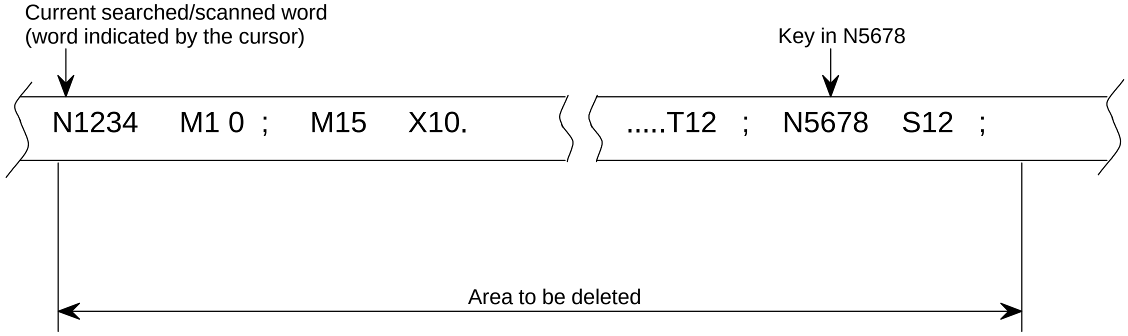
Underlined part is deleted.

3 Press the DELETE key.

```
Program                              O0050  N01234
 O0050 ;
 M02 ;
 %
```

Blocks from block containing N01234 to block containing N56789 have been deleted.

Procedure for DPL/MDI



1 Key in address N.

2 Key in, 5, 6, 7, and 8 in this example.

3 Press the <DELET> key. The program up to the N5678 block is deleted. The cursor moves to the address next to the deleted block.

9.3 PROGRAM NUMBER SEARCH

When memory holds multiple programs, a program can be searched for. There are three methods as follows.

Procedure for program number search

Procedure for CRT/MDI

Method 1

1 Select **EDIT** or **AUTO** mode.

2 Press PROG to display the program screen.

3 Key in address O .

4 Key in a program number to be searched for.

5 Press the **[O SRH]** key.

6 Upon completion of search operation, the program number searched for is displayed in the upper–right corner of the CRT screen
If the program is not found , P/S alarm No. 71 occurs.

Method 2

1 Select **EDIT** or **AUTO** mode.

2 Press PROG to display the program screen.

3 Press the **[O SRH]** key.
In this case, the next program in the directory is searched for .

Method 3

This method searches for the program number (0001 to 0255) corresponding to a signal on the machine tool side to start automatic operation. Refer to the relevant manual prepared by the machine tool builder for detailed information on operation.

1 Select **AUTO** mode.

2 Set the reset state(*1)
·The reset state is the state where the LED for indicating that automatic operation is in progress is off. (Refer to the relevant manual of the machine tool builder.)

3 Set the program number selection signal on the machine tool side to a number from 01 to 255.
·If the program corresponding to a signal on the machine tool side is not registered, P/S alarm (No. 59) is raised.

4 Press the cycle start button.
·When the signal on the machine tool side represents 00, program number search operation is not performed.

Procedure for DPL/MDI

Method 1

1. Select **EDIT** or **AUTO** mode.
2. Press the PRGRM button.
3. Key in address O .
4. Key in a program No. to be searched.
5. Press cursor<↓> button.
6. When searching is over, the program No. searching is indicated at the right top of the DPL screen.

Method 2

1. Select **EDIT** or **AUTO** mode.
2. Press the PRGRM button.
3. Key in address O .
4. Press cursor<↓> button. If it is kept pushed in the **EDIT** mode, the registered programs are displayed sequentially.

> **NOTE**
> After displaying all program numbers registered, the display retums to the first one.

Method 3

1. Select **AUTO** mode.
2. Reset the machine.
3. Set the signal for selecting the program No. on the machine side at 01 to 255. (For details, see the instruction manual of the machine tool builder.)
4. Press the cycle start button.
   The program No. (0001 to 0255) correspoding to the signal on the machine side is seached, and automatic operation starts.
   - The program No. is not searched when the signal on the machine side is ’00’.
   - In the reset condition, the cycle operation lamp is off.
     (See the instruction manual of the machine tool builder.)

Alarm

No.	Contents
59	The program with the selected number cannot be searched during external program number search.
71	The specified program number was not found during program number search.

9.4 SEQUENCE NUMBER SEARCH

Sequence number search operation is usually used to search for a sequence number in the middle of a program so that execution can be started or restarted at the block of the sequence number.

Example) Sequence number 02346 in a program (O0002) is searched for.

Program

O0001 ;
N01234 X100.0 Z100.0 ;
M12 ;
:

Selected program → ***O0002 ;***
N02345 X20.0 Z20.0 ;
Target sequence number is found. → ***N02346 X10.0 Y10.0 ;***
:
O0003 ;
:

This section is searched starting at the beginning. (Search operation is performed only within a program.)

Procedure for sequence number search

Procedure for CRT/MDI

1 Select **AUTO** mode.

2 Press PROG .

3 ·If the program contains a sequence number to be searched for, perform the operations 4 to 7 below.
·If the program does not contain a sequence number to be searched for,select the program number of the program that contains the sequence number to be searched for.

4 Key in address N .

5 Key in a sequence number to be searched for.

6 Press the **[N SRH]** key.

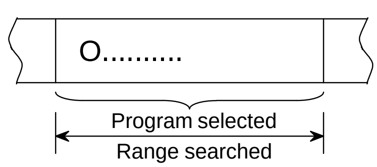
7 Upon completion of search operation, the sequence number searched for is displayed in the upper–right corner of the CRT screen.
If the specified sequence number is not found in the program currently selected, P/S alarm 60 occurs.

Procedure for DPL/MDI

1 Set the mode select switch to **AUTO**.

2 Press PRGRM button.

3 Select the program number to which the sequence number to be searched for belongs.
Proceed to 4 to 7 when the program contains the sequence number ; Otherwise, execute Program Number Search to select a program number to which the sequence number belongs.



4 Key in address N.

5 Key in a sequence number to be searched for.

6 Press the **CURSOR**<↓> key.

7 Upon completion of search operation, the sequence number searched for is displayed in the DPL screen.

Explanations

• Operation during Search

Those blocks that are skipped do not affect the controller. This means that the data in the skipped blocks such as coordinates and M code does not alter the controller coordinates and modal values.
So, in the first block where execution is to be started or restarted by using a sequence number search command, be sure to enter required M code and coordinates. A block searched for by sequence number search usually represents a point of shifting from one process to another. When a block in the middle of a process must be searched for to restart execution at the block, specify M code, G codes, coordinates, and so forth as required from the MDI after closely checking the machine tool and status of the controller at that point.

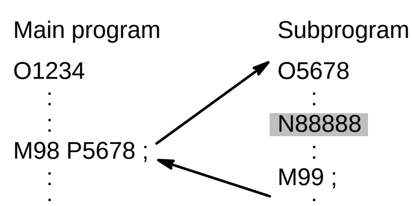
• Checking during search

During search operation, the following checks are made:
·Optional block skip
·P/S alarm (No. 003 to 010)

Restrictions

• Searching in sub–program

During sequence number search operation, M98Pxxxx (subprogram call) is not executed. So an alarm (No.060) is raised if an attempt is made to search for a sequence number in a subprogram called by the program currently selected.



If an attempt is made to search for N8888 in the example above, an alarm is raised.

Alarm

Number	Contents
60	Command sequence number was not found in the sequence number search.

9.5 DELETING PROGRAMS

Programs registered in memory can be deleted,either one program by one program or all at once. Also, More than one program can be deleted by specifying a range.

9.5.1 Deleting One Program

A program registered in memory can be deleted.

Procedure for deleting one program

Procedure for CRT/MDI

1 Select the **EDIT** mode.

2 Press PROG to display the program screen.

3 Key in address O .

4 Key in a desired program number.

5 Press the DELETE key.
The program with the entered program number is deleted.

Procedure for DPL/MDI

1 Set the mode select switch to **EDIT**.

2 Press the PRGRM button.

3 Key in address O .

4 Key in the program number.

5 When you push the **<DELET>** button, the program with the keyed in number will be deleted.

9.5.2 Deleting All Programs

All programs registered in memory can be deleted.

Procedure for deleting all programs

Procedure for CRT/MDI

1 Select the **EDIT** mode.

2 Press PROG to display the program screen.

3 Key in address O .

4 Key in –9999.

5 Press edit key DELETE to delete all programs.

Procedure for DPL/MDI

1 Set the mode select switch to **EDIT**.

2 Press the PRGRM button.

3 Key in address O .

4 Key in **[–]**, **[9]**, **[9]**, **[9]**, and **[9]** and push the **<DELET>** button.

9.5.3 Deleting More Than One Program by Specifying a Range

Programs within a specified range in memory are deleted.

Procedure for deleting more than one program by specifying a range

Procedure for CRT/MDI

1 Select the **EDIT** mode.

2 Press PROG to display the program screen.

3 Enter the range of program numbers to be deleted with address and numeric keys in the following format:
OXXXX,OYYYY
where XXXX is the starting number of the programs to be deleted and YYYY is the ending number of the programs to be deleted.

4 Press edit key DELETE to delete programs No. XXXX to No. YYYY.

9.6 EXTENDED PART PROGRAM EDITING FUNCTION

With the extended part program editing function, the operations described below can be performed using soft keys on CRT for programs that have been registered in memory.

Following editing operations are available :

·All or part of a program can be copied or moved to another program.

·One program can be merged at free position into other programs.

·A specified word or address in a program can be replaced with another word or address.

9.6.1 Copying an Entire Program

A new program can be created by copying a program.

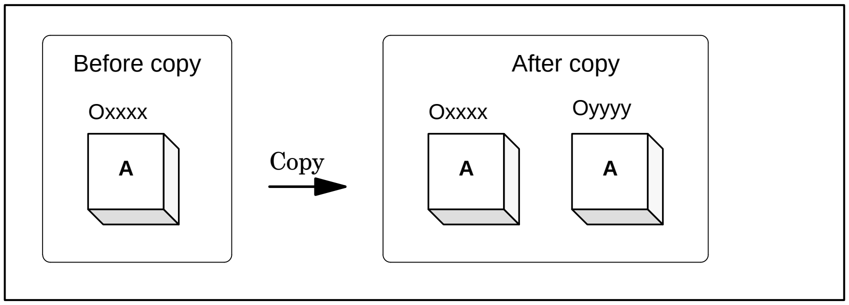


Fig. 9.6.1 Copying an Entire Program

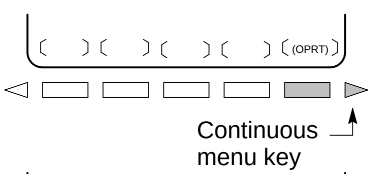
In Fig. 9.6.1, the program with program number xxxx is copied to a newly created program with program number yyyy. The program created by copy operation is the same as the original program except the program number.

Procedure of copying an entire program

Procedure for CRT/MDI

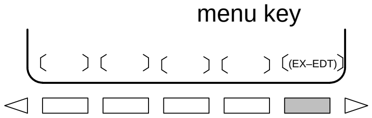
1 Enter the **EDIT** mode.

2 Press function key PROG .

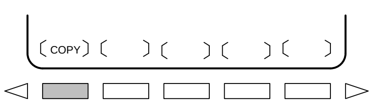


3 Press soft key **[(OPRT)]**.

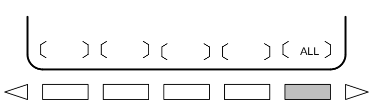
4 Press the continuous menu key.



5 Press soft key **[EX–EDT]**.



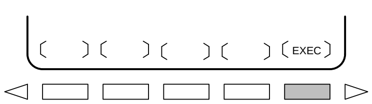
6 Check that the screen for the program to be copied is selected and press soft key **[COPY]**.



7 Press soft key **[ALL]**.

Numeric keys 0 ~ 9

8 Enter the number of the new program (with only numeric keys) and press the INPUT key.



9 Press soft key **[EXEC]**.

9.6.2 Copying Part of a Program

A new program can be created by copying part of a program.

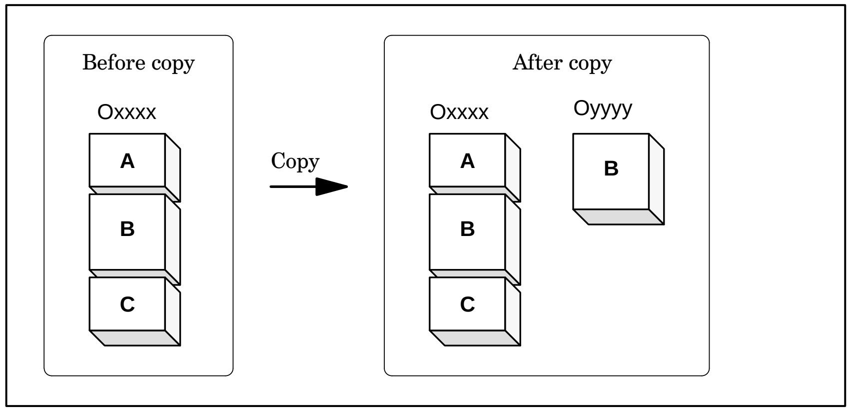


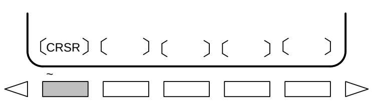
Fig. 9.6.2 Copying Part of a Program

In Fig. 9.6.2, part B of the program with program number xxxx is copied to a newly created program with program number yyyy. The program for which an editing range is specified remains unchanged after copy operation.

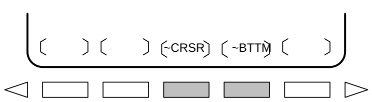
Procedure for copying part of a program

Procedure for CRT/MDI

1 Perform steps **1** to **6** in subsection **9.6.1**.



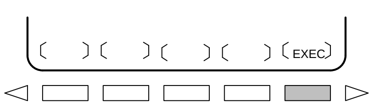
2 Move the cursor to the start of the range to be copied and press soft key **[CRSR~]**.



3 Move the cursor to the end of the range to be copied and press soft key **[~CRSR]** or **[~BTTM]** (in the latter case, the range to the end of the program is copied regardless of the position of the cursor).

Numeric keys 0 ~ 9

4 Enter the number of the new program (with only numeric keys) and press the INPUT key.



5 Press soft key **[EXEC]**.

9.6.3 Moving Part of a Program

A new program can be created by moving part of a program.

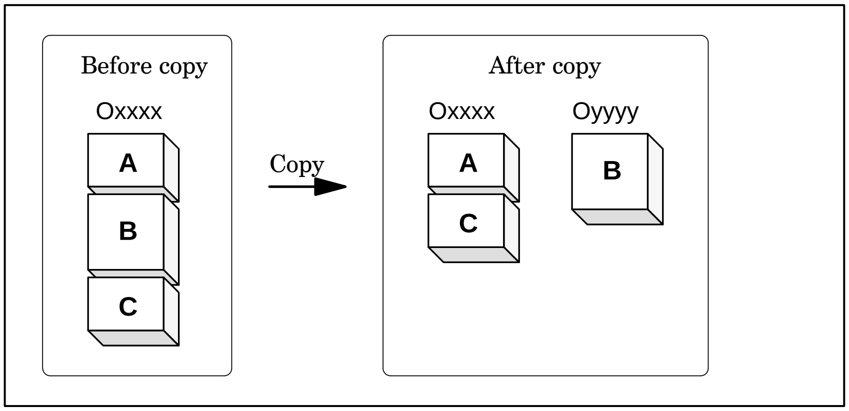


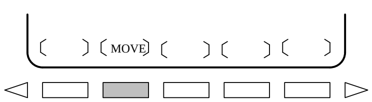
Fig. 9.6.3 Moving Part of a Program

In **Fig. 9.6.3**, part B of the program with program number xxxx is moved to a newly created program with program number yyyy; part B is deleted from the program with program number xxxx.

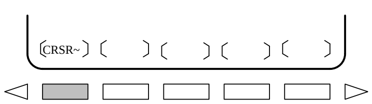
Procedure for moving part of a program

Procedure for CRT/MDI

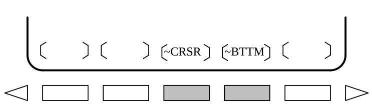
1 Perform steps 1 to 5 in subsection 9.6.1.



2 Check that the screen for the program to be moved is selected and press soft key **[MOVE]**.



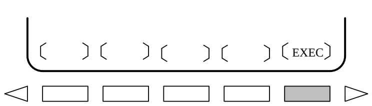
3 Move the cursor to the start of the range to be moved and press soft key **[CRSR~]**.



4 Move the cursor to the end of the range to be moved and press soft key **[~CRSR]** or **[~BTTM]**(in the latter case, the range to the end of the program is copied regardless of the position of the cursor).

Numeric keys 0 ~ 9

5 Enter the number of the new program (with only numeric keys) and press the INPUT key.



6 Press soft key **[EXEC]**.

9.6.4 Merging a Program

Another program can be inserted at an arbitrary position in the current program.

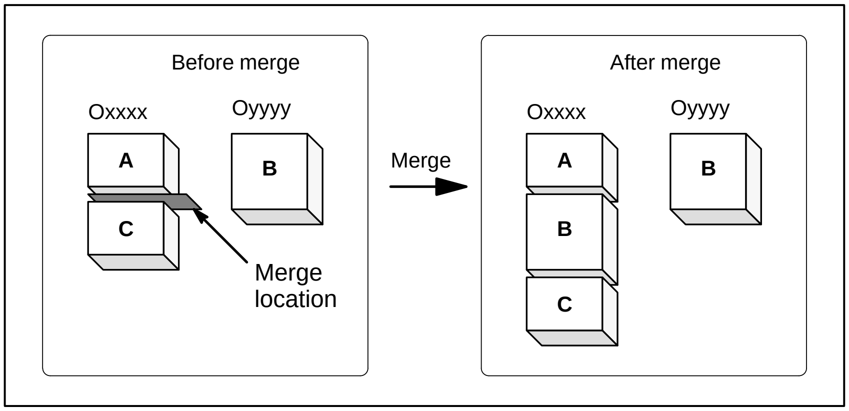
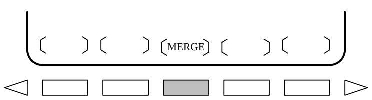


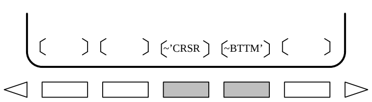
Fig. 9.6.4 Merging a program at a specified location

In **Fig. 9.6.4,** the program with program number XXXX is merged with the program with program number YYYY. The OYYYY program remains unchanged after merge operation.

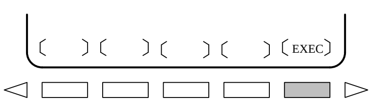
Procedure for merging a program

Procedure for CRT/MDI





Numeric keys 0 ~ 9



1 Perform steps **1** to **5** in subsection **9.6.1**.

2 Check that the screen for the program to be edited is selected and press soft key **[MERGE]**.

3 Move the cursor to the position at which another program is to be inserted and press soft key **[~'CRSR]** or **[~BTTM']**(in the latter case, the end of the current program is displayed).

4 Enter the number of the program to be inserted (with only numeric keys) and press the INPUT key.

5 Press soft key **[EXEC]**.
The program with the number specified in step 4 is inserted before the cursor positioned in step 3.

9.6.5 Supplementary Explanation for Copying, Moving and Merging

Explanations

● Setting an editing range

The setting of an editing range start point with **[CRSR~]** can be changed freely until an editing range end point is set with **[~CRSR]** or **[~BTTM]**. If an editing range start point is set after an editing range end point, the editing range must be reset starting with a start point.
The setting of an editing range start point and end point remains valid until an operation is performed to invalidate the setting.
One of the following operations invalidates a setting:

· An edit operation other than address search, word search/scan, and search for the start of a program is performed after a start point or end point is set.

· Processing is returned to operation selection after a start point or end point is set.

● Without specifying a program number

In copying program and moving program, if **[EXEC]** is pressed without specifying a program number after an editing range end point is set, a program with program number O0000 is registered as a work program. This O0000 program has the following features:

· The program can be edited in the same way as a general program. (Do not run the program.)

· If a copy or move operation is newly performed, the previous information is deleted at execution time, and newly set information (all or part of the program) is reregistered. (In merge operation, the previous information is not deleted.) However, the program, when selected for foreground operation, cannot be reregistered in the background. (A BP/S140 alarm is raised.) When the program is reregistered, a free area is produced. Delete such a free area with the RESET key.

· When the program becomes unnecessary, delete the program by a normal editing operation.

● Editing when the system waiting for a program number to be entered

When the system is waiting for a program number to be entered, no edit operation can be performed.

Restrictions

● Number of digits for program number

If a program number is specified by 5 or more digits, a format error is generated.

Alarm

Alarm no.	Contents
070	Memory became insufficient while copying or inserting a program. Copy or insertion is terminated.
101	The power was interrupted during copying, moving, or inserting a program and memory used for editing must be cleared. When this alarm occurs, press the key PROG while pressing function key RESET . Only the program being edited is deleted.

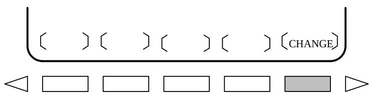
9.6.6 Replacement of Words and Addresses

Replace one or more specified words.
Replacement can be applied to all occurrences or just one occurrence of specified words or addresses in the program.

Procedure for hange of words or addresses

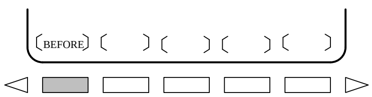
Procedure for CRT/MDI

1 Perform steps 1 to 5 in subsection 9.6.1.



2 Press soft key **[CHANGE]**.

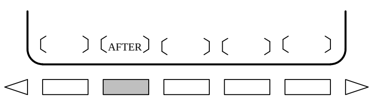
3 Enter the word or address to be replaced.



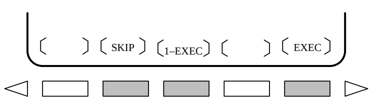
4 Press soft key **[BEFORE]**.

5 Enter the new word or address.

6 Press soft key **[AFTER]**.



7 Press soft key **[EXEC]** to replace all the specified words or addresses after the cursor.
Press soft key **[1–EXEC]** to search for and replace the first occurrence of the specified word or adress after the cursor.
Press soft key **[SKIP]** to only search for the first occurrence of the specified word or address after the cursor.



EXAMPLES

- **Replace X100 with Y200**

[CHANGE] X 1 0 0 **[BEFORE]** Y 2 0 0 **[AFTER][EXEC]**

- **Replace X100Y200 with X30**

[CHANGE] X 1 0 0 Y 2 0 0 **[BEFORE]** X 3 0 **[AFTER][EXEC]**

- **Replace IF with WHILE**

[CHANGE] I F **[BEFORE]** W H I L E **[AFTER] [EXEC]**

- **Replace X with ,C10**

[CHANGE] X **[BEFOR]** , C 1 0 **[AFTER][EXEC]**

Explanation

- **Replacing custom macros**

The following custom macro words are replaceable:
IF, WHILE, GOTO, END, DO, BPRNT, DPRINT, POPEN, PCLOS
The abbreviations of custom macro words can be specified.
When abbreviations are used, however, the screen displays the abbreviations as they are key input, even after soft key **[BEFORE]** and **[AFTER]** are pressed.

Restrictions

- **The number of characters for replacement**

Up to 15 characters can be specified for words before or after replacement. (Sixteen or more characters cannot be specified.)

- **The characters for replacement**

Words before or after replacement must start with a character representing an address.(A format error occurs.)

9.7 EDITING OF CUSTOM MACROS

Unlike ordinary programs, custom macro programs are modified, inserted, or deleted based on editing units.
Custom macro words can be entered in abbreviated form.
Comments can be entered in a program.
Refer to the section 10.1 for the comments of a program.

Explanations

- **Editing unit**

When editing a custom macro already entered, the user can move the cursor to each editing unit that starts with any of the following characters and symbols:

(a) Address

(b) # located at the start of the left side of a substitution statement

(c) /, (,=, and ;

(d) First character of IF, WHILE, GOTO, END, DO, POPEN, BPRNT, DPRNT and PCLOS

On the CRT screen, a blank is placed before each of the above characters and symbols.
(Example) Head positions where the cursor is placed
N001 X–#100 ;
#1 =123 ;
N002 /2 X[12/#3] ;
N003 X–SQRT[#3/3*[#4+1]] ;
N004 X–#2 Z#1 ;
N005 #5 =1+2–#10 ;
IF[#1NE0] GOTO10 ;
WHILE[#2LE5] DO1 ;
#[200+#2] =#2*10 ;
#2 =#2+1 ;
END1 ;

- **Abbreviations of custom macro word**

When a custom macro word is altered or inserted, the first two characters or more can replace the entire word.
Namely,

WHILE → WH	**GOTO → GO**	**XOR → XO**	**AND → AN**
SIN → SI	**COS → CO**	**TAN → TA**	**ATAN → AT**
SQRT → SQ	**ABS → AB**	**BCD → BC**	**BIN → BI**
FIX → FI	**FUP → FU**	**ROUND → RO**	**END → EN**
POPEN → PO	**BPRNT → BP**	**DPRNT → DP**	**PCLOS→PC**

(Example) Keying in
WH [AB [#2] LE RO [#3]]
has the same effect as
WHILE [ABS [#2] LE ROUND [#3]]
The program is also displayed in this way.

Editing macro statements using DPL

- **Switching the screen**

(a) To switch from the ordinary screen to the macro statement editing screen, press the <INPUT> key.

Ordinary screen

```
#100<=#101*1.0>
```

Macro statement editing screen

```
>■#101*1.0
```

The blinking cursor ■ is positioned to the ”=”.

(b) Pressing the <ALTER> key registers the displayed character string, overwriting the previous data, thus terminating edit mode.

(c) Pressing the <CAN> key cancels the macro statement editing screen.

(d) Pressing a function key to switch the screen also cancels the macro statement editing screen.

- **Edit**

(a) Pressing the <↑> key moves the cursor to the left. Pressing the <↓> key moves the cursor to the right.

Example

```
>■#101*1.0
```

The blinking cursor ■ is positioned to the ”=”.

Pressing the <↓> key six times positions the cursor as follows:

```
>=#101*■.0
```

The blinking cursor ■ is positioned to the ”1”.

(b) Pressing the <DELET> key deletes the character at the cursor position.

Example

```
>=#101*■.0
```

The blinking cursor ■ is positioned to the ”1”.

Pressing the <DELET> key once positions the cursor as follows:

```
>=#101*■0
```

The blinking cursor ■ is positioned to the ”.”.

(c) Pressing the <INSRT> key inserts a blank at the cursor position.

Example

```
>=#101*■0
```

The blinking cursor ■ is positioned to the ”.”.

Pressing the <INSRT> key twice positions the cursor as follows:

```
>=#101*■.0
```

The blinking cursor ■ is positioned to the blank.

(d) Pressing an alphanumeric key inserts the corresponding character at the cursor position.

(e) Pressing the <ALTER> key registers the displayed character string, overwriting the previous data.

WARNING

1 In step (a) of the procedure for switching the screen, if the macro statement exceeds 31 characters, only the first 31 characters can be edited. If the <ALTER> key is pressed when the macro statement of more than 31 characters is displayed, the first 31 characters are registered, the subsequent characters being lost.

2 This function does not perform syntax check.

NOTE

1 Up to 31 characters can be edited.

2 If an alarm occurs during editing, editing is canceled and the alarm screen appears.

3 Pressing the <#>, </>, and <EOB> keys displays the following characters cyclically:

<#> key
<#> ⇒ <[> ⇒ <]> ⇒ <*> ⇒ <=>
</> key
</> ⇒ <A> ⇒ <B> ⇒ <C> ⇒ <D> ⇒ <E> ⇒ <L>
<EOB> key
<;> ⇒ <U> ⇒ <V> ⇒ <W> ⇒ <X> ⇒ <Y> ⇒ <Z>

9.8 BACKGROUND EDITING

Editing a program while executing another program is called background editing. The method of editing is the same as for ordinary editing (foreground editing).
A program edited in the background should be registered in foreground program memory by performing the following operation:
During background editing, all programs cannot be deleted at once.

Procedure for background editing

Procedure for CRT/MDI

1 Enter **EDIT** or **AUTO** mode.
AUTO mode is allowed even while the program is being executed.

2 Press function key PROG .

3 Press soft key **[(OPRT)]**, then press soft key **[BG–EDT]**.
The background editing screen is displayed (PROGRAM (BG–EDIT) is displayed at the top left of the screen).

4 Edit a program on the background editing screen in the same way as for ordinary program editing.

5 After editing is completed, press soft key **[(OPRT)]**, then press soft key **[BG–END].** The edited program is registered in foreground program memory.

Procedure for DPL/MDI

1 Display the background edit screen by pressing the <PRGRM> button while pressing and holding the <CAN> key.

```
{ }
```

2 Specify a program to be edited.
a) When generating a new program:

<O> Program No. <INSERT>

b) When editing the existing program:

<O> Program No. <↓>

3 Program edit
The program editing is the same as with the foreground program editing operation.

```
{01000} ;
N10  G92  X0  ;
```

4 End of Background edit
It is necessary to save a program completely edited in background into the foreground program memory.
Press the <PRGRM> button while pressing and holding the <CAN> key.

Explanation

- **Alarms during background editing**

Alarms that may occur during background editing do not affect foreground operation. Conversely, alarms that may occur during foreground operation do not affect background editing. In background editing, if an attempt is made to edit a program selected for foreground operation, a BP/S alarm (No. 140) is raised. On the other hand, if an attempt is made to select a program subjected to background editing during foreground operation (by means of subprogram calling or program number search operation using an external signal), a P/S alarm (Nos. 059, 078) is raised in foreground operation. As with foreground program editing, P/S alarms occur in background editing. However, to distinguish these alarms from foreground alarms, BP/S is displayed in the data input line on the background editing screen.

10 CREATING PROGRAMS

Programs can be created using any of the following methods:

- MDI keyboard
- PROGRAMMING IN TEACH IN MODE

This chapter describes creating programs using the MDI panel. This chapter also describes the automatic insertion of sequence numbers.

10.1 CREATING PROGRAMS USING THE MDI PANEL

Programs can be created in the EDIT mode using the program editing functions described in Chapter 9.

Procedure for Creating Programs Using the MDI Panel

Procedure for CRT/MDI

1 Enter the **EDIT** mode.

2 Press the PROG key.

3 Press address key O and enter the program number.

4 Press the INSERT key.

5 Create a program using the program editing functions described in Chapter 9.

Procedure for DPL/MDI

1 Select EDIT mode.

2 Press the <PRGRM> key.

3 Key in address O.

4 Enter the number of the program to be registered.

5 Press the <INSRT> key.
By pressing this key, the entered program number will be registered. Enter each word of the program followed by the <INSRT> key to register it. (See the section on word insertion.)

Explanation

- **Comments in a program**

Comments can be written in a program using the control in/out codes.

Example) O0001 (FANUC POWER Mate) ;
M08 (COOLANT ON) ;

· When the INSERT key is pressed after the control–out code ”(”,comments, and control–in code ”)” have been typed, the typed comments are registered.

· When the INSERT key is pressed midway through comments, to enter the rest of comments later, the data typed before the INSERT key is pressed may not be correctly registered (not entered, modified, or lost) because the data is subject to an entry check which is performed in normal editing.

Note the following to enter a comment:

· Control–in code ”)” cannot be registered by itself.

· Comments entered after the INSERT key is pressed must not begin with a number, space, or address O.

· If an abbreviation for a macro is entered, the abbreviation is converted into a macro word and registered (see Section 9.7).

· Address O and subsequent numbers, or a space can be entered but are omitted when registered.

10.2 AUTOMATIC INSERTION OF SEQUENCE NUMBERS

Sequence numbers can be automatically inserted in each block when a program is created using the MDI keys in the EDIT mode.
Set the increment for sequence numbers in parameter 3216.

Procedure for automatic insertion of sequence numbers

Procedure for CRT/MDI

1 Set 1 for SEQUENCE NO. (see subsjection 11.4.2).

2 Enter the **EDIT** mode.

3 Press PROG to display the program screen.

4 Search for or register the number of a program to be edited and move the cursor to the EOB (;) of the block after which automatic insertion of sequence numbers is started. When a program number is registered and an EOB (;) is entered with the INSERT key, sequence numbers are automatically inserted starting with 0. Change the initial value, if required, according to step 10, then skip to step 7.

5 Press address key N and enter the initial value of N.

6 Press INSERT

7 Enter each word of a block.

8 Press EOB.

9 Press INSERT. The EOB is registered in memory and sequence numbers are automatically inserted. For example, if the initial value of N is 10 and the parameter for the increment is set to 2, N12 inserted and displayed below the line where a new block is specified.

```
PROGRAM                                      O0040  N00012

 O0040 ;
 N10 G92 X0 Y0 Z0 ;
 N12
 %




>_
 EDIT  ****   ***   ***
(PRGRM)( LIB )(      )(      )( (OPRT) )
```

10 · In the example above, if N12 is not necessary in the next block, pressing the DELETE key after N12 is displayed deletes N12.

· To insert N100 in the next block instead of N12, enter N100 and press ALTER after N12 is displayed. N100 is registered and initial value is changed to 100.

Procedure for DPL/MDI

1 Set the setting parameter SEQ to 1.

2 Select EDIT mode.

3 Press <PRGRM> key.

4 Key in address N.

5 Key in the initial value of N, e.g. 10.

6 Press <INSRT> key.

7 Insert each word of the data in one block.

8 Key in <EOB>.

9 Press <INSRT> key. EOB is stored in the memory. In case 2 is set at the incremental value parameter, N12 is inserted to the next line and indicated.

· If N12 is desired not to be inserted in the next block in the example above, Press the <DELET> key to delet N12.

· If N100 is desired to be inserted to the next block instead of N12 in the example above, key in N100 and Press < ALTER> key. With this, N100 is registered, and the initial value is changed to 100.

10.3 CREATING PROGRAMS IN TEACH IN MODE

In the **TEACH IN JOG** mode or **TEACH IN STEP** mode, a machine position along the X, Y, and Z axes obtained by manual operation is stored in memory as a program position to create a program.
The words other than X, Y, and Z, which include O, N, G, R, F, C, M, P, Q, and EOB, can be stored in memory in the same way as in **EDIT** mode.

Procedure for Creating Programs in TEACH IN Mode

Procedure for CRT/MDI

The procedure described below can be used to store a machine position along the X, Y, and Z axes.

1 Select the **TEACH IN JOG** mode or **TEACH IN STEP** mode.

2 Move the tool to the desired position with jog or step feed.

3 Press PROG key to display the program screen. Search for or register the number of a program to be edited and move the cursor to the position where the machine position along each axis is to be registered (inserted).

4 Key in address X .

5 Press the INSERT key. Then a machine position along the X axis is stored in memory.

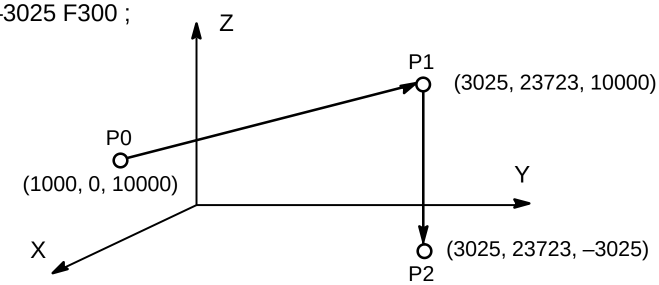
(Example) X10.521 Absolute positon (for mm input)
X10521 Data stored in memory

6 Similarly, key in Y , then press the INSERT key. Then a machine position along the Y axis is stored in memory. Further, key in Z , then press the INSERT key. Then a machine position along the Z axis is stored in memory.

All coordinates stored using this method are absolute coordinates.

Examples for CRT/MDI

```
O1234 ;
N1 G92 X10000 Y0 Z10000 ;
N2 G00 G90 X3025 Y23723 ;
N3 G01 Z–3025 F300 ;
N4 M02 ;
```



1 Set the setting data SEQUENCE NO. to 1 (on). (The incremental value parameter (No. 3212) is assumed to be “1”.)

2 Select the **TEACH IN STEP** mode.

3 Make positioning at position P0 by step feed.

4 Select the program screen.

5 Enter program number O1234 as follows:

[O] [1] [2] [3] [4] [INSERT]

This operation registers program number O1234 in memory.
Next, press the following keys:

[EOB] [INSERT]

An EOB (;) is entered after program number O1234. Because no number is specified after N, sequence numbers are automatically inserted for N0 and the first block (N1) is registered in memory.

6 Enter the P0 machine position for data of the first block as follows:

[G] [9] [2] [INSERT] [X] [INSERT] [Y] [INSERT] [Z] [INSERT] [EOB] [INSERT]

This operation registers G92X10000Y0Z10000; in memory. The automatic sequence number insertion function registers N2 of the second block in memory.

7 Position the tool at P1 with the step feed.

8 Enter the P1 machine position for data of the second block as follows:

[G] [0] [0] [INSERT] [G] [9] [0] [X] [INSERT] [Y] [INSERT] [EOB] [INSERT]

This operation registers G00G90X3025Z23723; in memory. The automatic sequence number insertion function registers N3 of the third block in memory.

9 Position the tool at P2 with the step feed.

10 Enter the P2 machine position for data of the third block as follows:

[G] [0] [1] [INSERT] [Z] [INSERT] [F] [3] [0] [0] [INSERT] [EOB] [INSERT]

This operation registers G01Z –3025F300; in memory.
The automatic sequence number insertion function registers N4 of the fourth block in memory.

11 Register M02; in memory as follows:

[M] [0] [2] [INSERT] [EOB] [INSERT]

N5 indicating the fifth block is stored in memory using the automatic sequence number insertion function. Press the [DELETE] key to delete it.

This completes the registration of the sample program.

Procedure for DPL/MDI

It is possible to register the machine position (work coordinate system's current position) in the memory in the following procedure.

1 Select the TEACH IN JOG mode or TEACH IN STEP mode.

2 Move the machine to the required position.

3 Press the <PRGRM> key.

4 Enter the address X.

5 Press the <INSRT> key, then the machine position along the X axis is stored in the memory.

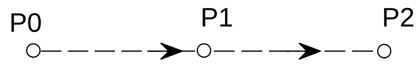
(Example) X10.521 Absolute position (for metric input)
X10.521 Content stored in the program

6 Enter address <Y>, then press the <INSRT> key. The machine position on the Y-axis is stored in the memory.

- After entering the address X, Y enter a numerical value and Press <INSRT> key, then the value entered is added to the machine position. This is used to correct the machine position through key entry.
- The coordinate value registered in this way will be an absolute coordinate value. Enter G90 (Absolute programming) at the beginning of the program.
- The command to be entered before and after machine position shall be entered by the same operation as that conducted in the EDIT mode before and after registering the machine position, respectively.
- Insert the <EOB>, the block registration completes.

Examples for DPL/MDI

(Example) O1234 N1 G92X -; P0
N2 G00X -; P1
N3 G01X -F300; P2
N4 M02;



The program of the above example is stored in the following procedure.

1 Set the setting parameter “SEQ” to 1 (For the incremental value parameter, “1” is assumed.)

2 Select the TEACH IN STEP mode.

3 Make positioning at P0 with the step feed.

4 Press the <PRGRM> button.

5 Enter the address 0, numeric value 1234, and Press the <INSRT> key. Then the program number 01234 is stored in the memory.

6 Enter the address N, numeric value 1, and Press the <INSRT> key. The sequence number 1 is stored in the memory as the initial value of the automatic insertion.

7 Enter the address G, numeric value 92, and Press the <INSRT> key. Then the G92 is stored in the memory.

8 Enter address X and press <EOB>. The machine position, P0, is registered in memory.

9 When X is entered and <EOB> is pressed in step (8), EOB is inserted. The input of one block, O1234N1G92X-;, is now complete.

10 With the operation in (9), N2 is registered in memory by the automatic sequence number insertion function.

11 Position the tool at P1 with the step feed.

12 Enter address G and then 00. Press <INSRT>. Enter address X and then press <EOB> to register the second block, N2G00X_;, in memory.

13 N3 is registered in memory.

14 Position the tool at P2.

15 Enter address G and then 01. Press <INSRT>. Then enter address X and press <INSRT>. Enter address F and then 300<EOB> to register the third block, N3G01X-F300;, in memory.

16 N4 is registered in memory. Enter address M and then 02. Press <EOB> to register the last block, N4M02;, in memory. This completes the registration of the sample program.

The contents of memory can be checked in TEACH IN mode according to the same procedure as in EDIT mode.

Explanations

- **Checking contents of the memory**

The contents of memory can be checked in the **TEACH IN** mode by using the same procedure as in **EDIT** mode.

```
PROGRAM                                     O1234  N00004

   (RELATIVE) (WORK)
   X      –6.975X       3.025
   Y      23.723Y      23.723
   Z     –10.325Z      –0.325

   O1234  ;
   N1 G92 X10000 Y0 Z10000  ;
   N2 G00 G90 X3025 Y23723  ;
   N3 G01 Z–325 F300  ;
   N4 M02 ;
   %

 >_
 TJOG      ****   ***   ***
[PRGRM] [ LIB ] [ ] [ ] [(OPRT)]
```

- **Registering a position with compensation**

When a value is keyed in after keying in address X , Y , or Z , then the INSERT key is pressed, the value keyed in for a machine position is added for registration. This operation is useful to correct a machine position by key–in operation.

- **Registering commands other than position commands**

Commands to be entered before and after a machine position must be entered before and after the machine position is registered, by using the same operation as program editing in **EDIT** mode.

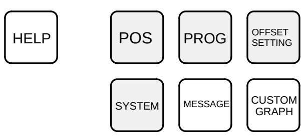
11 SETTING AND DISPLAYING DATA

General

To operate a machine, various data must be set through the CRT/MDI panel or DPL/MDI panel. The operator can monitor the state of operation with data displayed during operation.
This chapter describes how to display and set data for each function.

Explanations

·Screen transition chart for CRT/MDI



MDI function keys
(Shaded keys ( ) are described in this chapter.)

The screen transition for when each function key on the MDI panel is pressed is shown below. The subsections referenced for each screen are also shown. See the appropriate subsection for details of each screen and the setting procedure on the screen. See other chapters for screens not described in this chapter.

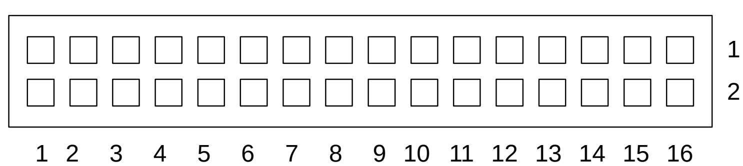
See Chapter 7 for the screen that appears when function key MESSAGE is pressed. See Chapter 12 for the screen that appears when function key HELP is pressed. In general, function key CUSTOM GRAPH is prepared by the machine tool builder and used for macros. Refer to the manual issued by the machine tool builder for the screen that appears when function key CUSTOM GRAPH is pressed.

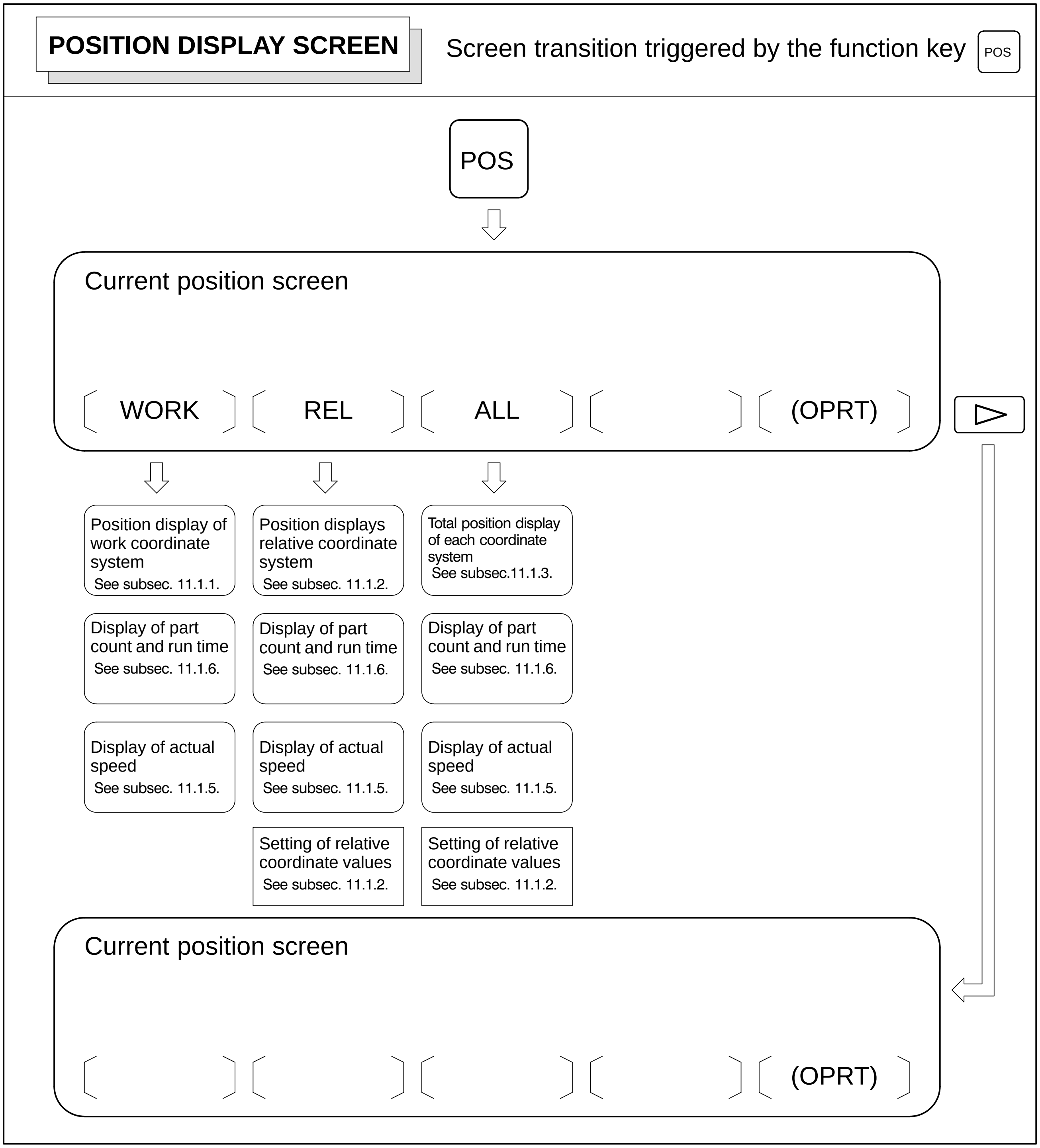
• Data protection key

The machine may have a data protection key to protect part programs, machine compensation values, setting data, and custom macro variables. Refer to the manual issued by the machine manufacture for where the data protection key is located and how to use it.

• Display of DPL/MDI

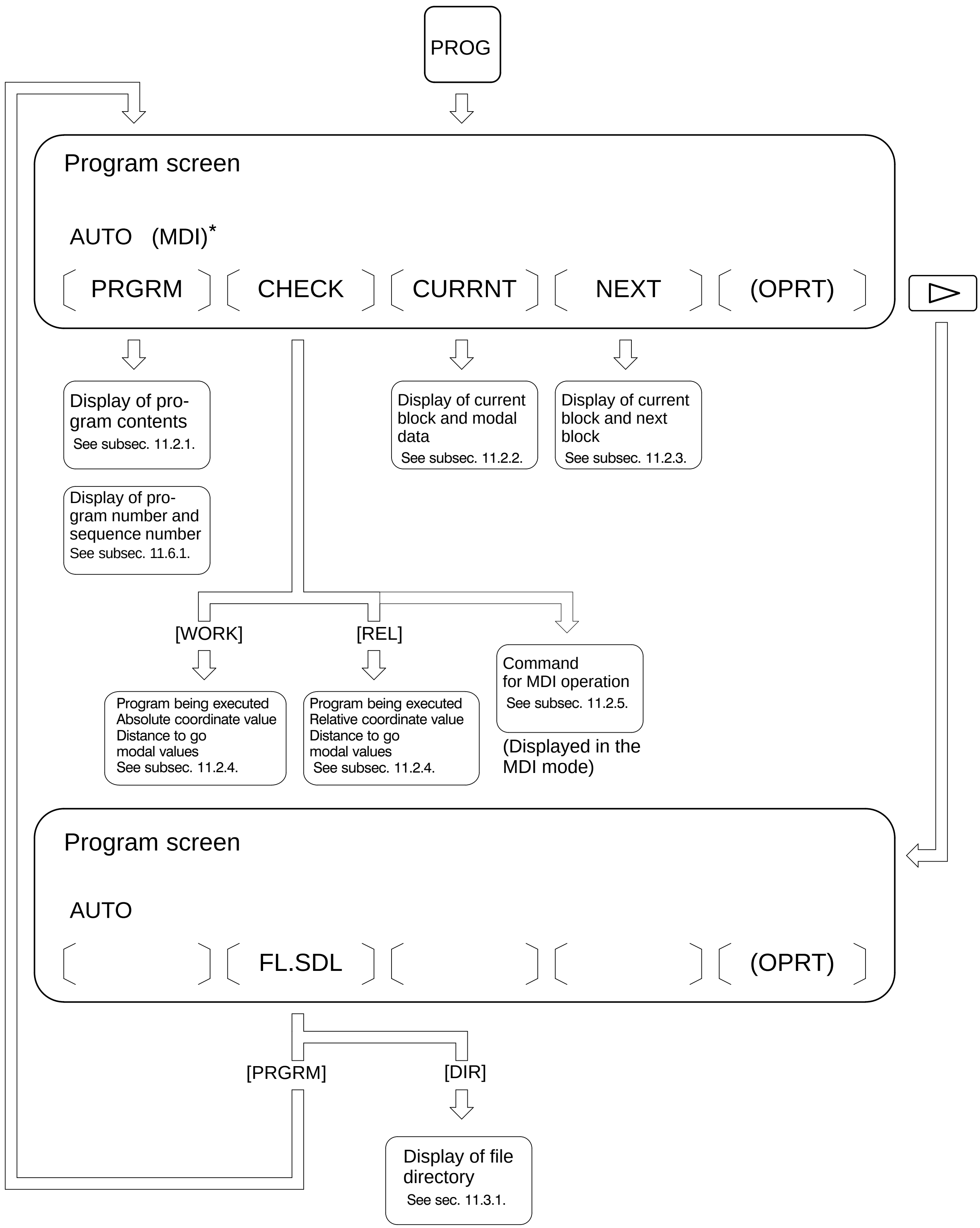
DPL/MDI displays data within 16 digits by 2 rows.




POSITION DISPLAY SCREEN
Screen transition triggered by the function key
POS
POS
Current position screen
WORK
REL
ALL
(OPRT)
Position display of work coordinate system
See subsec. 11.1.1.
Position displays relative coordinate system
See subsec. 11.1.2.
Total position display of each coordinate system
See subsec.11.1.3.
Display of part count and run time
See subsec. 11.1.6.
Display of part count and run time
See subsec. 11.1.6.
Display of part count and run time
See subsec. 11.1.6.
Display of actual speed
See subsec. 11.1.5.
Display of actual speed
See subsec. 11.1.5.
Display of actual speed
See subsec. 11.1.5.
Setting of relative coordinate values
See subsec. 11.1.2.
Setting of relative coordinate values
See subsec. 11.1.2.
Current position screen
(OPRT)

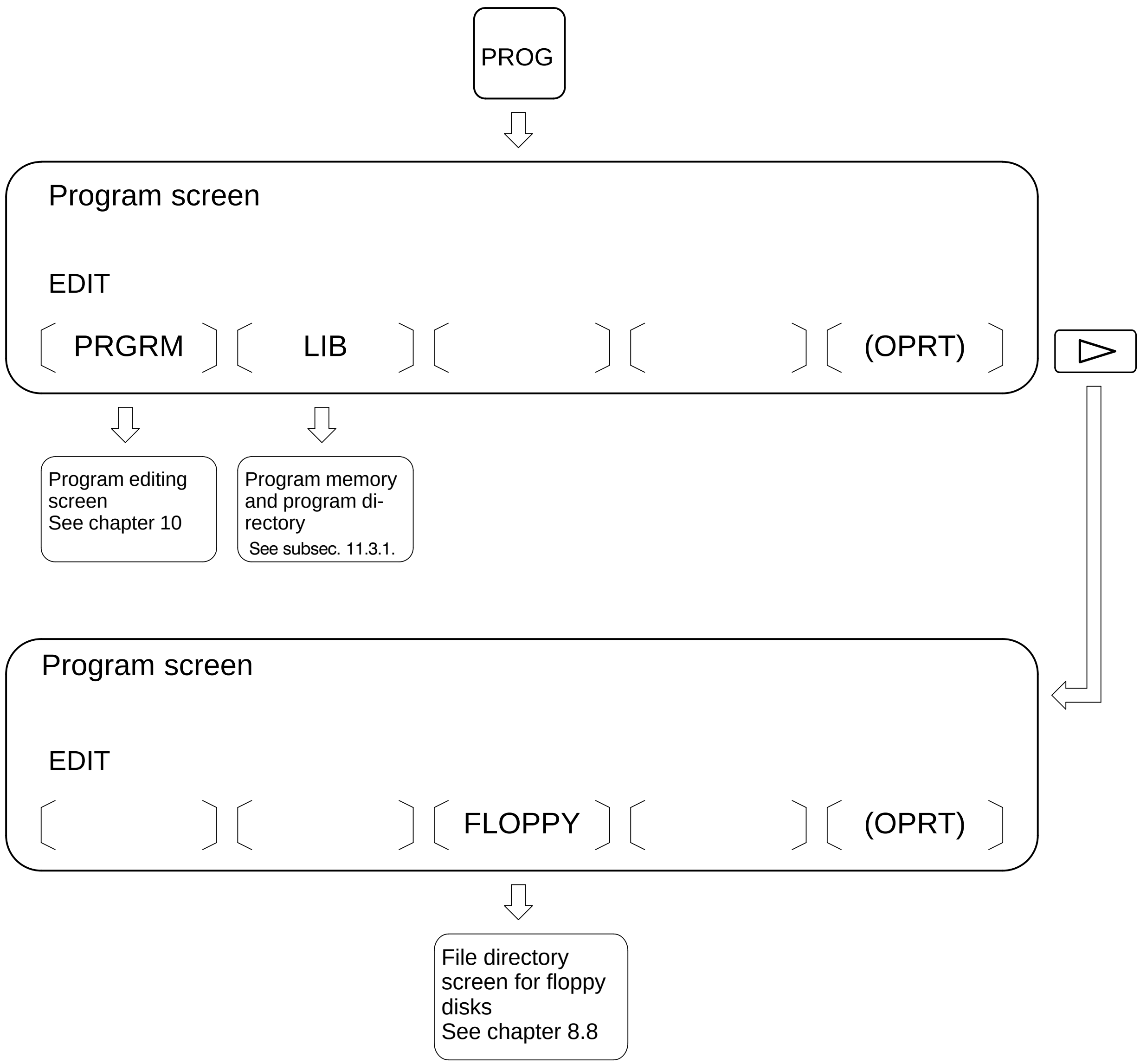
PROGRAM SCREEN

Screen transition triggered by the function key PROG in the AUTO or MDI mode

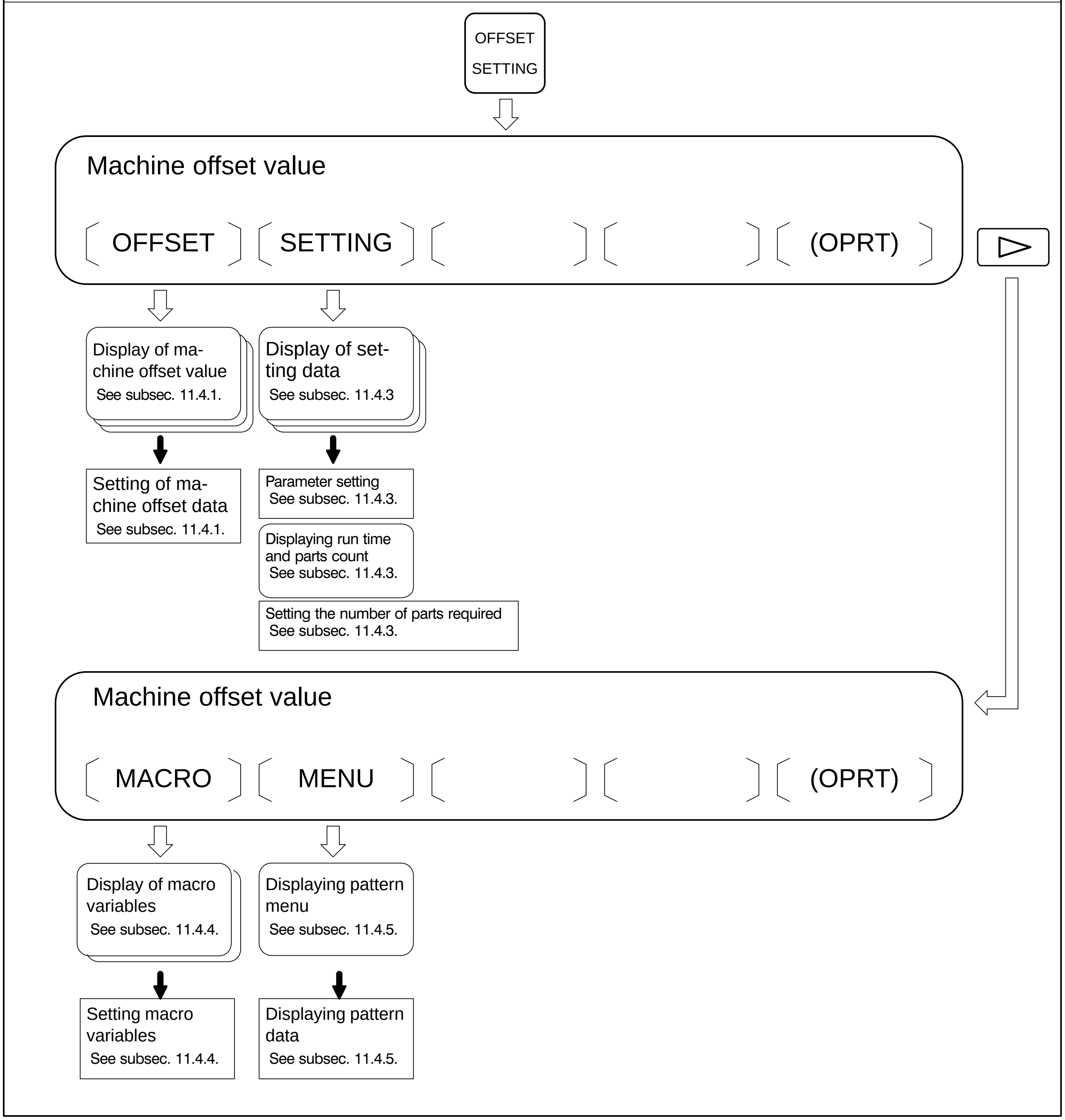


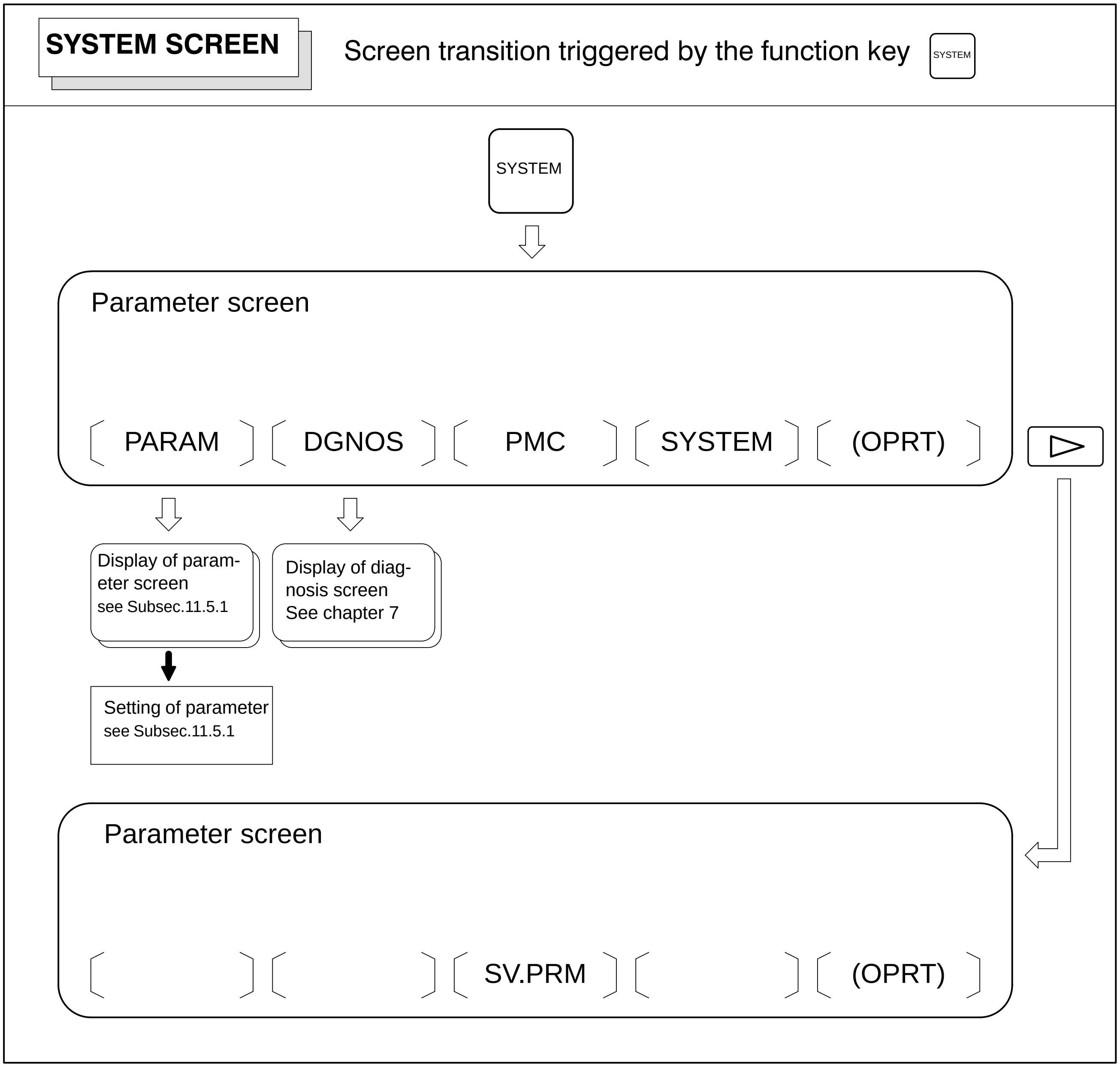
PROGRAM SCREEN

Screen transition triggered by the function key PROG in the EDIT mode



OFFSET/SETTING SCREEN Screen transition triggered by the function key OFFSET SETTING




SYSTEM SCREEN
Screen transition triggered by the function key
SYSTEM
SYSTEM
Parameter screen
PARAM
DGNOS
PMC
SYSTEM
(OPRT)
Display of param-
eter screen
see Subsec.11.5.1
Display of diag-
nosis screen
See chapter 7
Setting of parameter
see Subsec.11.5.1
Parameter screen
SV.PRM
(OPRT)

- **Setting screens**

The table below lists the data set on each screen.

Table.11. Setting screens and data on them

No.	Setting screen	Contents of setting	Reference item
1	Machine offset value	Machine length offset value	Subsec. 11.4.1
2	Setting data(handy)	Parameter write TV check Punch code Input unit (mm/inch) I/O channel Automatic insert of Sequence No.	Subsec. 11.4.2
3	Setting data (mirror image)	Mirror image	Subsec. 11.4.2
4	Setting data	Parts required	Subsec. 11.4.3
5	Macro variables	Custom macro common variables (#100 to #199) (#500 to #699)	Subsec. 11.4.4
6	Parameter	Parameter	Subsec. 11.5.1

11.1 SCREENS DISPLAYED BY FUNCTION KEY POS

Press function key POS to display the current position of the tool.

The following three screens are used to display the current position of the tool:
·Position display screen for the work coordinate system.
·Position display screen for the relative coordinate system.
·Overall position display screen.
The above screens can also display the feedrate, run time, and the number of parts.

11.1.1 Position Display in the Work Coordinate System

Displays the current position of the tool in the workpiece coordinate system. The current position changes as the tool moves. The least input increment is used as the unit for numeric values. The title at the top of the screen indicates that work coordinates are used.

Display procedure for the current position screen in the workpiece coordinate system

Procedure for CRT/MDI

1 Press function key POS .

2 Press soft key **[WORK]**.

```
ACTUAL POSITION(WORK)          O1000 N00010

   X        123.456
   Y        363.233
   Z          0.000

            PART COUNT       5
 RUN TIME    0H15M  CYCLE TIME   0H 0M38S
 ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][      ][ (OPRT) ]
```

Procedure for DPL/MDI

1 Press the <POS> key.

2 The cursor keys can be used to toggle back and forth between the WORK display and the MCHN display.

WORK X 12345. 678 Y–99999. 999	←――→ Cursor key	MCHN X 12345. 678 Y–99999. 999

Explanations

- **Display including compensation values**

Bit 6 of parameter 3104 can be used to select whether the displayed values include tool length offset.

11.1.2 Position Display in the Relative Coordinate System

Displays the current position of the tool in a relative coordinate system based on the coordinates set by the operator. The current position changes as the tool moves. The increment system is used as the unit for numeric values. The title at the top of the screen indicates that relative coordinates are used.

Display procedure for the current position screen with the relative coordinate system

Procedure for CRT/MDI

1 Press function key POS .

2 Press soft key **[REL]**.

```
ACTUAL POSITION(RELATIVE)     O1000 N00010

   X      123.456
   Y      363.233
   Z        0.000

              PART COUNT       5
  RUN TIME     0H15M  CYCLE TIME   0H 0M38S
  ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][      ][ (OPRT) ]
```

See Explanations for the procedure for setting the coordinates.

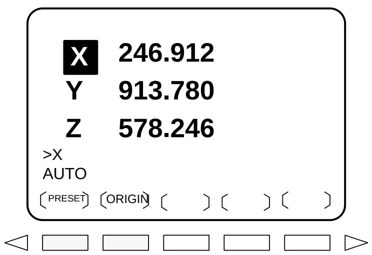
Explanations

- **Setting the relative coordinates**

The current position of the tool in the relative coordinate system can be reset to 0 or preset to a specified value as follows:

Procedure to set the axis coordinate to a specified value

Procedure for CRT/MDI



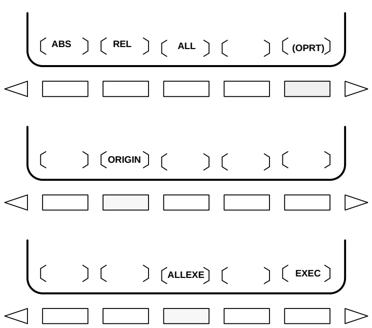
1 Enter an axis address (such as X or Y) on the screen for the relative coordinates. The indication for the specified axis blinks and the soft keys change as shown on the left.

2 · To reset the coordinate to 0, press soft key **[ORGIN]**. The relative coordinate for the blinking axis is reset to 0.

· To preset the coordinate to a specified value, enter the value and press soft key **[PRESET]**. The relative coordinate for the blinking axis is set to the entered value.

Procedure to reset all axes

Procedure for CRT/MDI



1 Press soft key **[(OPRT)]**.

2 Press soft key **[ORIGIN]**.

3 Press soft key **[ALLEXE]**.
The relative coordinates for all axes are reset to 0.

- **Display including compensation values**

Bit 6 of parameter 3104 can be used to select whether the displayed values include tool length offset.

- **Presetting by setting a coordinate system**

Bit 3 of parameter 3104 is used to specify whether the displayed positions in the relative coordinate system are preset to the same values as in the workpiece coordinate system when a coordinate system is set by a G92 command or when the manual reference position return is made.

11.1.3 Overall Position Display

Displays the following positions on a screen : Current positions of the tool in the workpiece coordinate system, relative coordinate system, and machine coordinate system, and the remaining distance. The relative coordinates can also be set on this screen. See subsection 11.1.2 for the procedure.

Procedure for displaying overall position display screen

Procedure for CRT/MDI

1 Press function key POS .

2 Press soft key **[ALL]**.

```
ACTUAL POSITION                        O1000 N00010
  (RELATIVE)                  (WORK)
  X  246.912                  X  123.456
  Y  913.780                  Y  456.890
  Z 1578.246                  Z  789.123

  (MACHINE)                   (DISTANCE TO
  X    0.000                  GO)
  Y    0.000                  X    0.000
  Z    0.000                  Y    0.000
                              Z    0.000
                             PART COUNT        5
  RUN TIME    0H15M  CYCLE TIME   0H 0M38S
  ACT.F     3000 MM/M
AUTO **** *** ***
[ WORK ][  REL  ][  ALL  ][       ][ (OPRT) ]
```

Explanations

• Coordinate display

The current positions of the tool in the following coordinate systems are displayed at the same time:

· Current position in the relative coordinate system (relative coordinate)

· Current position in the work coordinate system (work coordinate)

· Current position in the machine coordinate system (machine coordinate)

· Distance to go (distance to go)

• Distance to go

The distance remaining is displayed in the AUTO or MDI mode. The distance the tool is yet to be moved in the current block is displayed.

• Machine coordinate system

The least command increment is used as the unit for values displayed in the machine coordinate system. However, the least input increment can be used by setting bit 0 (MCN) of parameter 3104.

• Resetting relative coordinates

On the overall position display screen, relative coordinates can be reset to 0 or preset to specified values. Use the same procedure used to reset relative coordinates (Section 11.1.2).

11.1.4 Actual Feedrate Display

The actual feedrate on the machine (per minute) can be displayed on a current position display screen or program check screen by setting bit 0 (DPF) of parameter 3105.

Display procedure for the actual feedrate on the current position display screen

Procedure for CRT/MDI

1 Press function key POS to display a current position display screen.

```
ACTUAL POSITION(WORK)          O1000 N00010

   X       123.456
   Y       363.233
   Z         0.000

                         PART COUNT       5
 RUN TIME    0H15M   CYCLE TIME   0H 0M38S
 ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][      ][ (OPRT) ]
```

Actual feedrate is displayed after ACT.F.

Explanations

The actual feedrate is displayed in units of millimeter/min or inch/min (depending on the specified least input increment) under the display of the current position.

- **Actual feedrate value**

The actual rate is calculated by the following expression:

$$Fact = \sqrt{\sum_{i=1}^{n} (fi)^2}$$

where

n : Number of axes

fi : Cutting feed rate in the tangential direction of each axis or rapid traverse rate

Fact : Actual feedrate displayed

The display unit:

mm/min (metric input).

inch/min (Inch input, Two digits below the decimal point are displayed.)

The feedrate along the PMC axis can be omitted by setting bit 1 (PCF) of parameter 3105.

- **Actual feedrate display of feed per revolution**

In the case of feed per revolution, the actual feedrate displayed is the feed per minute rather than feed per revolution.

- **Actual feedrate display of rotary axis**

In the case of movement of rotary axis, the speed should be displayed in units of deg/min but is displayed on the screen in units of input system at that time. For example, when the rotary axis moves at 50 deg/min with inch system machine, the following is displayed: 0.50 INCH/M

- **Actual feedrate display on the other screen**

The program check screen also displays the actual feedrate.

11.1.5 Display of Run Time and Parts Count

The run time, cycle time, and the number of machined parts are displayed on the current position display screens.

Procedure for displaying run time and parts count on the current position display screen

Procedure for CRT/MDI

1 Press function key POS to display a current position display screen.

```
ACTUAL POSITION(RELATIVE)     O1000 N00010

  X      123.456
  Y      363.233
  Z        0.000

                    PART COUNT       5
 RUN TIME    0H15M  CYCLE TIME   0H 0M38S
 ACT.F     3000 MM/M

AUTO STRT MTN ***
[ WORK ][  REL  ][  ALL  ][       ][(OPRT)]
```

The number of machined parts (PART COUNT), run time (RUN TIME), and cycle time (CYCLE TIME) are displayed under the current position.

Explanations

- **PART COUNT**

Indicates the number of machined parts. The number is incremented each time M02, M30, or an M code specified by parameter 6710 is executed.

- **RUN TIME**

Indicates the total run time during automatic operation, excluding the stop and feed hold time.

- **CYCLE TIME**

Indicates the run time of one automatic operation, excluding the stop and feed hold time. This is automatically preset to 0 when a cycle start is performed at reset state. It is preset to 0 even when power is removed.

- **Display on the other screen**

Details of the run time and the number of machined parts are displayed on the setting screen. See subsection 11.4.3.

- **Parameter setting**

The number of machined parts and run time cannot be set on current position display screens. They can be set by parameters 6711, 6751, and 6752 or on the setting screen.

- **Incrementing the number of machined parts**

Bit 0 (PCM) of parameter 6700 is used to specify whether the number of machined parts is incremented each time M02, M30, or an M code specified by parameter 6710 is executed, or only each time an M code specified by parameter 6710 is executed.

11.2 SCREENS DISPLAYED BY FUNCTION KEY PROG (IN AUTO MODE OR MDI MODE)

This section describes the screens displayed by pressing function key PROG in AUTO or MDI mode.The first four of the following screens display the execution state for the program currently being executed in AUTO or MDI mode and the last screen displays the command values for MDI operation in the MDI mode:

1. Program contents display screen
2. Current block display screen
3. Next block display screen
4. Program check screen
5. Program screen for MDI operation

11.2.1 Program Contents Display

Displays the program currently being executed in AUTO or MDI mode.

Procedure for displaying the program contents

Procedure for CRT/MDI

1 Press function key PROG to display a program display screen.

2 Press chapter selection soft key **[PRGRM]**.
The cursor is positioned at the block currently being executed.

```
PROGRAM                          O2000 N00130
O2000 ;
N100 G92 X0 Y0 Z70. ;
N110 G91 G00 Y–70. ;
N120 Z–70. ;
N130 X20.0 Y–20.0;
N140 X–17.5 Y17.5 ;
N150 G01 X–25. ;
N160 X27.5 Y27.5 ;
N170 G01 X20. ;
N180 X45. Y45.  ;


> _
AUTO STRT    ***
[ PRGRM ][ CHECK ][CURRNT][ NEXT ][ (OPRT) ]
```

Procedure for DPL/MDI

1 Press the <PRGRM> key.
Program screen.

```
<O0001>N010G90
  G01  G43  X10  ;
```

Macro programs can only be displayed (i.e. cannot be edited).

The place on the editing now is shown by < >.

The place on execution now is shown by >.

11.2.2 Current Block Display Screen

Displays the block currently being executed and modal data in the AUTO or MDI mode.

Procedure for displaying the current block display screen

1 Press function key PROG .

2 Press chapter selection soft key **[CURRNT]**.
The block currently being executed and modal data are displayed.

```
 PROGRAM                          O2000 N00130

        (CURRNT)          (MODAL)

    G01  X   17.500      G01   F
    2000
    G17  F     2000      G17
         H        2      G91
                         G22
                         G94
                         G21  H  2



>_
AUTO STRT    ***
[ PRGRM ][ CHECK ][ CURRNT ][ NEXT ][ (OPRT) ]
```

11.2.3 Next Block Display Screen

Displays the block currently being executed and the block to be executed next in the AUTO or MDI mode.

Procedure for displaying the next block display screen

1 Press function key PROG .

2 Press chapter selection soft key **[NEXT]**.
The block currently being executed and the block to be executed next are displayed.

```
PROGRAM                            O2000 N00130

         (CURRNT)            (NEXT)
    G01  X   17.500   G90  X    -17.500
    G17  F     2000   G00
    G43  H        2



> _
AUTO STRT     ***
[ PRGRM ][ CHECK ][ CURRNT ][ NEXT ][ (OPRT) ]
```

11.2.4 Program Check Screen

Displays the program currently being executed, current position of the tool, and modal data in the AUTO mode.

Procedure for displaying the program check screen

1 Press function key PROG .

2 Press chapter selection soft key **[CHECK]**.
The program currently being executed, current position of the tool, and modal data are displayed.

```
  PROGRAM                            O2000 N00130

   O001
   G92 G90 X100. Y200. Z50. ;
   G00 X0 Y0 Z0 ;
   G01 Z250. F1000 ;
   (WORK)    (DIST TO GO)  G00  G94
   X    0.000 X    0.000   G17  G21
   Y    0.000 Y    0.000   G90
   Z    0.000 Z    0.000   G22  G49

                           H    M


     F
 > _
 AUTO STRT    ***
 [ PRGRM ][ CHECK ][ CURRNT ][ NEXT ][ (OPRT) ]
```

Explanations

• Program display

For the program currently being executed, the block currently being executed is displayed first.

• Current position display

The position in the workpiece coordinate system or relative coordinate system and the remaining distance are displayed. The absolute positions and relative positions are switched by soft keys **[WORK]** and **[REL]**.

11.2.5 Program Screen for MDI Operation

Displays the program input from the MDI and modal data in the MDI mode.

Procedure for displaying the program screen for MDI operation

Procedure

1 Press function key PROG .

2 Press chapter selection soft key **[MDI]**.
The program input from the MDI and modal data are displayed.

Program

Modal information

```
   PROGRAM  (MDI)                 O2000 N00130

   O0000 G00 X100.0 Z200.0 ;
   M03 ;
   G01 Z120.0 F500 ;
   M98 P9010 ;
   G00 Z0.0 ;
   ■

   G00  G90  G94
   G17  G22  G21  G49
                            H    M

         F
 > _
 MDI **** *** ***
 [ PRGRM ][ MDI ][ CURRNT ][ NEXT ][ (OPRT) ]
```

Explanations

- **MDI operation**

See Section 4.2 for MDI operation.

- **Modal information**

The modal data is displayed when bit 7 (MDL) of parameter 3107 is set to 1.

11.3 SCREENS DISPLAYED BY PROG FUNCTION KEY (IN THE EDIT MODE)

This section describes the screens displayed by pressing function key PROG in the EDIT mode. Function key PROG in the EDIT mode can display the program editing screen and the library screen (displays memory used and a list of programs). Pressing function key PROG in the EDIT mode can also display the floppy file directory screen. See Chapter 9 for the program editing screen. See Chapter 8 for the floppy file directory screen.

11.3.1 Displaying Memory Used and a List of Programs

Displays the number of registered programs, memory used, and a list of registered programs.

Procedure for displaying memory used and a list of programs

1 Select the **EDIT** mode.

2 Press function key PROG .

3 Press chapter selection soft key **[LIB]**.

```
  PROGRAM                          O2000 N00130

     SYSTEM EDITION    8880 – 01
   PROGRAM NO.  USED :    11 FREE :    52
   MEMORY AREA  USED :  1200 FREE :  4320
 PROGRAM LIBRARY LIST
   O0010 O0001 O0003 O0002 O0555 O0999
   O0062 O0004 O0005 O1111 O0969 O6666
   O0021 O1234 O0588 O0020 O0040




 > _
 EDIT **** *** ***
 [ PRGRM ][  LIB  ][      ][      ][ (OPRT) ]
```

Explanations

• Details of memory used

PROGRAM NO. USED

PROGRAM NO. USED : The number of the programs registered (including the subprograms)

FREE : The number of programs which can be registered additionally.

MEMORY AREA USED

MEMORY AREA USED : The capacity of the program memory in which data is registered (indicated by the number of characters).

FREE : The capacity of the program memory which can be used additionally (indicated by the number of characters).

• Program library list

Program Nos. registered are indicated.
Also, the program name can be displayed in the program table by setting parameter NAM (No. 3107#0) to 1.

```
PROGRAM                         O2000 N00130
    SYSTEM EDITION    8880 – 02
  PROGRAM NO.  USED :    11 FREE :    52
  MEMORY AREA  USED :  1200 FREE :  4320
PROGRAM LIBRARY LIST
  O0001 (MACRO–GCODE.MAIN)
  O0002 (MACRO–GCODE.SUB1)
  O0010 (TEST–PROGRAM.ARTHMETIC NO.1)
  O0020 (TEST–PROGRAM.F10–MACRO)
  O0040 (TEST–PROGRAM.OFFSET)
  O0050
  O0100 (INCH/MM CONVERT CHECK NO.1)
  O0200 (MACRO–MCODE.MAIN)
> _
EDIT **** *** ***
[ PRGRM ][  LIB  ][      ][      ][ (OPRT) ]
```

• Program name

Always enter a program name between the control out and control in codes immediately after the program number.
Up to 31 characters can be used for naming a program within the parentheses. If 31 characters are exceeded, the exceeded characters are not displayed.
Only program number is displayed for the program without any program name.

○ □□□□ (ΔΔΔΔ…Δ) ;

Program number   Program name (up to 31 characters)

• Software series

Software series of the system is displayed.
It is used for maintenance ; user is not required this information.

• Order in which programs are displayed in the program library list

Programs are displayed in the same order that they are registered in the program library list. However, if bit 4 (SOR) of parameter 3107 is set to 1, programs are displayed in the order of program number starting from the smallest one.

- **Order in which programs are registered**

Immediately after all programs are cleared (by turning on the power while pressing the DELETE key), each program is registered after the last program in the list.
If some programs in the list were deleted, then a new program is registered, the new program is inserted in the empty location in the list created by the deleted programs.

Example) When bit 4 (SOR) of parameter 3107 is 0

1. **After clearing all programs, register programs O0001, O0002, O0003, O0004, and O0005 in this order. The program library list displays the programs in the following order:**
   O0001, O0002, O0003, O0004, O0005

2. **Delete O0002 and O0004. The program library list displays the programs in the following order:**
   O0001, O0003, O0005

3. **Register O0009. The program library list displays the programs in the following order:**
   O0001, O0009, O0003, O0005

11.4 SCREENS DISPLAYED BY FUNCTION KEY OFFSET SETTING

Press function key OFFSET SETTING to display or set tool compensation values and other data.

This section describes how to display or set the following data:

1. Machine offset value
2. Settings
3. Run time and part count
4. Custom macro common variables
5. Pattern menu and pattern data

This section also describes measurement of tool length.

The pattern menu and pattern data, depend on the specifications of the machine tool builder. See the manual issued by the machine tool builder for details.

11.4.1 Setting and Displaying the Machine Offset Value

Machine length offset values are specified by H codes in a program. Compensation values corresponding to H codes are displayed or set on the screen.

Procedure for setting and displaying the machine compensation value

Procedure for CRT/MDI

1 Press function key OFFSET SETTING .

2 Press chapter selection soft key **[OFFSET]** or press OFFSET SETTING several times until the machine compensation screen is displayed.

```
OFFSET                                  O0001 N00000
   NO.       DATA        NO.       DATA
   001       1.000         009          0.000
   002      -2.000         010         -7.500
   003       0.000         011         12.000
   004       5.000         012        -20.000
   005       0.000         013          0.000
   006       0.000         014          0.000
   007       0.000         015          0.000
   008       0.000         016          0.000
ACTUAL POSITION (RELATIVE)
      X     0.000          Y            0.000
> _   Z     0.000
MDI **** *** ***
[ OFFSET ][ SETING ][ WORK ][     ][ (OPRT) ]
```

3 Move the cursor to the compensation value to be set or changed using page keys and cursor keys, or enter the compensation number for the compensation value to be set or changed and press soft key **[NO.SRH]**.

4 To set a compensation value, enter a value and press soft key **[INPUT]**. To change the compensation value, enter a value to add to the current value (a negative value to reduce the current value) and press soft key **[+INPUT]**. Or, enter a new value and press soft key **[INPUT]**.

Procedure for DPL/MDI

1 Press the <VAR> key to display the offset screen.

2 Use the cursor keys or enter <No.><(number key)><INPUT> to display the offset number to be set.

```
> H0001    0.000
  H0002    0.000
```

3 Enter a value using the data input keys.

4 Press the <INPUT> key. The offset value is input and displayed.

Explanations

- **Decimal point input**

A decimal point can be used when entering a compensation value.

11.4.2 Displaying and Entering Setting Data

Data such as the TV check flag and punch code is set on the setting data screen. On this screen, the operator can also enable/disable parameter writing, and enable/disable the automatic insertion of sequence numbers in program editing.
See Chapter 9 for automatic insertion of sequence numbers.
This subsection describes how to set data.

Procedure for setting the setting data

Procedure for CRT/MDI

1 Select the **MDI** mode.

2 Press function key OFFSET SETTING .

3 Press soft key **[SETING]** to display the setting data screen.
This screen consists of several pages.
Press page key PAGE ↑ or PAGE ↓ until the desired screen is displayed.
An example of the setting data screen is shown below.

```
SETTING (HANDY)                  O0001 N00000

  PARAMETER WRITE =1 (0:DISABLE 1:ENABLE)
  TV CHECK        =0 (0:OFF   1:ON)
  PUNCH CODE      =1 (0:EIA   1:ISO)
  INPUT UNIT      =0 (0:MM    1:INCH)
  I/O CHANNEL     =0 (0–1:CHANNEL NO.)
  SEQUENCE NO.    =0 (0:OFF   1:ON)

> _
MDI **** *** ***
[ OFFSET ][ SETING ][ WORK ][      ][ (OPRT) ]
```

```
SETTING (HANDY)                  O0001 N00000

  MIRROR IMAGE X= 0 (0:OFF   1:ON)
  MIRROR IMAGE Y= 0 (0:OFF   1:ON)
  MIRROR IMAGE Z= 0 (0:OFF   1:ON)

> _
MDI **** *** ***
[ OFFSET ][ SETING ][ WORK ][      ][ (OPRT) ]
```

4 Move the cursor to the item to be changed by pressing cursor keys [↑], [↓], [←], or [→].

5 Enter a new value and press soft key **[INPUT]**.

Procedure for DPL/MDI

1 Press the <VAR> key to display the settings screen.

```
> TVON=0
  ISO=1
```

2 Use the cursor keys to move the cursor to the item to be changed.

3 Enter either ”0” or ”1”, according to the explanation below.

4 Press <INPUT> key. Each parameter is set and displayed.

Contents of settings

- **PARAMETER WRITE (PWE)**

Setting whether parameter writing is enabled or disabled.
0 : Disabled
1 : Enabled

- **TV CHECK (TVON)**

Setting to perform TV check.
0 : No TV check
1 : Perform TV check

- **PUNCH CODE (ISO)**

Setting code when data is output through reader puncher interface.
0 : EIA code output
1 : ISO code output

- **INPUT UNIT (INCH)**

Setting a program input unit, inch or metric system
0 : Metric
1 : Inch

- **I/O CHANNEL (I/O)**

Using channel of reader/puncher interface.
0 : Channel 0
1 : Channel 1

- **SEQUENCE NO. (SEQ)**

Setting of whether to perform automatic insertion of the sequence number or not at program edit in the EDIT mode.
0 : Does not perform automatic sequence number insertion.
1 : Perform automatic sequence number insertion.

- **MIRROR IMAGE**

Setting of mirror image ON/OFF for each axes. Cannot be set with DPL/MDI.
0 : Mirror image off
1 : Mirror image on

- **Writing PMC data (DWE)**

Specifies whether PMC data can be written from the DPL/MDI.
0 : PMC data cannot be written from the DPL/MDI.
1 : PMC data can be written from the DPL/MDI.

11.4.3 Displaying and Setting Run Time and Parts Count

Various run times, the total number of machined parts, number of parts required, and number of machined parts can be displayed. This data can be set by parameters or on this screen (the total number of machined parts can be set only by parameters).

Procedure for Displaying and Setting Run Time and Parts Count

Procedure for CRT/MDI

1 Select the MDI mode.

2 Press function key OFFSET SETTING .

3 Press chapter selection soft key **[SETING]**.

4 Press page key ↑ PAGE or PAGE ↓ several times until the following screen is displayed.

```
SETTING (TIMER)                    O0001 N0000

   PARTS TOTAL     =    14
   PARTS REQUIRED  =     0
   PARTS COUNT     =    23
   POWER ON        =    4H 31M
   OPERATING TIME  =    0H  0M  0S
   CUTTING TIME    =    0H 37M  5S
   FREE PURPOSE    =    0H  0M  0S
   CYCLE TIME      =    0H  0M  0S



>_
MDI **** *** ***
[ OFFSET ][ SETING ][ WORK ][     ][ (OPRT) ]
```

5 To set the number of parts required, move the cursor to PARTS REQUIRED and enter the number of parts to be machined.

Display items

- **PARTS TOTAL**

This value is incremented by one when M02, M30, or an M code specified by parameter 6710 is executed. This value cannot be set on this screen. Set the value in parameter 6712.

- **PARTS REQUIRED**

It is used for setting the number of machined parts required.
When the “0” is set to it, there is no limitation to the number of parts. Also, its setting can be made by the parameter (NO. 6713).

- **PARTS COUNT**

This value is incremented by one when M02, M30, or an M code specified by parameter 6710 is executed. The value can also be set by parameter 6711. In general, this value is reset when it reaches the number of parts required. Refer to the manual issued by the machine tool builder for details.

- **POWER ON**

Displays the total time which the power is on. This value cannot be set on this screen but can be preset in parameter 6750.

- **OPERATING TIME**

Indicates the total run time during automatic operation, excluding the stop and feed hold time.
This value can be preset in parameter 6751 or 6752.

- **CUTTING TIME**

Displays the total time taken by cutting that involves cutting feed such as linear interpolation (G01). This value can be preset in parameter 6753 or 6754.

- **FREE PURPOSE**

This value can be used, for example, as the total time during which coolant flows. Refer to the manual issued by the machine tool builder for details.

- **CYCLE TIME**

Indicates the run time of one automatic operation, excluding the stop and feed hold time. This is automatically preset to 0 when a cycle start is performed at reset state. It is preset to 0 even when power is removed.

Explanations

- **Usage**

When the command of M02 or M30 is executed, the total number of machined parts and the number of machined parts are incremented by one. Therefore, create the program so that M02 or M30 is executed every time the processing of one part is completed. Furthermore, if an M code set to the parameter (NO. 6710) is executed, counting is made in the similar manner. Also, it is possible to disable counting even if M02 or M30 is executed (parameter PCM (No. 6700#0) is set to 1). For details, see the manual issued by machine tool builders.

Restrictions

- **Run time and part count settings**

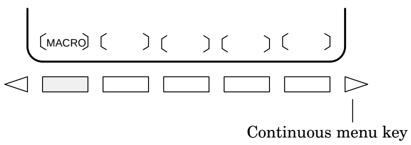
Negative value cannot be set.

11.4.4 Displaying and Setting Custom Macro Common Variables

Displays common variables (#100 to #199, and #500 to #699) on the CRT. When the absolute value for a common variable exceeds 99999999, ******** is displayed. The values for variables can be set on this screen. Relative coordinates can also be set to valiables.

Procedure for displaying and setting custom macro common variables

Procedure for CRT/MDI



1 Press function key OFFSET SETTING .

2 Press the continuous menu key ▷ , then press chapter selection soft key **[MACRO]**.The following screen is displayed:

```
VAR.:                                O0000 N00000
  NO.        NAME         DATA      COMMENT
  100                 01000.000
  101                -50000.000
  102                -20000.000
  103                 01000.000
  104                 1000000.0
  105                     0.000
  106                     0.000
  107                     0.000
 ACTUAL POSITION (WORK)
     X 0.000     Y 0.000
     Z 0.000
> _
MDI **** *** ***
[ NO.SRH ][     ][ INP.C. ][     ][ INPUT ]
```

3 Move the cursor to the variable number to set using either of the following methods:
– Enter the variable number and press soft key **[NO.SRH]**.
– Move the cursor to the variable number to set by pressing page keys PAGE ↑ and/or PAGE ↓ and cursor keys ↑ , ↓ , ← , and/or → .

4 Enter data with numeric keys and press soft key **[INPUT]**.

5 To set a work coordinate in a variable, press address key X , Y , or Z , then press soft key **[INP.C.]**.

6 To set a blank in a variable, just press soft key **[INPUT]**. The value field for the variable becomes blank.

Procedure for DPL/MDI

1 Press the <VAR> key to display the custom macro variable screen.

2 Use the cursor keys or enter <No.><(number key)><INPUT> to display the variable to be set.

```
> #0100
  #0101
```

Then select the following operation according to the type of data.

- **Setting macro variables from the DPL/MDI panel**

3 Enter a value using the data input keys.

4 Press the <INPUT> key. The macro variable is input and displayed.

- **Setting coordinates in macro variables**

3 Press the <X> key. The work coordinates will be displayed on the input line. When the two-axis is in effect, press the <X> key a second time to set the coordinates for the second axis. The work coordinates for the second axis will be displayed on the input line.

4 Press the <INPUT> key. The coordinates will be stored in the variables and displayed.

- **Erasing the contents of a macro variable**

3 Press the <.> key.

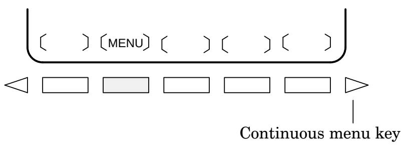
4 Press the <INPUT> key. The variable is erased and the display is made blank.

11.4.5 Displaying Pattern Data and Pattern Menu

This subsection uses an example to describe how to display or set machining menus (pattern menus) created by the machine tool builder. Refer to the manual issued by the machine tool builder for the actual pattern menus and pattern data. See PROGRAMMING for the pattern data entry function.

Procedure for displaying the pattern data and the pattern menu

Procedure for CRT/MDI



1 Press function key OFFSET SETTING .

2 Press the continuous menu key ▷ , then press chapter selection soft key **[MENU]**.
The following screen (pattern menu screen) is displayed:

```
MENU : HOLE PATTERN               O0000 N00000
       1.   BOLT HOLE
       2.   GRID
       3.   LINE ANGLE
       4.   TAPPING
       5.   DRILLING
       6.   BORING
       7.   POCKET
       8.   PECK
       9.
      10.

> _
MDI **** *** ***
[ MACRO ][ MENU ][       ][      ][ (OPRT) ]
```

3 Enter a pattern number and press soft key **[SELECT]**.
In this example, press 1 , then press **[SELECT]**.
The following screen (pattern data screen) is displayed:

```
VAR. : BOLT HOLE                   O0001 N00000
 NO.  NAME            DATA    COMMENT
 500   TOOL            0.000
 501   STANDARD X      0.000 *BOLT HOLE
 502   STANDARD Y      0.000 CIRCLE*
 503   RADIUS          0.000 SET PATTERN
 504   S. ANGL         0.000 DATA TO VAR.
 505   HOLES NO        0.000 NO.500–505.
 506                   0.000
 507                   0.000

ACTUAL POSITION (RELATIVE)
     X      0.000         Y      0.000
     Z      0.000
> _
MDI **** *** ***
[ MACRO ][ MENU ][     ][     ][ (OPRT) ]
```

4 Enter necessary pattern data and press INPUT.

5 After entering all necessary data, enter the **AUTO** mode and press the cycle start button to start machining.

Explanations

- **Explanation of the pattern menu screen**

HOLE PATTERN : Menu title
An optional character string can be displayed within 12 characters.

BOLE HOLE : Pattern name
An optional character string can be displayed within 10 characters.

The machine tool builder should program character strings of menu title and pattern name by custom macro, and load them into the program memory.

- **Explanation of the pattern data screen**

BOLT HOLE : Pattern data title
An optional character string can be displayed within 12 characters.

TOOL : Variable name
An optional character string can be displayed within 10 characters.

BOLT HOLE CIRCLE : Comment statement
An optional character string comment can be displayed up to 12 characters/line by 8 lines.

The machine tool builder should program the character strings of variable name and comment statement by custom macro, and load them into the program memory.

11.5 SCREENS DISPLAYED BY FUNCTION KEY SYSTEM

When the controller and machine are connected, parameters must be set to determine the specifications and functions of the machine in order to fully utilize the characteristics of the servo motor or other parts.
This chapter describes how to set parameters on the MDI panel. Parameters can also be set with external input/output devices such as the Handy File (see Chapter 8).
See Chapter 7 for the diagnostic screens displayed by pressing function key SYSTEM .

11.5.1 Displaying and Setting Parameters

When the controller and machine are connected, parameters are set to determine the specifications and functions of the machine in order to fully utilize the characteristics of the servo motor. The setting of parameters depends on the machine. Refer to the parameter list prepared by the machine tool builder.
Normally, the user need not change parameter setting.

Procedure for displaying and setting parameters

Procedure for CRT/MDI

1 Set 1 for **PARAMETER WRITE** to enable writing. See the procedure for enabling/disabling parameter writing described below.

2 Press function key SYSTEM .

3 Press chapter selection soft key **[PARAM]** to display the parameter screen.

```
PARAMETER (SETTING)                        O0010 N00002

   0000       SEQ                 INI  ISO   TVC
          0    0    0    0    0    0    0    0
   0001
          0    0    0    0    0    0    0    0
   0002
          0    0    0    0    0    0    0    0
   0012                                      MIR
    X     0    0    0    0    0    0    0    0
    Y     0    0    0    0    0    0    0    0
    Z     0    0    0    0    0    0    0    0

 > _
 MDI   **** *** ***
 [ PARAM ][ DGNOS ][  PMC  ][ SYSTEM ][ (OPRT) ]
```

4 Move the cursor to the parameter number to be set or displayed in either of the following ways:

- Enter the parameter number and press soft key **[NO.SRH]** .
- Move the cursor to the parameter number using the page keys, ↑ PAGE and PAGE ↓ , and cursor keys, ↑, ↓, ←, and → .

5 To set the parameter, enter a new value with numeric keys and press soft key **[INPUT]**. The parameter is set to the entered value and the value is displayed.

6 Set 0 for **PARAMETER WRITE** to disable writing.

Procedure for DPL/MDI

Press the <DGNOS/PARAM> key to toggle between the parameter screen and diagnostic screen.

Parameter screen

```
> &0001   01010101
  &0002   01010101
```

Diagnostic screen

```
> @0800    0
  @0801    0
```

Display of PMC data ⇓ ⇑ Key operation

```
> D0000     00000000
  D0001     00000000
```

Procedure for enabling/displaying parameter writing

Procedure for CRT/MDI

1 Select the **MDI** mode or enter state emergency stop.

2 Press function key OFFSET SETTING .

3 Press soft key **[SETING]** to display the setting screen.

```
SETTING (HANDY)                          O0001 N00000

  PARAMETER WRITE = 1 (0:DISABLE 1:ENABLE)
  TV CHECK        = 0 (0:OFF   1:ON)
  PUNCH CODE      = 1 (0:EIA   1:ISO)
  INPUT UNIT      = 0 (0:MM    1:INCH)
  I/O CHANNEL     = 0 (0–1:CHANNEL NO.)
  SEQUENCE NO.    = 0 (0:OFF   1:ON)





> _
MDI **** *** ***
[ OFFSET ][ SETING ][ WORK ][      ][ (OPRT) ]
```

4 Move the cursor to **PARAMETER WRITE** using cursor keys.

5 Press soft key **[(OPRT)]**, then press **[1: ON]** to enable parameter writing.
At this time, the CNC enters the P/S alarm state (No. 100).

6 After setting parameters, return to the setting screen. Move the cursor to **PARAMETER WRITE** and press soft key **[(OPRT)]** , then press **[0: OFF]**.

7 Depress the RESET key to release the alarm condition. If alarm No. 000 has occurred, however, turn off the power supply and then turn it on, otherwise the alarm is not released.

Procedure for DPL/MDI

1 Press the <VAR> key to display the settings screen.

2 Use the cursor keys to position the cursor at PWE.

3 Press the <1> key and the <INPUT> key, in that order, to enable parameters to be written. The CNC unit will generate P/S alarm 100.

4 Press the <DGNOS/PARAM> key to display the parameter screen.

```
> &0001    00000000
  &0002    00000000
```

5 Move the cursor to the number of the parameter to change.
Method 1
Use the cursor keys. The cursor will continue to move while a cursor key is being pressed.
Method 2
Press the following keys and enter data in the order shown: <No.><(parameter No.)><INPUT>

6 Enter a parameter value with the data input keys.

7 Press the <INPUT> key. The parameter value is input and displayed.

8 After all parameters have been set and confirmed, return to the settings screen and set PWE to 0.

9 Normally, in order to release the alarm state, press the <CAN> key. However, in order to release alarm No. 000, the power needs to be turned off and then on again.

Explanations

- **Setting parameters with external input/output devices**

See Chapter 8 for setting parameters with external input/output devices such as the Handy File.

- **Parameters that require turning off the power**

Some parameters are not effective until the power is turned off and on again after they are set. Setting such parameters causes alarm 000. In this case, turn off the power, then turn it on again.

- **Parameter list**

Refer to the FANUC Power Mate–MODEL H Manual (B–62005E) for the parameter list.

- **Setting data**

Some parameters can be set on the setting screen if the parameter list indicates ”Setting entry is acceptable”. Setting 1 for **PARAMETER WRITE** is not necessary when these parameters are set on the setting screen.

11.6 DISPLAYING THE PROGRAM NUMBER, SEQUENCE NUMBER, AND STATUS, AND WARNING MESSAGES FOR DATA SETTING OR INPUT/OUTPUT OPERATION

The program number, sequence number, and current controller status are always displayed on the screen except when the power is turned on, a system alarm occurs, or the PMC screen is displayed.
If data setting or the input/output operation is incorrect, the controller does not accept the operation and displays a warning message.
This section describes the display of the program number, sequence number, and status, and warning messages displayed for incorrect data setting or input/output operation.

11.6.1 Displaying the Program Number and Sequence Number

The program number and sequence number are displayed at the top right on the screen as shown below.

```
  PROGRAM                             O2000 N00130
  O2000 ;
  N100 G92 X0 Y0 Z70. ;
  N110 G91 G00 Y–70. ;
  N120 Z–70. ;
  N130 G01 X17.5 F2000 ;
  N140 G00 X–17.5 Y17.5 Z17.5 ;
  N150 G01 X–25. ;
  N160 G00 X27.5 Y27.5 Z27.5 ;
  N170 G01 X20. ;
  N180 G00 X45. Y45. Z45. ;


> _
EDIT **** *** ***
[ PRGRM ][  LIB  ][       ][       ][ (OPRT) ]
```

Sequence No.
Program No.

The program number and sequence number displayed depend on the screen and are given below:

On the program screen in the EDIT mode on Background edit screen : The program No. being edited and the sequence number just prior to the cursor are indicated.

Other than above screens :

The program No. and the sequence No. executed last are indicated.

Immediately after program number search or sequence number search :

Immediately after the program No. search and sequence No. search, the program No. and the sequence No. searched are indicated.

11.6.2 Displaying the Status and Warning for Data Setting or Input/Output Operation

The current mode, automatic operation state, alarm state, and program editing state are displayed on the next to last line on the CRT screen allowing the operator to readily understand the operation condition of the system.
If data setting or the input/output operation is incorrect, the controller does not accept the operation and a warning message is displayed on the next to last line of the CRT screen. This prevents invalid data setting and input/output errors.

Explanations

- **Description of each display (CRT/MDI)**

(8) Data is out of range.
(Note) Actually, (8) is displayed in the area starting from (2).

(5)
—EMG— (Note) Actually, 5 is displayed in the area for (3) and (4).

(1)	(2)	(3)	(4)	(6)	(7)
EDIT	STOP	MTN	FIN	ALM	INPUT

(Display soft keys)

- **(1) Current mode**

MDI : Manual data input
AUTO : Automatic operation
EDIT : Memory editing
JOG : Jog feed
TJOG : TEACH IN JOG
STEP : Manual incremental feed
ZRN : Manual reference position return

- **(2) Automatic operation status**

**** : Reset (When the power is turned on or the state in which program execution has terminated and automatic operation has terminated.)
STOP : Automatic operation stop (The state in which one block has been executed and automatic operation is stopped.)
HOLD : Feed hold (The state in which execution of one block has been interrupted and automatic operation is stopped.)
STRT : Automatic operation start–up (The state in which the system operates automatically)

- **(3) Axis moving status/dwell status**

MTN : Indicates that the axis is moving.
DWL : Indicates the dwell state.
*** : Indicates a state other than the above.

- **(4) State in which an auxiliary function is being executed**

FIN : Indicates the state in which an auxiliary function is being executed. (Waiting for the complete signal from the PMC)
*** : Indicates a state other than the above.

- **(5) Emergency stop or reset status**

--EMG-- : Indicates emergency stop. (Reversed display)
–RESET– : Indicates that the reset signal is being received.
–WAIT– : Waits for MCC to turn on if the servo alarm for MCC being turned off has been disabled.

- **(6) Alarm status**

ALM : Indicates that an alarm is issued. (Reversed display)
BAT : Indicates that the battery is low. (Reversed display)
Space : Indicates a state other than the above.

- **(7) Program editing status**

INPU T	:	Indicates that data is being input.
OUTPUT	:	Indicates that data is being output.
SRCH	:	Indicates that a search is being performed.
EDIT	:	Indicates that another editing operation is being performed (insertion, modification, etc.)
LSK	:	Indicates that labels are skipped when data is input.
Space	:	Indicates that no editing operation is being performed.

- **(8) Warning for data setting or input/output operation**

When invalid data is entered (wrong format, value out of range, etc.), when input is disabled (wrong mode, write disabled, etc.), or when input/output operation is incorrect (wrong mode, etc.), a warning message is displayed. In this case, the controller does not accept the setting or input/output operation (retry the operation according to the message).
The following are examples of warning messages:

Example 1)
When a parameter is output to an external input/output device

```
> _
AUTO    WRONG MODE

              (Display soft keys)
```

Contents of data for DPL/MDI

The program edit status and data set status are displayed.

```
<O0001>N010  G90
        READ
```

Display items:

EDIT	: Editing a program
SEARCH	: Searching
WRITE	: Outputting data
READ	: Inputting data
COMPARE	: Collating data
LSK	: Label skip status
EXECUTE	: Waiting for ladder input/output

12 HELP FUNCTION

The help function displays on the CRT screen detailed information about alarms issued in the controller and about controller operations. The following information is displayed.

- **Detailed information of alarms**

When the controller is operated incorrectly or an erroneous machining program is executed, the controller enters the alarm state. The help screen displays detailed information about the alarm that has been issued and how to reset it. The detailed information is displayed only for a limited number of P/S alarms. These alarms are often misunderstood and are rather difficult to understand.

- **Operation method**

If you are not sure about a controller operation, refer to the help screen for information about each operation.

- **Parameter table**

When setting or referring to a system parameter, if you are not sure of the number of the parameter, the help screen displays a list of parameter Nos. for each function.

Help Function Procedure

Procedure

1 Press the HELP key on the MDI panel. HELP (INITIAL MENU) screen is displayed.

```
HELP (INITIAL MENU)O1234  N00001


***** HELP *****
  1.  ALARM DETAIL
  2.  OPERATION METHOD
  3.  PARAMETER TABLE







 AUTO ****  ***  ***
[1 ALAM ][2 OPR ][3 PARA ][      ][      ]
```

Fig.12(a) HELP (INITIAL MENU) Screen

The user cannot switch the screen display from the PMC screen or CUSTOM screen to the help screen. The user can return to the normal controller screen by pressing the HELP key or another function key.

ALARM DETAIL screen

2 Press soft key **[1 ALAM]** on the HELP (INITIAL MENU) screen to display detailed information about an alarm currently being raised.

```
HELP (ALARM DETAIL)O0010   N00001

NUMBER : 027
M'SAGE  :  NO  AXES  COMMANDED  IN  G43/G44
FUNCTION  :  TOOL  LENGTH  COMPENSATION  C
ALARM  :
 IN  TOOL  LENGTH  COMPENSATION  TYPE C,
 NO  AXIS  IS  DESIGNATED  IN  G43  &  G44
 BLOCKS.  IN  TOOL  LENGTH  COMPENSATION
 TYPE  C,  IT  TRIES  TO  LATCH  ON  TO
 ANOTHER  AXIS  WITHOUT  OFFSET  CANCE–
 LING.

 > _
 AUTO  ****  ***  ***
[1 ALAM] [2 OPR] [3 PARA] [        ] [(OPRT)]
```

Alarm No.
Normal explana–tion on alarm
Function classification
Alarm details

Fig.12(b) ALARM DETAIL Screen when Alarm P/S 27 is issued

Note that only details of the alarm identified at the top of the screen are displayed on the screen.
If the alarms are all reset while the help screen is displayed, the alarm displayed on the ALARM DETAIL screen is deleted, indicating that no alarm is issued.

```
HELP  (ALARM DETAIL) O1234 N00001


NUMBER :
M'SAGE :
FUNCTION:
ALARM   :

          <<ALARM IS NOT GENERATED>>
    ENTER THE DETAIL–REQUIRED ALARM NUMBER,
    AND PRESS [SELECT] KEY


 >100
 AUTO  ****  ***  ***
[        ] [        ] [        ] [        ] [SELECT]
```

Fig.12(c) ALARM DETAIL Screen when No Alarm is issued

3 To get details on another alarm number, first enter the alarm number, then press soft key **[SELECT]**. This operation is useful for investigating alarms not currently being raised.

```
HELP (ALARM DETAIL)O1234 N00001


NUMBER : 100
M'SAGE : PARAMETER WRITE ENABLE
FUNCTION:
ALARM  :

      <<ALARM IS NOT GENERATED>>
  ENTER THE DETAIL-REQUIRED ALARM NUMBER,
  AND PRESS [SELECT] KEY.


 >100
 AUTO  ****  ***  ***
 [        ][        ][        ][        ][ SELECT ]
```

Fig.12(d) ALARM DETAIL Screen when P/S 100 is selected

OPERATION METHOD screen

4 To determine an operating procedure for the controller, press the soft key **[2 OPR]** key on the HELP (INITIAL MENU) screen. The OPERATION METHOD menu screen is then displayed. (See Fig. 12 (e).)

```
HELP (OPERATION METHOD) O1234   N00001


1. PROGRAM EDIT
2. SEARCH
3. RESET
4. DATA INPUT WITH MDI
5. DATA INPUT WITH TAPE
6. OUTPUT
7. INPUT WITH FANUC CASSETTE
8. OUTPUT WITH FANUC CASSETTE
9. MEMORY CLEAR

 AUTO ****  ***  ***
[ 1 ALAM ][ 2 OPR ][ 3 PARA ][        ][ (OPRT) ]
```

Fig.12(e) OPERATION METHOD Menu Screen

To select an operating procedure, enter an item No. from the keyboard then press the **[SELECT]** key.

```
 > 1
 AUTO ****  ***  ***
[        ][        ][        ][        ][SELECT]
```

Fig.12(f) How to select each OPERATION METHOD screen

When “1. PROGRAM EDIT” is selected, for example, the screen in Figure 12 (g) is displayed.
On each OPERATION METHOD screen, it is possible to change the displayed page by pressing the PAGE key. The current page No. is shown at the upper right corner on the screen.

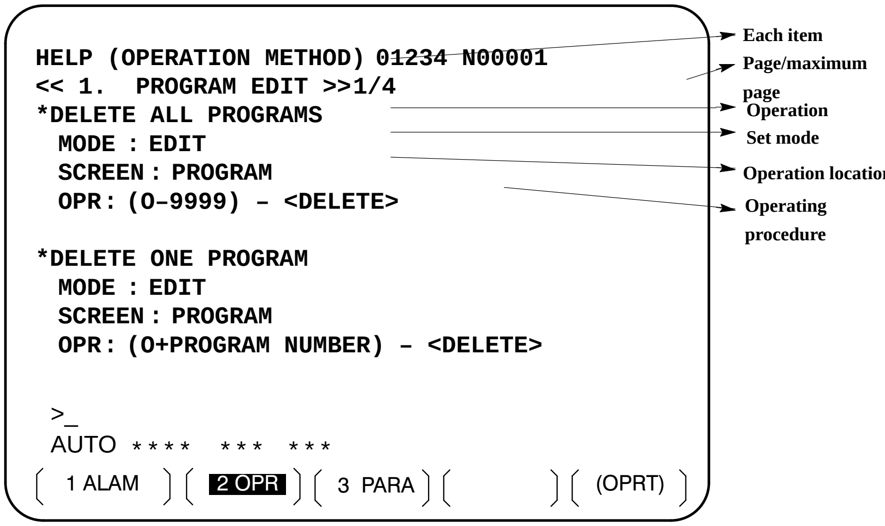
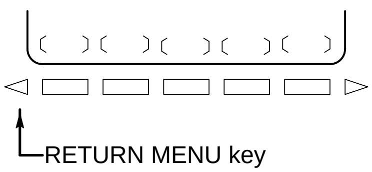


Fig.12(g) Selected OPERATION METHOD screen



5 To return to the OPERATION METHOD menu screen, press the RETURN MENU key to display “**[2 OPR]**” again, and then press the **[2 OPR]** key again.
To directly select another OPERATION METHOD screen on the screen shown in Figure 12 (g), enter an item No. from the keyboard and press the **[SELECT]** key.

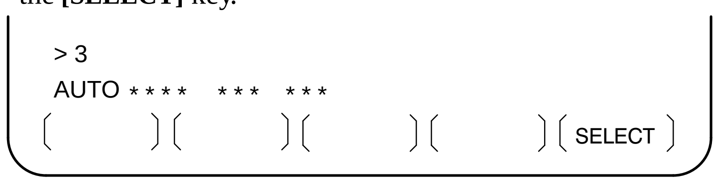


Fig.12(h) How to select another OPERATION METHOD screen

PARAMETER TABLE screen

6 If you are not sure of the No. of a system parameter to be set, or to refer to a system parameter, press the **[3 PARA]** key on the HELP (INITIAL MENU) screen. A list of parameter Nos. for each function is displayed. (See Figure 12 (i).)
It is possible to change the displayed page on the parameter screen. The current page No. is shown at the upper right corner on the screen.

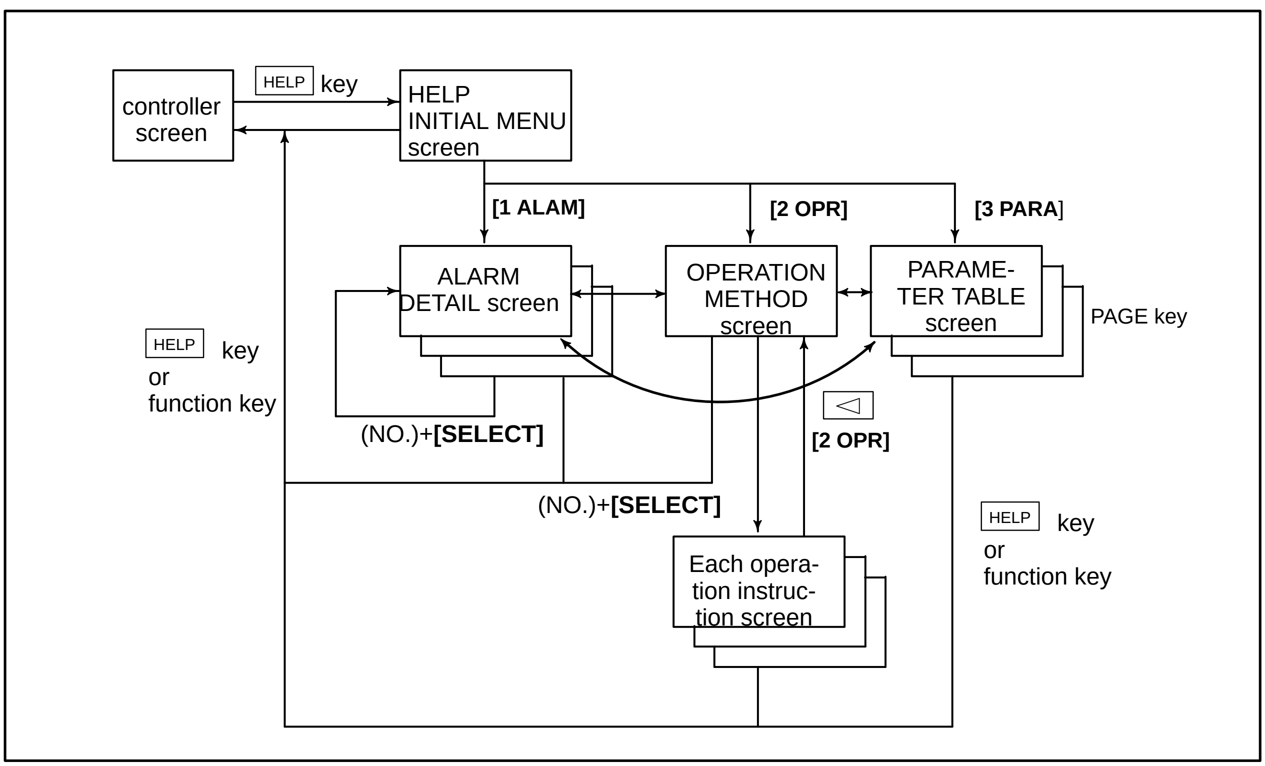
```
HELP (PARAMETER TABLE)01234  N00001
     1/4

* SETTEING   (No. 0000~)
* READER/PUNCHER INTERFACE (No. 0100~)
* AXIS CONTROL/SETTING UNIT(No. 1000~)
* COORDINATE SYSTEM(No. 1200~)
* STROKE LIMIT(No. 1300~)
* FEED RATE  (No. 1400~)
* ACCEL/DECELERATION CTRL(No. 1600~)
* SERVORELATED(No. 1800~)
* DI/DO    (No. 3000~)

  AUTO ****  ***  ***
[ 1 ALAM ][ 2 OPR ][ 3 PARA ][        ][ (OPRT) ]
```

Fig. 12(i) PARAMETER TABLE screen

7 To exit from the help screen, press the HELP key or another function key.

Explanation

● Configuration of the Help Screen



IV MAINTENANCE

1 DAILY MAINTENANCE

1.1 CLEANING OF COOLING SYSTEM

Air filters and suchlike are not used in the Power Mate itself, but heat exchangers or air filters are used in the machine side locker incorporating the Power Mate.
Clean periodically in accordance with the manuals issued by the MTB.

1.2 BATTERY REPLACEMENT

(1) Absolute pulsecoder battery
When using the absolute pulsecoder, replace the battery quickly when one of the alarms 306-308 is displayed on DPL/MDI or CRT/MDI.
Regarding the installation position of the battery case housing the battery, refer to the manuals issued by the MTB.

- Carry out the battery replacement in the ”Power ON” state.
- Be careful with the battery polarity: do not insert the wrong way around.
- Use only the specified replacement battery (lithium battery : A06B–6073–K001). Battery life is approximately one year; therefore replace regularly once a year even if the above alarm does not occur.

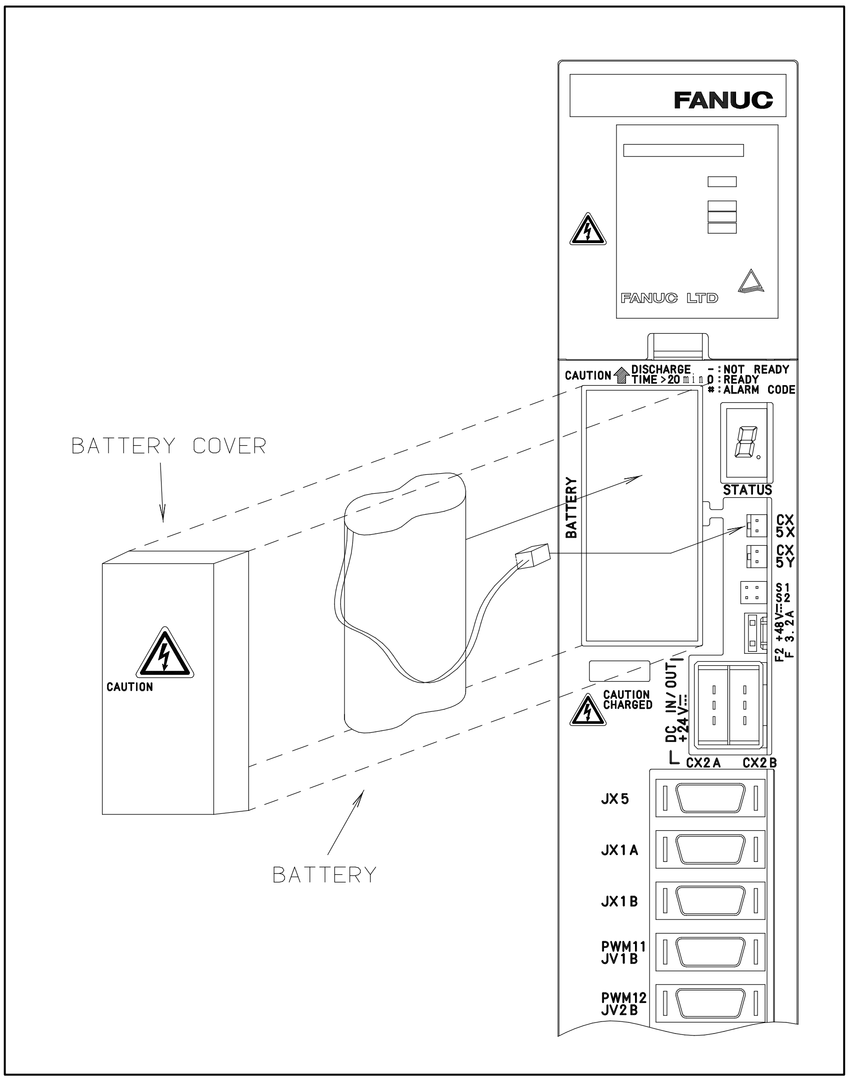


Fig 1.2 (a) Mounting location of lithium battery in α series servo amplifier

(2) Power Mate main unit battery

A lithium battery (A02B-0118-K111) to backup the nonvolatile power supply for memorizing the parameters and NC part programs in the Power Mate main unit is installed in the battery holder on the back of the front cover of the plastic case.

Replace the lithium battery quickly when the BAT alarm display appears on DPL/MDI or CRT/MDI, Refer to Fig. 1.2 when replacing.

- Carry out the lithium battery replacement in the ”Power ON” state.
- Take care not to insert the connector for the lithium battery the wrong way around.
- For the replacement battery, use the specified product (lithium battery : A02B-0118-K111). Battery life is approximately one year; therefore replace regularly once a year even if the BAT alarm does not occur.

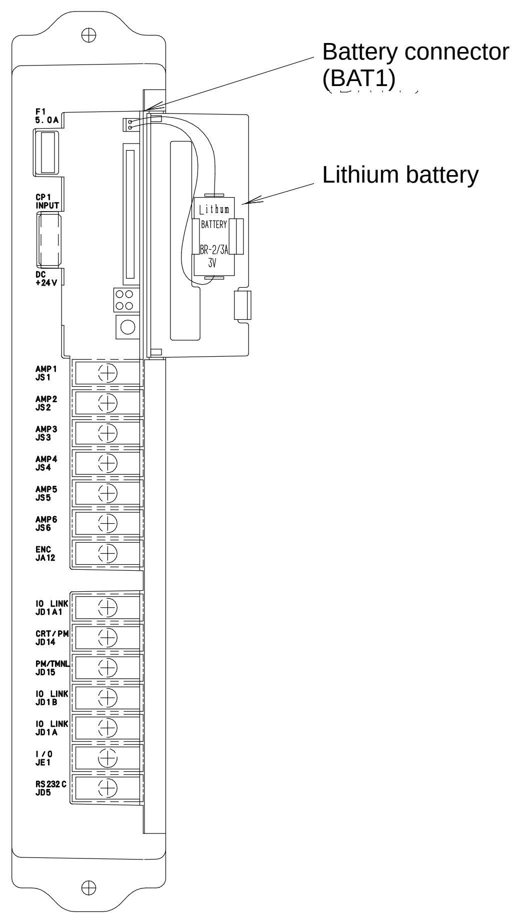


Fig 1.2 (b) Mounting location of lithium battery in Power Mate

APPENDIX

A TAPE CODE LIST

ISO code										EIA code											Meaning
Character	8	7	6	5	4		3	2	1	Character	8	7	6	5	4		3	2	1		
0			○	○		∘				0			○			∘					Number 0
1	○		○	○		○			○	1						○			○		Number 1
2	○		○	○		∘		○		2						∘		○			Number 2
3			○	○		∘		○	○	3				○		∘		○	○		Number 3
4	○		○	○		∘	○			4						∘	○				Number 4
5			○	○		∘	○		○	5				○		∘	○		○		Number 5
6			○	○		∘	○	○		6				○		∘	○	○			Number 6
7	○		○	○		∘	○	○	○	7						∘	○	○	○		Number 7
8	○		○	○	○	∘				8					○	∘					Number 8
9			○	○	○	∘			○	9				○	○	∘			○		Number 9
A		○				∘			○	a		○	○			∘			○		Address A
B		○				∘		○		b		○	○			∘		○			Address B
C	○	○				∘		○	○	c		○	○	○		∘		○	○		Address C
D		○				∘	○			d		○	○			∘	○				Address D
E	○	○				∘	○		○	e		○	○	○		∘	○		○	?	Address E
F	○	○				∘	○	○		f		○	○	○		∘	○	○			Address F
G		○				∘	○	○	○	g		○	○			∘	○	○	○		Address G
H		○			○	∘				h		○	○		○	∘					Address H
I	○	○			○	∘			○	i		○	○	○	○	∘			○		Address I
J	○	○			○	∘		○		j		○		○		∘		○	○		Address J
K		○			○	∘		○	○	k		○		○		∘		○			Address K
L	○	○			○	∘	○			l		○				∘		○	○		Address L
M		○			○	∘	○		○	m		○		○		∘	○				Address M
N		○			○	∘	○	○		n		○				∘	○		○		Address N
O	○	○			○	∘	○	○	○	o		○				∘	○	○			Address O
P		○		○		∘				p		○		○		∘	○	○	○		Address P
Q	○	○		○		∘			○	q		○		○	○	∘					Address Q
R	○	○		○		∘		○		r		○			○	∘			○		Address R
S		○		○		∘		○	○	s			○	○		∘		○			Address S
T	○	○		○		∘	○			t			○			∘		○	○		Address T
U		○		○		∘	○		○	u			○	○		∘	○				Address U
V		○		○		∘	○	○		v			○			∘	○		○		Address V
W	○	○		○		∘	○	○	○	w			○			∘	○	○			Address W
X	○	○		○	○	∘				x			○	○		∘	○	○	○		Address X
Y		○		○	○	∘			○	y			○	○	○	∘					Address Y
Z		○		○	○	∘		○		z			○		○	∘			○		Address Z

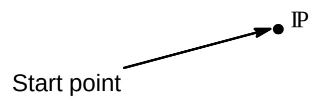
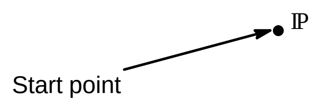
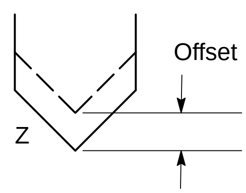
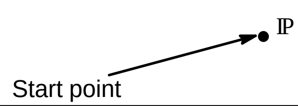
ISO code										EIA code										Meaning	
Character	8	7	6	5	4		3	2	1	Character	8	7	6	5	4		3	2	1		
DEL	○	○	○	○	○	∘	○	○	○	Del		○	○	○	○	∘	○	○	○	*	
NUL						∘				Blank						∘				*	
BS	○				○	∘				BS			○		○	∘		○		*	
HT					○	∘			○	Tab			○	○	○	∘	○	○		*	
LF or NL					○	∘		○		CR or EOB	○					∘					
CR	○				○	∘	○		○	___										*	
SP	○		○			∘				SP				○		∘				*	
%	○		○			∘	○		○	ER					○	∘		○	○		
(			○		○	∘				(2–4–5)				○	○	∘		○			
)	○		○		○	∘			○	(2–4–7)		○			○	∘		○			
+			○		○	∘		○	○	+		○	○	○		∘				*	
–			○		○	∘	○		○	–		○				∘					
:			○	○	○	∘		○		___											
/	○		○		○	∘	○	○	○	/			○	○		∘			○		
.			○		○	∘	○	○		.		○	○		○	∘		○	○		
#	○		○			∘		○	○	___											
$			○			∘	○			___										*	
&	○		○			∘	○	○		&					○	∘	○	○		*	
▽			○			∘	○	○	○	___										*	
*	○		○		○	∘		○		___										*	
,	○		○		○	∘	○			,			○	○	○	∘		○	○	*	
;	○		○	○	○	∘		○	○	___										*	
<			○	○	○	∘	○			___										*	
=	○		○	○	○	∘	○		○	___											
>	○		○	○	○	∘	○	○		___										*	
?			○	○	○	∘	○	○	○	___										*	
@	○	○				∘				___										*	
”			○					○		___										*	

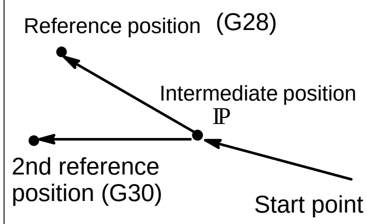
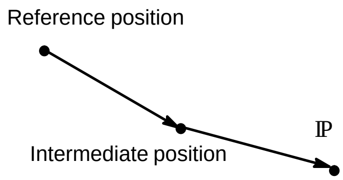
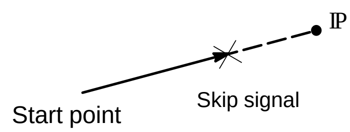
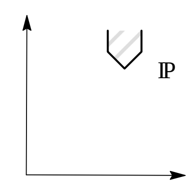
NOTE

1. Codes with *in the remarks column are ignored in the significant information section.
2. Codes with ? in the remarks column cause an alarm in the significant information section but are registered in memory.
3. Codes not in this table are ignored if their parity is correct.
4. Codes with incorrect parity cause the TH alarm. But they are ignored without generating the TH alarm when they are in the comment section.
5. A character with all eight holes punched is ignored and does not generate TH alarm in EIA code.

B LIST OF FUNCTIONS AND TAPE FORMAT

Some functions cannot be added as options depending on the model.
In the tables below, IP _:presents a combination of arbitrary axis addresses using X,Y,Z,A,B and C (such as X_Y_Z_A_).
x = 1st basic axis (X usually)
y = 2nd basic axis (Y usually)
z = 3rd basic axis (Z usually)

Functions	Illustration	Tape format
Positioning (G00)		G00 IP_ ;
Linear interpolation (G01)		G01 IP_ F_;
Dwell (G04) (In case of X–Y plane)		G04 { X_ P_ } ;
Change of offset value by program (G10)		G10 P_R_;
Tool length offset A (G43, G44, G49)		{ G43 G44 } Z_ H_ ; { G43 G44 } H_ ; H : Tool offset G49 : Cancel
Inch/millimeter conversion (G20, G21)		G20 ; Inch input G21 ; Millimeter input
Reference position return check (G27)		G27 IP_ ;

Functions	Illustration	Tape format
Reference position return (G28) 2nd, reference position return (G30)		G28 IP_ ; G30 IP_ ;
Return from reference position to start point (G29)		G29 IP_ ;
Skip function (G31)		G31 IP_ F_;
Absolute/incremental programming (G90/G91)		G90_ ; Absolute command G91_ ; Incremental command G90_ G91_ ; Combined use
Change of workpiece coordinate system (G92)		G92 IP_ ;

C RANGE OF COMMAND VALUE

Linear axis

- **In case of millimeter input, feed screw is millimeter**

	Increment system	
	IS–A	**IS–B**
Least input increment	0.01 mm	0.001 mm
Least command increment	0.01 mm	0.001 mm
Max. programmable dimension	±999999.99 mm	±99999.999 mm
Max. rapid traverse **NOTE**	240000 mm/min	240000 mm/min
Feedrate range **NOTE**	1 to 240000 mm/min	1 to 240000 mm/min
Incremental feed	0.01, 0.01, 0.1, 1, 10 mm/step	0.001, 0.01, 0.1, 1 mm/step
Tool compensation	0 to ±9999.99 mm	0 to ±999.999 mm
Dwell time	0 to 999999.99 sec	0 to 99999.999 sec

- **In case of inch input, feed screw is millimeter**

	Increment system	
	IS–A	**IS–B**
Least input increment 0.001 inch	0.001 inch	0.0001 inch
Least command increment	0.01 mm	0.001 mm
Max. programmable dimension	±99999.999 inch	±9999.9999 inch
Max. rapid traverse **NOTE**	240000 mm/min	240000 mm/min
Feedrate range **NOTE**	0.01 to 9600 inch/min	0.01 to 9600 inch/min
Incremental feed	0.001, 0.001, 0.01, 0.1, 1 inch/step	0.0001, 0.001, 0.01, 0.1 inch/step
Tool compensation	0 to ±999.999 inch	0 to ±99.9999 inch
Dwell time	0 to 999999.99 sec	0 to 99999.999 sec

- **In case of inch input, feed screw is inch**

	Increment system	
	IS–A	**IS–B**
Least input increment	0.001 inch	0.0001 inch
Least command increment	0.001 inch	0.0001 inch
Max. programmable dimension	±99999.999 inch	±9999.9999 inch
Max. rapid traverse **NOTE**	9600 inch/min	9600 inch/min
Feedrate range **NOTE**	0.01 to 9600 inch/min	0.01 to 9600 inch/min
Incremental feed	0.001, 0.01, 0.1, 1 inch/step	0.0001, 0.001, 0.01, 0.1 inch/step
Tool compensation	0 to ±999.999 inch	0 to ±99.9999 inch
Dwell time	0 to 999999.99 sec	0 to 99999.999 sec

- **In case of millimeter input, feed screw is inch**

	Increment system	
	IS–A	**IS–B**
Least input increment	0.01 mm	0.001 mm
Least command increment	0.001 inch	0.0001 inch
Max. programmable dimension	±999999.99 mm	±99999.999 mm
Max. rapid traverse **NOTE**	9600 inch/min	9600 inch/min
Feedrate range **NOTE**	1 to 240000 mm/min	1 to 240000 mm/min
Incremental feed	0.01, 0.1, 1, 10 mm/step	0.001, 0.01, 0.1, 1 mm/step
Tool compensation	0 to ±9999.99 mm	0 to ±999.999 mm
Dwell time	0 to 999999.99 sec	0 to 99999.999 sec

Rotation axis

	Increment system	
	IS–A	**IS–B**
Least input increment	0.01 deg	0.001 deg
Least command increment	0.01 deg	0.001 deg
Max. programmable dimension	±999999.99 deg	±99999.999 deg
Max. rapid traverse **NOTE**	240000 deg/min	240000 deg/min
Feedrate range **NOTE**	1 to 240000 deg/min	1 to 240000 deg/min
Incremental feed	0.01, 0.1, 1, 10 deg/step	0.001, 0.01, 0.1, 1 deg/step

NOTE
The feedrate range shown above are limitations depending on controller interpolation capacity. As a whole system, limitations depending on servo system must also be considered.

D NOMOGRAPHS

D.1 TOOL PATH AT CORNER

When servo system delay (by exponential acceleration/deceleration at cutting or caused by the positioning system when a servo motor is used) is accompanied by cornering, a slight deviation is produced between the tool path (tool center path) and the programmed path as shown in Fig. D.1 (a).
Time constant T_1 of the exponential acceleration/deceleration is fixed to 0.

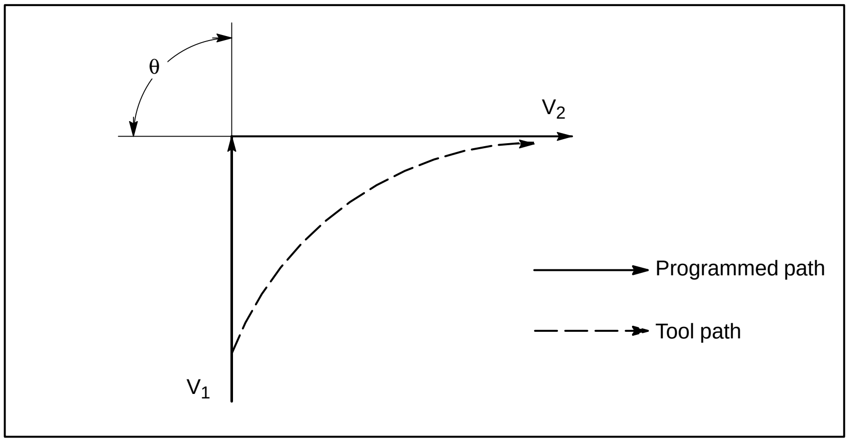


Fig. D.1 (a) Slight deviation between the tool path and the programmed path

This tool path is determined by the following parameters:

· Feedrate (V_1, V_2)
· Corner angle (θ)
· Exponential acceleration / deceleration time constant (T_1) at cutting (T_1 = 0)
· Presence or absence of buffer register.

The above parameters are used to theoretically analyze the tool path and above tool path is drawn with the parameter which is set as an example. When actually programming, the above items must be considered and programming must be performed carefully so that the shape of the workpiece is within the desired precision.
In other words, when the shape of the workpiece is not within the theoretical precision, the commands of the next block must not be read until the specified feedrate becomes zero. The dwell function is then used to stop the machine for the appropriate period.

Analysis

The tool path shown in Fig. D.1 (b) is analyzed based on the following conditions:
Feedrate is constant at both blocks before and after cornering.
The controller has a buffer register. (The error differs with the reading speed of the tape reader, number of characters of the next block, etc.)

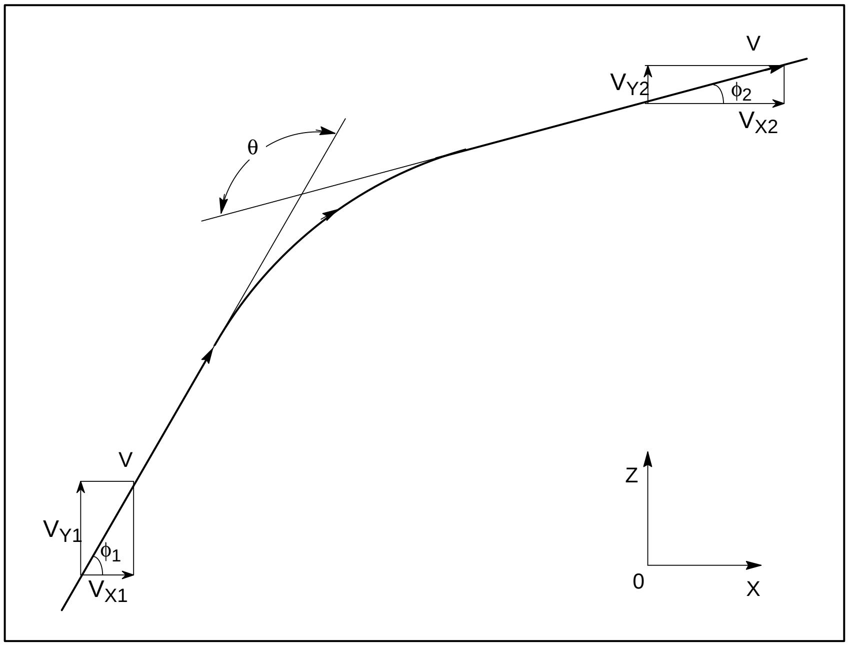


Fig. D.1(b) Example of tool path

- **Description of conditions and symbols**

$$V_{X1} = V\cos\phi_1$$

$$V_{Y1} = V\sin\phi_1$$

$$V_{X2} = V\cos\phi_2$$

$$V_{Y2} = V\sin\phi_2$$

V : Feedrate at both blocks before and after cornering
V_{X1} : X–axis component of feedrate of preceding block
V_{Y1} : Y–axis component of feedrate of preceding block
V_{X2} : X–axis component of feedrate of following block
V_{Y2} : Y–axis component of feedrate of following block
θ : Corner angle
ϕ_1 : Angle formed by specified path direction of preceding block and X–axis
ϕ_2 : Angle formed by specified path direction of following block and X–axis

• Initial value calculation

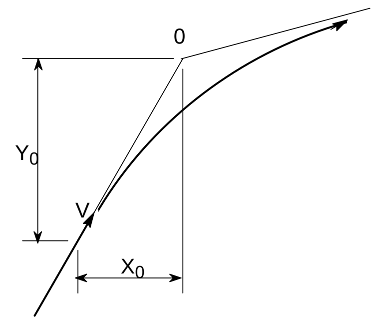


Fig. D.1(c) Initial value

The initial value when cornering begins, that is, the X and Y coordinates at the end of command distribution by the controller, is determined by the feedrate and the positioning system time constant of the servo motor.

$$X_0 = V_{X1}(T_1 + T_2)$$

$$Y_0 = V_{Y1}(T_1 + T_2)$$

T_1:Exponential acceleration / deceleration time constant. (T=0)
T_2:Time constant of positioning system (Inverse of position loop gain)

• Analysis of corner tool path

The equations below represent the feedrate for the corner section in X–axis direction and Y–axis direction.

$$V_X(t) = (V_{X2}-V_{X1})[1-\frac{V_{X1}}{T_1-T_2}\{T_1\exp(-\frac{t}{T_1})-T_2\exp(-\frac{t}{T_2})\} + V_{X1}]$$

$$= V_{X2}[1-\frac{V_{X1}}{T_1-T_2}\{T_1\exp(-\frac{t}{T_1})-T_2\exp(-\frac{t}{T_2})\}]$$

$$V_Y(t) = \frac{V_{Y1}-V_{Y2}}{T_1-T_2}\{T_1\exp(-\frac{t}{T_1})-T_2\exp(-\frac{t}{T_2})\} + V_{Y2}$$

Therefore, the coordinates of the tool path at time *t* are calculated from the following equations:

$$X(t) = \int_0^t V_X(t)dt-X_0$$

$$= \frac{V_{X2}-V_{X1}}{T_1-T_2}\{T_1^2\exp(-\frac{t}{T_1})-T_2^2\exp(-\frac{t}{T_2})\}-V_{X2}(T_1 + T_2-t)$$

$$Y(t) = \int_0^t V_Y(t)dt-Y_0$$

$$= \frac{V_{Y2}-V_{Y1}}{T_1-T_2}\{T_1^2\exp(-\frac{t}{T_1})-T_2^2\exp(-\frac{t}{T_2})\}-V_{Y2}(T_1 + T_2-t)$$

E STATUS WHEN TURNING POWER ON, WHEN CLEAR AND WHEN RESET

Parameter 3402 (CLR) is used to select whether resetting the CNC places it in the cleared state or in the reset state (0: reset state/1: cleared state).
The symbols in the tables below mean the following :
○ : The status is not changed or the movement is continued.
× : The status is cancelled or the movement is interrupted.

Item		When turning power on	Cleared	Reset
Setting data	Offset value	○	○	○
	Data set by the MDI setting operation	○	○	○
	Parameter	○	○	○
Various data	Programs in memory	○	○	○
	Contents in the buffer storage	×	×	○ : MDI mode × : Other mode
	Display of sequence number	○	○ (NOTE 1)	○ (NOTE 1)
	One shot G code	×	×	×
	Modal G code	Initial G codes. (The G20 and G21 codes return to the same state they were in when the power was last turned off.)	Initial G codes. (G20/G21 are not changed.)	○
	F	Zero	Zero	○
	M	×	○	○
Work coordinate value		Zero	○	○
Action in operation	Movement	×	×	×
	Dwell	×	×	×
	Issuance of M code	×	×	×
	Tool length compensation	×	Depending on parameter LVK(No.5003#6)	○ : MDI mode Other modes depend on parameter LVK(No.5003#6).
	Storing called subprogram number	×	× (Note 2)	○ : MDI mode × : Other modes (NOTE 2)

Item		When turning power on	Cleared	Reset
Output signals	CNC alarm signal AL	Extinguish if there is no cause for the alarm	Extinguish if there is no cause for the alarm	Extinguish if there is no cause for the alarm
	Reference position return completion LED	×	○ (× : Emergency stop)	○ (× : Emergency stop)
	M code	×	×	×
	M, T strobe signal	×	×	×
	Controller ready signal MA	ON	○	○
	Servo ready signal SA	ON (When other than servo alarm)	ON (When other than servo alarm)	ON (When other than servo alarm)
	Cycle start LED (STL)	×	×	×
	Feed hold LED (SPL)	×	×	×

NOTE

1. When heading is performed, the main program number is displayed.
2. When a reset is performed during execution of a subprogram, control returns the head of main program by heading function.
   Execution cannot be started from the middle of the subprogram.

F CHARACTER–TO–CODES CORRESPONDENCE TABLE

Character	Code	Comment	Character	Code	Comment
A	065		6	054	
B	066		7	055	
C	067		8	056	
D	068		9	057	
E	069			032	Space
F	070		”	034	Quotation mark
G	071		#	035	Hash sign
H	072		$	036	Dollar sign
I	073		%	037	Percent
J	074		&	038	Ampersand
K	075		’	039	Apostrophe
L	076		(	040	Left parenthesis
M	077		)	041	Right parenthesis
N	078		*	042	Asterisk
O	079		+	043	Plus sign
P	080		,	044	Comma
Q	081		–	045	Minus sign
R	082		.	046	Period
S	083		/	047	Slash
T	084		:	058	Colon
U	085		;	059	Semicolon
V	086		<	060	Left angle bracket
W	087		=	061	Equal sign
X	088		>	062	Right angle bracket
Y	089		?	063	Question mark
Z	090		@	064	At mark
0	048		[	091	Left square bracket
1	049		^	092	
2	050		¥	093	Yen sign
3	051		]	094	Right square bracket
4	052		_	095	Underscore
5	053				

G ALARM LIST

1) Program errors (P/S alarm)

Number	Message	Contents
000	PLEASE TURN OFF POWER	A parameter which requires the power off was input, turn off power.
001	TH PARITY ALARM	TH alarm (A character with incorrect parity was input). Correct the tape.
002	TV PARITY ALARM	TV alarm (The number of characters in a block is odd). This alarm will be generated only when the TV check is effective.
003	TOO MANY DIGITS	Data exceeding the maximum allowable number of digits was input. (Refer to the item of max. programmable dimensions.)
004	ADDRESS NOT FOUND	A numeral or the sign “ – ” was input without an address at the beginning of a block. Modify the program .
005	NO DATA AFTER ADDRESS	The address was not followed by the appropriate data but was followed by another address or EOB code. Modify the program.
006	ILLEGAL USE OF NEGATIVE SIGN	Sign “ – ” input error (Sign “ – ” was input after an address with which it cannot be used. Or two or more “ – ” signs were input.) Modify the program.
007	ILLEGAL USE OF DECIMAL POINT	Decimal point “ .” input error (A decimal point was input after an address with which it can not be used. Or two decimal points were input.) Modify the program.
009	ILLEGAL ADDRESS INPUT	Unusable character was input in significant area. Modify the program.
010	IMPROPER G–CODE	An unusable G code or G code corresponding to the function not provided is specified. Modify the program.
011	NO FEEDRATE COMMANDED	Feedrate was not commanded to a cutting feed or the feedrate was inadequate. Modify the program.
014	CAN NOT COMMAND G95	A synchronous feed is specified without the option for synchronous feed.
015	TOO MANY AXES COMMANDED	The number of the commanded axes exceeded that of simultaneously controlled axes.
021	ILLEGAL PLANE AXIS COMMANDED	An axis not included in the selected plane (by using G17, G18, G19) was commanded. Modify the program.
027	NO AXES COMMANDED IN G43/G44	No axis is specified in G43 and G44 blocks for the tool length offset. Offset is not canceled but another axis is offset for the tool length offset. Modify the program.
028	ILLEGAL PLANE SELECT	In the plane selection command, two or more axes in the same direction are commanded. Modify the program.
029	ILLEGAL OFFSET VALUE	The offset values specified by H code is too large. Modify the program.
030	ILLEGAL OFFSET NUMBER	The offset number specified by H code for tool length offset is too large. Modify the program.
031	ILLEGAL P COMMAND IN G10	In setting an offset amount by G10, the offset number following address P was excessive or it was not specified. Modify the program.

Number	Message	Contents
032	ILLEGAL OFFSET VALUE IN G10	In setting an offset amount by G10 or in writing an offset amount by system variables, the offset amount was excessive.
046	ILLEGAL REFERENCE RETURN COMMAND	Other than P2 and P3 are commanded for 2nd and 3rd reference position return command.
059	PROGRAM NUMBER NOT FOUND	In an external program number search, a specified program number was not found. Otherwise, a program specified for searching is being edited in background processing. Check the program number and external signal. Or discontinue the background eiting.
060	SEQUENCE NUMBER NOT FOUND	Commanded sequence number was not found in the sequence number search. Check the sequence number.
070	NO PROGRAM SPACE IN MEMORY	The memory area is insufficient. Delete any unnecessary programs, then retry.
071	DATA NOT FOUND	The address to be searched was not found. Or the program with specified program number was not found in program number search. Check the data.
072	TOO MANY PROGRAMS	The number of programs to be stored exceeded 63 (basic), 125 (option), 200 (option), or 400 (option). Delete unnecessary programs and execute program registeration again.
073	PROGRAM NUMBER ALREADY IN USE	The commanded program number has already been used. Change the program number or delete unnecessary programs and execute program registeration again.
074	ILLEGAL PROGRAM NUMBER	The program number is other than 1 to 9999. Modify the program number.
075	PROTECT	An attempt was made to register a program whose number was protected.
076	ADDRESS P NOT DEFINED	Address P (program number) was not commanded in the block which includes an M98, G65, or G66 command. Modify the program.
077	SUB PROGRAM NESTING ERROR	The subprogram was called in five folds. Modify the program.
078	NUMBER NOT FOUND	A program number or a sequence number which was specified by address P in the block which includes an M98, M99, M65 or G66 was not found. The sequence number specified by a GOTO statement was not found. Otherwise, a called program is being edited in background processing. Correct the program, or discontinue the background editing.
079	PROGRAM VERIFY ERROR	In memory or program collation,a program in memory does not agree with that read from an external I/O device. Check both the programs in memory and those from the external device.
085	COMMUNICATION ERROR	When entering data in the memory by using Reader / Puncher interface, an overrun, parity or framing error was generated. The number of bits of input data or setting of baud rate or specification No. of I/O unit is incorrect.
086	DR SIGNAL OFF	When entering data in the memory by using Reader / Puncher interface, the ready signal (DR) of reader / puncher was off. Power supply of I/O unit is off or cable is not connected or a P.C.B. is defective.
087	BUFFER OVERFLOW	When entering data in the memory by using Reader / Puncher interface, though the read terminate command is specified, input is not interrupted after 10 characters read. I/O unit or P.C.B. is defective.
090	REFERENCE RETURN INCOMPLETE	The reference position return cannot be performed normally because the reference position return start point is too close to the reference position or the speed is too slow. Separate the start point far enough from the reference position, or specify a sufficiently fast speed for reference position return.

Number	Message	Contents
092	AXES NOT ON THE REFERENCE POINT	The commanded axis by G27 (Reference position return check) did not return to the reference position.
100	PARAMETER WRITE ENABLE	On the PARAMETER(SETTING) screen, PWE(parameter writing enabled) is set to 1. Set it to 0, then reset the system.
101	PLEASE CLEAR MEMORY	The power turned off while rewriting the memory by program edit operation. If this alarm has occurred, press <RESET> while pressing <PROG>, and only the program being edited will be deleted. Register the deleted program.
110	DATA OVERFLOW	The absolute value of fixed decimal point display data exceeds the allowable range. Modify the program.
111	CALCULATED DATA OVERFLOW	The result of calculation result is out of the allowable range. (-10^{47} to -10^{-29}, 0, and 10^{-29} to 10^{47}).
112	DIVIDED BY ZERO	Division by zero was specified. (including tan 90°)
113	IMPROPER COMMAND	A function which cannot be used in custom macro is commanded. Modify the program.
114	FORMAT ERROR IN MACRO	There is an error in other formats than <Formula>. Modify the program.
115	ILLEGAL VARIABLE NUMBER	A value not defined as a variable number is designated in the custom macro. This alarm is given in the following cases: 1. The header corresponding to the specified machining cycle number called is not found. 2. The cycle connection data value is out of the allowable range (0 – 999). 3. The number of data in the header is out of the allowable range (0 – 32767). 4. The start data variable number of executable format data is out of the allowable range (#20000 – #85535). 5. The storing data variable number of executable format data is out of the allowable range (#85535). 6. The storing start data variable number of executable format datais overlapped with the variable number used in the header. Modify the program.
116	WRITE PROTECTED VARIABLE	The left side of substitution statement is a variable whose substitution is inhibited. Modify the program.
118	PARENTHESIS NESTING ERROR	The nesting of bracket exceeds the upper limit (quintuple). Modify the program.
119	ILLEGAL ARGUMENT	The SQRT argument is negative, BCD argument is negative, or other values than 0 to 9 are present on each line of BIN argument. Modify the program.
122	DUPLICATE MACRO MODAL–CALL	The macro modal call is specified in double. Modify the program.
124	MISSING END STATEMENT	DO – END does not correspond to 1 : 1. Modify the program.
125	FORMAT ERROR IN MACRO	<Formula> format is erroneous. Modify the program.
126	ILLEGAL LOOP NUMBER	In DOn, $1 \leqq n \leqq 3$ is not established. Modify the program.
127	NC, MACRO STATEMENT IN SAME BLOCK	NC and custom macro commands coexist. Modify the program.
128	ILLEGAL MACRO SEQUENCE NUMBER	The sequence number specified in the branch command was not 0 to 9999. Or, it cannot be searched. Modify the program.
129	ILLEGAL ARGUMENT ADDRESS	An address which is not allowed in <Argument Designation > is used. Modify the program.

Number	Message	Contents
130	ILLEGAL AXIS OPERATION	An axis control command was given by PMC to an axis controlled by controller. Or an axis control command was given by controller to an axis controlled by PMC. Modify the program.
131	TOO MANY EXTERNAL ALARM MESSAGES	Five or more alarms have generated in external alarm message. Consult the PMC ladder diagram to find the cause.
132	ALARM NUMBER NOT FOUND	No alarm No. concerned exists in external alarm message clear. Check the PMC ladder diagram.
133	ILLEGAL DATA IN EXT. ALARM MSG	Small section data is erroneous in external alarm message or external operator message. Check the PMC ladder diagram.
139	CAN NOT CHANGE PMC CONTROL AXIS	An axis is selected in commanding by PMC axis control. Modify the program.
159	TOOL DATA SETTING INCOMPLETE	During executing a life data setting program, power was turned off. Set again.
179	PARAM. (NO. 7510) SETTING ERROR	The number of controlled axes set by the parameter 7510 exceeds the maximum number. Modify the parameter setting value.
199	MACRO WORD UNDEFINED	Undefined macro word was used. Modify the custom macro.
222	DNC OP. NOT ALLOWED IN BG.–EDIT	Input and output are executed at a time in the background edition. Execute a correct operation.
224	RETURN TO REFERENCE POINT	Reference position return has not been performed before the automatic operation starts. Perform reference position return only when bit 0 of parameter 1005 ZRN_X is 0.
231	ILLEGAL FORMAT IN G10 OR L50	Any of the following errors occurred in the specified format at the programmable–parameter input. 1) Address N or R was not entered. 2) A number not specified for a parameter was entered. 3) The axis number was too large. 4) An axis number was not specified in the axis–type parameter. 5) An axis number was specified in the parameter which is not an axis type.
233	DEVICE BUSY	When an attempt was made to use a unit such as that connected via the RS–232–C interface, other users were using it.
239	BP/S ALARM	While punching was being performed with the function for controlling external I/O units ,background editing was performed.
240	BP/S ALARM	Background editing was performed during MDI operation.
5010	END OF RECORD	The end of record (%) was specified.

2) Background edit alarm

Number	Message	Contents
???	BP/S alarm	BP/S alarm occurs in the same number as the P/S alarm that occurs in ordinary program edit. (070, 071, 072, 073, 074 085,086,087 etc.)
140	BP/S alarm	It was attempted to select or delete in the background a program being selected in the foreground. **(Note)** Use background editing correctly.

NOTE

Alarm in background edit is displayed in the key input line of the background edit screen instead of the ordinary alarm screen and is resettable by any of the MDI key operation.

3) Absolute pulse coder (APC) alarm

Number	Message	Contents
300	nth–axis origin return	Manual reference position return is required for the nth–axis (n=1 – 6).
301	APC alarm: nth–axis communication	nth–axis (n=1 – 6) APC communication error. Failure in data transmission Possible causes include a faulty APC, cable, or servo interface module.
302	APC alarm: nth–axis over time	nth–axis (n=1 – 6) APC overtime error. Failure in data transmission. Possible causes include a faulty APC, cable, or servo interface module.
303	APC alarm: nth–axis framing	nth–axis (n=1 – 6) APC framing error. Failure in data transmission. Possible causes include a faulty APC, cable, or servo interface module.
304	APC alarm: nth–axis parity	nth–axis (n=1 – 6) APC parity error. Failure in data transmission. Possible causes include a faulty APC, cable, or servo interface module.
305	APC alarm: nth–axis pulse error	nth–axis (n=1 – 6) APC pulse error alarm. APC alarm.APC or cable may be faulty.
306	APC alarm: nth–axis battery voltage 0	nth–axis (n=1 – 6) APC battery voltage has decreased to a low level so that the data cannot be held. APC alarm. Battery or cable may be faulty.
307	APC alarm: nth–axis battery low 1	nth–axis (n=1 – 6) axis APC battery voltage reaches a level where the battery must be renewed. APC alarm. Replace the battery.
308	APC alarm: nth–axis battery low 2	nth–axis (n=1 – 6) APC battery voltage has reached a level where the battery must be renewed (including when power is OFF). APC alarm .Replace battery.

4) Serial pulse coder (SPC) alarms

When either of the following alarms is issued, a possible cause is a faulty serial pulse coder or cable.

Number	Message	Contents
350	SPC ALARM: n AXIS PULSE CODER	The n axis (axis 1–6) pulse coder has a fault. Refer to diagnosis display No. 202 for details.
351	SPC ALARM: n AXIS COMMUNICATION	n axis (axis 1–6) serial pulse coder communication error (data transmission fault) Refer to diagnosis display No. 203 for details.

- **The details of serial pulse coder alarm No.350**

The details of serial pulse coder alarm No. 350 (pulse coder alarm) are displayed in the diagnosis display (No. 202) as shown below.

	#7	#6	#5	#4	#3	#2	#1	#0
202		CSA	BLA	PHA	RCA	BZA	CKA	SPH

CSA : The serial pulse coder is defective. Replace it.
BLA : The battery voltage is low. Replace the batteries. This alarm has nothing to do with alarm 350 (serial pulse coder alarm).
SPH : The serial pulse coder or feedback cable is defective. Replace the serial pulse coder or cable.
RCA: The serial pulse coder is defective. Replace it.
BZA : The pulse coder was supplied with power for the first time. Make sure that the batteries are connected. Turn the power off, then turn it on again and perform a reference position return. This alarm has nothing to do with alarm 350 (serial pulse coder alarm).
CKA: The serial pulse coder is defective. Replace it.
PHA : The serial pulse coder or feedback cable is defective. Replace the serial pulse coder or cable.

- **The details of serial pulse coder alarm No.351**

The details of serial pulse coder alarm No. 351 (communication alarm) are displayed in the diagnosis display (No. 203) as shown below.

	#7	#6	#5	#4	#3	#2	#1	#0
203	DTE	CRC	STB	PRM				

DTE : The serial pulse coder encountered a communication error. The pulse coder, feedbak cable, or feedback receiver circuit is defective. Replace the pulse coder, feedback cable, or NC–axis module.
CRC: The serial pulse coder encountered a communication error. The pulse coder, feedback cable, or feedback receiver circuit is defective. Replace the pulse coder, feedback cable, or NC–axis module.
STB : The serial pulse coder encountered a communication error. The pulse coder, feedback cable, or feedback receiver circuit is defective. Replace the pulse coder, feedback cable, or NC–axis module.
PRM: An invalid parameter was found. Alarm 417 (invalid servo parameter) is also issued.

5) Servo alarms

Number	Message	Contents
400	SERVO ALARM: n–TH AXIS OVERLOAD	The n–th axis (axis 1–6) overload signal is on. Refer to diagnosis display No. 201 for details.
401	SERVO ALARM: n–TH AXIS VRDY OFF	The n–th axis (axis 1–6) servo amplifier READY signal (DRDY) went off.
404	SERVO ALARM: n–TH AXIS VRDY ON	Even though the n–th axis (axis 1–6) READY signal (MCON) went off, the servo amplifier READY signal (DRDY) is still on. Or, when the power was turned on, DRDY went on even though MCON was off. Check that the servo interface module and servo amp are connected.
405	SERVO ALARM: (ZERO POINT RETURN FAULT)	Position control system fault. Due to an NC or servo system fault in the reference position return, there is the possibility that reference position return could not be executed correctly. Try again from the manual reference position return.
410	SERVO ALARM: n–TH AXIS – EXCESS ERROR	The position deviation value when the n–th axis (axis 1–6) stops is larger than the set value. Note) Limit value must be set to parameter No.1829 for each axis.
411	SERVO ALARM: n–TH AXIS – EXCESS ERROR	The position deviation value when the n–th axis (axis 1–6) moves is larger than the set value. Note) Limit value must be set to parameter No.1828 for each axis.
413	SERVO ALARM: n–th AXIS – LSI OVERFLOW	The contents of the error register for the n–th axis (axis 1–6) and beyond the range of -2^{31} to 2^{31}. This error usually occurs as the result of an improperly set parameters.
414	SERVO ALARM: n–TH AXIS – DETECTION RELATED ERROR	N–th axis (axis 1–6) digital servo system fault. Refer to diagnosis display No. 200 and No.204 for details.
415	SERVO ALARM: n–TH AXIS – EXCESS SHIFT	A speed higher than 511875 units/s was attempted to be set in the n–th axis (axis 1–6). This error occurs as the result of improperly set CMR.
416	SERVO ALARM: n–TH AXIS – DISCONNECTION	Position detection system fault in the n–th axis (axis 1–6) pulse coder (disconnection alarm). Refer to diagnosis display No. 201 for details.
417	SERVO ALARM: n–TH AXIS – PARAMETER INCORRECT	This alarm occurs when the n–th axis (axis 1–6) is in one of the conditions listed below. (Digital servo system alarm) 1) The value set in Parameter No. 2020 (motor form) is out of the specified limit. 2) A proper value (111 or –111) is not set in parameter No.2022 (motor revolution direction). 3) Illegal data (a value below 0, etc.) was set in parameter No. 2023 (number of speed feedback pulses per motor revolution). 4) Illegal data (a value below 0, etc.) was set in parameter No. 2024 (number of position feedback pulses per motor revolution). 5) Parameters No. 2084 and No. 2085 (flexible field gear rate) have not been set. 6) A value outside the limit of {1 to the number of control axes} or a non–continuous value (Parameter 1023 (servo axis number) contains a value out of the range from 1 to the number of axes, or an isolated value (for example, 7 not prceded by 6).was set in parameter No. 1023 (servo axisnumber).

- **Details of servo alarm No.414**

The details of servo alarm No. 414 are displayed in the diagnosis display (No. 200 and No.204) as shown below.

	#7	#6	#5	#4	#3	#2	#1	#0
200	OVL	LV	OVC	HCA	HVA	DCA	FBA	OFA

OVL : An overload alarm is being generated.
(This bit causes servo alarm No. 400. The details are indicated in diagnostic data No.201).
LV : A low voltage alarm is being generated in servo amp.
Check LED.
OVC : A overcurrent alarm is being generated inside of digital servo.
HCA : An abnormal current alarm is being generated in servo amp.
Check LED.
HVA : An overvoltage alarm is being generated in servo amp.
Check LED.
DCA : A regenerative discharge circuit alarm is being generated in servo amp.
Check LED.
FBA : A disconnection alarm is being generated.
(This bit causes servo alarm No.416.The details are indicated in diagnostic data No. 201)
OFA : An overflow alarm is being generated inside of digital servo.

	#7	#6	#5	#4	#3	#2	#1	#0
204		OFS	MCC	LDA	PMS			

OFS : A current conversion error has occured in the digital servo.
MCC : A magnetic contactor contact in the servo amplifier has welded.
LDA : The LED indicates that serial pulse coder C is defective
PMS : A feedback pulse error has occured because the feedback cable is defective.

- **Details of servo alarms No. 400 and No.416**

The details of servo alarms No. 400 and No. 416 are displayed in the diagnosis display (No. 201) as shown below.

	#7	#6	#5	#4	#3	#2	#1	#0
201	ALDF			EXP				

When OVL equal 1 in diagnostic data No.200 (servo alarm No. 400 is being generated):

ALDF0 : Motor overheating
1 : Amplifier overheating

When FBAL equal 1 in diagnostic data No.200 (servo alarm No. 416 is being generated):

ALDF	EXP	Alarm details
1	0	Built–in pulse coder disconnection (hardware)
1	1	Separately installed pulse coder disconnection (hardware)
0	0	Pulse coder is not connected due to software.

6) Over travel alarms

Number	Message	Contents
500	OVER TRAVEL : +n	Exceeded the n–th axis (axis 1–6) + side stored stroke limit I. (Parameter No.1320 or 1326 **Notes**)
501	OVER TRAVEL : –n	Exceeded the n–th axis (axis 1–6) – side stored stroke limit I. (Parameter No.1321 or 1327 **Notes**)

7) Overheat alarms

Number	Message	Contents
700	OVERHEAT: CONTROL UNIT	Control unit overheat Check that the fan motor operates normally, and clean the air filter.
701	OVERHEAT: FAN MOTOR	The fan motor on the top of the cabinet for the contorl unit is overheated. Check the operation of the fan motor and replace the motor if necessary.

8) System alarms

(These alarms cannot be reset with reset key.)

Number	Message	Contents
900	ROM PARITY	ROM parity error (CNC/MACRO/LADDER/Servo) Rewrite the flash ROM with the indicated ROM number.
910	DRAM PARITY : (LOW)	RAM parity error in the DRAM module. Replace the DRAM module.
911	DRAM PARITY: (HIGH)	RAM parity error in the DRAM module. Replace the DRAM module.
912	SRAM PARITY: (LOW)	RAM parity error in the tape memory RAM module. Clear the memory or replace the base PCB. After this operation, reset all data including the parameters.
913	SRAM PARITY : (HIGH)	RAM parity error in the tape memory RAM module. Clear the memory or replace the base PCB. After this operation, reset all data including the parameters.
920	SERVO ALARM (1/2 AXIS)	Servo alarm (1st or 2nd axis). A watchdog alarm or a RAM parity error in the servo module occurred. Replace the servo control module on the main CPU board.
921	SERVO ALARM (3/4 AXIS)	Servo alarm (3rd or 4th axis). A watchdog alarm or a RAM parity error in the servo module occurred. Replace the servo control module on the main CPU board.
922	SERVO ALARM (5/6 AXIS)	Servo alarm (5th or 6th axis). A watchdog alarm or a RAM parity error in the servo module occurred. Replace the servo control module on the main CPU board.
924	SERVO MODULE SETTING ERROR	The digital servo module is not installed. Check that the servo control module or servo interface module on the main CPU board is mounted securely.
930	CPU INTERRUPUT	CPU error (abnormal interrupt) The main CPU board is faulty.
950	PMC SYSTEM ALARM	Fault occurred in the PMC. The PMC control module on the main CPU board may be faulty.
951	PMC–PA WATCH DOG ALARM	Fault occurred in the PMC–PA (watchdog alarm).
970	NMI OCCURRED IN BOC	RAM parity error or NMI occurred in the PMC–PA module.
971	NMI OCCURRED IN SLC	An alarm condition occurred in the interface with an I/O unit. Check that the PMC control module on the main CPU board is conneted to the I/O unit securely. Check that the I/O unit is supplied with power and that the interface module is intact.
973	NON MASK INTERRUPT	NMI occurred for an unknown reason.

H LIST OF OPERATION

(CRT/MDI)

Classification	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Reset	Resetting the operating time			–	POS	[(OPRT)] [TIME: 0] → [EXEC]
	Resetting the number of machined parts			–	POS	[(OPRT)] [TIME: 0] → [EXEC]
	Resetting the OT alarm			When the power is on	–	P and CAN
	Resetting alarm 100			–	–	CAN and RESET
Data input from the MDI	Inputting parameters		○	MDI or emergency stop	SYSTEM (PARAM)	Parameter No. → [NO.SRH] → Data → INPUT → PWE = 0 → RESET
	Inputting offset data	○		–	OFFSET	Offset No. → [NO.SRH] → Offset value → INPUT
	Inputting setting data	○		MDI	SETTING	Setting No. → [NO.SRH] → Data → INPUT
	Inputting PMC parameters (for the counter and data table)	○		MDI or emergency stop	SYSTEM (PMC)	[PMCPRM] → [COUNTR] [DATA] → Data → INPUT
	Inputting PMC parameters (for the timer and keep relay)		○			[PMCPRM] → [TIMER] [KEEPRL] → Data → INPUT
Data input from external I/O units	Inputting parameters		○	EDIT or emergency stop	SYSTEM (PARAM)	[(OPRT)] → [▷] → [READ] → [EXEC]
	Inputting PMC parameters		○	Emergency stop	SYSTEM (PMC)	[▷] → [I/O]→ (CANNEL NO) 1 INPUT → (DEVICE NAME) [FDCAS] → (KIND OF DATA) [PARAM] → [READ] → (FILE NO) File No INPUT → [EXEC]
	Inputting offset data	○		EDIT	OFFSET	[(OPRT)] → [▷] → [READ] → [EXEC]
	Inputting programs	○		EDIT	PROG	[(OPRT)] → [▷] → [READ] → [EXEC]
Data output to external I/O units	Outputting parameters			EDIT	SYSTEM (PARAM)	[(OPRT)] → [▷] → [PUNCH] → [EXEC]
	Outputting PMC parameters			EDIT	SYSTEM (PMC)	[Continuous menu key] → [I/O]→ (CANNEL NO) 1 INPUT → (DEVICE NAME) [FDCAS] → (KIND OF DATA) [PARAM] → [WRITE] → (FILE NO) – 1 INPUT [EXEC]
	Outputting offset data			EDIT	OFFSET	[(OPRT)] → [▷] → [PUNCH] → [EXEC]
	Outputting all the programs			EDIT	PROG	O → –9999 → ▷] → [PUNCH] → [EXEC]
	Outputting one program			EDIT	PROG	O → Program No. → [▷] → [PUNCH] → [EXEC]

Classifi-cation	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Search	Searching for a program number			AUTO or EDIT	PROG	$\boxed{\text{O}}$ → Program No. → [O SRH]
	Searching for a sequence number			AUTO	PROG	Program No. search → $\boxed{\text{N}}$ Sequence No. → [N SRH]
	Searching for an address word			EDIT	PROG	Data to be searched for → [SRH↑] or [SRH ↓]
	Searching for an address only			EDIT	PROG	Address to be searched for → [SRH↑] or [SRH ↓]
	Searching for an offset number			–	OFFSET	Offset No. → [NO.SRH]
	Searching for a diagnosis number			–	SYSTEM (DGNOS)	Diagnosis No. → [NO.SRH]
	Searching for a parameter number			–	SYSTEM (PARAM)	Parameter No. → [NO.SRH]
Edit	Displaying the amount of memory used			EDIT	PROG	[LIBRARY]
	Deleting all the programs	○		EDIT	PROG	$\boxed{\text{O}}$ → –9999 → $\boxed{\text{DELETE}}$
	Deleting one program	○		EDIT	PROG	$\boxed{\text{O}}$ → Program No. → $\boxed{\text{DELETE}}$
	Deleting some blocks	○		EDIT	PROG	$\boxed{\text{N}}$ → Sequence No. → $\boxed{\text{DELETE}}$
	Deleting one block	○		EDIT	PROG	$\boxed{\text{EOB}}$ → $\boxed{\text{DELETE}}$
	Deleting a word	○		EDIT	PROG	Searching for the word to be deleted → $\boxed{\text{DELETE}}$
	Changing a word	○		EDIT	PROG	Searching for the word to be changed → New data → $\boxed{\text{ALTER}}$
	Inserting a word	○		EDIT	PROG	Searching for the word immediately before the word to be inserted → New data → $\boxed{\text{INSERT}}$
Verify	Verifying the memory			EDIT	PROG	[(OPRT)] → [▷] → [READ] → [EXEC]
Input/ output to/ from the FANUC Cassette	Searching a file for its beginning			EDIT	PROG	$\boxed{\text{N}}$ → FILE No. → [▷] → [F SRH] → [EXEC]
	Deleting a file	○		EDIT	PROG	$\boxed{\text{N}}$ → FILE No. → [▷] → [F DELETE] → [EXEC]
	Inputting a program	○		EDIT	PROG	$\boxed{\text{N}}$ → FILE No. → [▷] → [READ] → [EXEC]
	Outputting all the programs			EDIT	PROG	$\boxed{\text{O}}$ → –9999 → [▷] → [PUNCH] → [EXEC]
	Outputting one program			EDIT	PROG	$\boxed{\text{O}}$ → Program No. → [▷] → [PUNCH] → [EXEC]
	Verifying a program			EDIT	PROG	Searching a file for its beginning → $\boxed{\text{O}}$ → Program No. → [(OPRT)] → [▷] → [READ] → [EXEC]
Playback	Inputting NC data			TEACH-IN JOG	PROG	Move the machine. → $\boxed{\text{X}}$, $\boxed{\text{Y}}$ $\boxed{\text{Z}}$ → $\boxed{\text{INSERT}}$ → NC data → $\boxed{\text{INSERT}}$ → $\boxed{\text{EOB}}$ → $\boxed{\text{INSERT}}$

Classification	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Clear	Memory all clear			When the power is on	–	: [RESET] AND [DELETE]
	Parameters/offset		○	When the power is on	–	[RESET]
	Program clear		○	When the power is on		[DELETE]
	Program under editing when the power is off (PS101)			–	–	[PROG] AND [RESET]
	PMC RAM clear			When the power is on	–	[X] AND [O]

(DPL/MDI)

Classification	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Clear	All memory clear			Power ON	—	[7] AND [9]
	Parameter clear		○	Power ON	—	[PARAM]
	Program clear		○	Power ON	—	[DELETE]
	Alarm clear			—	—	[CAN] or Power OFF/ON
	Alarm P/S101 due to power-off during editing			—	—	[CAN] AND [ALARM]
	PMC RAM clear			Power ON	—	[O] AND [X]
Reset	OT alarm reset			Power ON	—	[P] AND [CAN]
Registration from MDI	Parameter input		○	—	DGNOS /PARAM	PARAM screen → [No.] → Number → [INPUT] → Data → [INPUT] PWE=0 → [CAN]
	PMC parameter input		SETTING DWE=1	—	DGNOS /PARAM	DGNOS screen → [PMC address] → Number → [INPUT] Data → [INPUT]
	Setting data input			—	VAR	Setting data screen → Cursor movement → Data → [INPUT]
	Offset data input	○		—	VAR	Offset data screen → [No.] → Data number → [INPUT] Data → [INPUT]
	Macro variable data input	○		—	VAR	Macro variable screen → [No.] → Data number → [INPUT] Data → [INPUT]

Classifi-cation	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Search	Program number search			EDIT/AUTO	PRGRM	O → Program number → ↓
	Sequence number search			AUTO	PRGRM	After program number search; N → Sequence number → ↓
	Address word search			EDIT	PRGRM	Word to be searched for → ↓
	Search address only			EDIT	PRGRM	Address to be searched for → ↓
	Parameter search			—	DGNOS /PARAM	PARAM screen → No. → Number → INPUT
	PMC parameter search			—	DGNOS /PARAM	DGNOS screen → PMC address → Number → INPUT
	Offset data search			—	VAR	Offset screen → No. → Data number → INPUT
	Macro variable data search			—	VAR	Macro variable screen → No. → Data number → INPUT
	Diagnosis search			—	DGNOS /PARAM	DGNOS screen → No. → Number→ INPUT
Editing	All program delete	○		EDIT	PRGRM	O → –9999 → DELETE
	One program delete	○		EDIT	PRGRM	O → Program number → DELETE
	Multiple block delete	○		EDIT	PRGRM	N → Sequence number → DELETE
	One block delete	○		EDIT	PRGRM	EOB → DELETE
	Word delete	○		EDIT	PRGRM	Search for word to be deleted → DELETE
	Word change	○		EDIT	PRGRM	After searching for word to be deleted; New data → ALTER
	Word insertion	○		EDIT	PRGRM	After searching for word after which word is to be inserted; New data → INSERT
Collation	Program collation			EDIT	PRGRM	READ
Registra-tion from external I/O	Parameter input		○	EDIT or emergency stop	DGNOS /PARAM	PARAM screen → READ
	Program input	○		EDIT	PRGRM	READ
	Offset data input	○		EDIT	VAR	Offset data screen → READ
	Macro variable data input	○		EDIT	PRGRM	READ → Mode AUTO → Execute the loaded program.
Output to external I/O	Parameter output			EDIT	DGNOS /PARAM	PARAM screen → WRITE
	All program output			EDIT	PRGRM	O → –9999 → WRITE
	One program output			EDIT	PRGRM	O → Program number → WRITE
	Offset data output			EDIT	VAR	Offset screen → WRITE
	Macro variable data output			EDIT	VAR	Macro variable screen → WRITE
Input/out-put to and from P-G and PG-mate	Ladder program input/output			—	DGNOS /PARAM	DGNOS screen → READ or WRITE → Operation on host Input/output is automatically identified with operation on host.

Classification	Function	KEY SW	SETTING PWE = 1	Mode	Function key	Operation
Input/output to and from FANUC cassette	Program registration	○		EDIT	PRGRM	N → File number → READ → READ
	All program output			EDIT	PRGRM	O → –9999 → WRITE
	One program output			EDIT	PRGRM	O → Program number → WRITE
	Search for beginning of file			EDIT	PRGRM	N → Program number, –9999, or –9998 → READ
	File delete	○		EDIT	PRGRM	N → File number → WRITE
	Program collation			EDIT	PRGRM	N → File number → READ → READ
	PMC parameter Ladder program input		○ (Only when PMC parameter is input)	Emergency stop	DGNOS /PARAM	DGNOS screen → No. File number → READ Data type is automatically identified.
	PMC parameter output			EDIT	DGNOS /PARAM	PMC parameter display → No. File number → WRITE
	Ladder program output			—	DGNOS /PARAM	DGNOS screen → No. File number → WRITE

WARNING

After completion of ladder program input, the power must be turned on again because the ladder program is in halt state.

– Numbers –

– A –

– B –

– C –

– D –

– E –

– M –

– N –

– O –

– P –

– R –

– S –

– T –

– U –

– V –

– W –

Revision Record

FANUC Power Mate–MODEL H OPERATOR'S MANUAL (B–62684EN)

Edition	Date	Contents	Edition	Date	Contents
02	Jan., '96	• Correction of erros			
01	Nov., '94	——————			